

ACCURATE EIGENVALUES AND SVDs OF TOTALLY NONNEGATIVE MATRICES*

PLAMEN KOEV†

Abstract. We consider the class of totally nonnegative (TN) matrices—matrices all of whose minors are nonnegative. Any nonsingular TN matrix factors as a product of nonnegative bidiagonal matrices. The entries of the bidiagonal factors parameterize the set of nonsingular TN matrices.

We present new $O(n^3)$ algorithms that, given the bidiagonal factors of a nonsingular TN matrix, compute its eigenvalues and SVD to high relative accuracy in floating point arithmetic, independent of the conventional condition number. All eigenvalues are guaranteed to be computed to high relative accuracy despite arbitrary nonnormality in the TN matrix.

We prove that the entries of the bidiagonal factors of a TN matrix determine its eigenvalues and SVD to high relative accuracy.

We establish necessary and sufficient conditions for computing the entries of the bidiagonal factors of a TN matrix to high relative accuracy, given the matrix entries.

In particular, our algorithms compute all eigenvalues and the SVD of TN Cauchy, Vandermonde, Cauchy–Vandermonde, and generalized Vandermonde matrices to high relative accuracy.

Key words. eigenvalue, singular value, high relative accuracy, totally positive matrix, totally nonnegative matrix, oscillatory matrix, sign regular matrix, bidiagonal matrix

AMS subject classifications. 65F15, 15A18

DOI. 10.1137/S0895479803438225

1. Introduction. The matrices with all minors nonnegative are called *totally nonnegative*. They appear in a wide area of problems and applications [6, 16, 19, 21, 22, 31] and can be notoriously ill-conditioned—the Hilbert matrix and the Vandermonde matrix with increasing nonnegative nodes are two examples of many.

When traditional algorithms are used to compute the eigenvalues or the singular values of an ill-conditioned TN matrix, only the largest eigenvalues and the largest singular values are computed with guaranteed relative accuracy. The tiny eigenvalues and singular values may be computed with no relative accuracy at all, even though they may be the only quantities of practical interest [8]. Their accurate computation using traditional algorithms is then only possible through an increase in the working precision, which may lead to a drastic increase in the computational time.

As our first major contribution we present new $O(n^3)$ algorithms that compute *all* eigenvalues and singular values of a nonsingular TN matrix to high relative accuracy. In particular, all computed eigenvalues and singular values, including the tiniest ones, must have correct sign and leading digits:

$$|\sigma_i - \hat{\sigma}_i| \leq O(\epsilon)\sigma_i \quad \text{and} \quad |\lambda_i - \hat{\lambda}_i| \leq O(\epsilon)|\lambda_i|.$$

The error bound for the computed eigenvalues must hold despite arbitrary nonnormality of A .

*Received by the editors December 1, 2003; accepted for publication (in revised form) by U. Helmke June 12, 2004; published electronically June 22, 2005. This work was supported by National Science Foundation grant DMS-0314286.

<http://www.siam.org/journals/simax/27-1/43822.html>

†Department of Mathematics, Massachusetts Institute of Technology, Cambridge, MA 02139 (plamen@math.mit.edu).

In contrast, traditional SVD and symmetric eigenvalue algorithms guarantee only small relative errors in the computed singular values and eigenvalues of TN matrices

$$|\sigma_i - \hat{\sigma}_i| \leq O(\epsilon)\sigma_{\max} \quad \text{and} \quad |\lambda_i - \hat{\lambda}_i| \leq O(\epsilon)|\lambda_{\max}|.$$

Previously proposed nonsymmetric eigenvalue algorithms guarantee only the Perron root of a TN matrix to be computed to high relative accuracy [15]. The accuracy in the smaller eigenvalues is also affected by the angle between left and right eigenvectors.

Relative accuracy is lost in the traditional eigenvalue and SVD algorithms because of certain roundoff errors in floating point arithmetic: cancellation of significant digits during subtraction of approximate intermediate quantities. On the other side, relative accuracy is preserved in addition of nonnegative quantities, multiplication, division and taking of square roots.

The idea in our algorithms is to structure the computations in such a way that subtractions are avoided.

Throughout this paper we will use the word “relative error” to mean “relative error in the eigenvalues and singular values.” We consider only nonsingular TN matrices.

The first step in our quest for accuracy is to choose a parameterization of the TN matrices to work with.

The matrix entries are a poor choice of parameters to represent a TN matrix because tiny relative perturbations in them can cause enormous relative perturbations in the eigenvalues and the singular values. For example, a small ϵ relative perturbation in the $(2, 2)$ entry of the 2×2 TN matrix

$$\begin{bmatrix} 1 & 1 \\ 1 & 1 + \epsilon \end{bmatrix} = \begin{bmatrix} 1 & 1 \\ 1 & 1 \end{bmatrix} \cdot \begin{bmatrix} 1 & \\ & \epsilon \end{bmatrix} \cdot \begin{bmatrix} 1 & 1 \\ & 1 \end{bmatrix}$$

from $1 + \epsilon$ to $1 + 2\epsilon$ perturbs the smallest eigenvalue from $\epsilon/2 + O(\epsilon^2)$ to $\epsilon + O(\epsilon^2)$ —a 100% relative change. In other words, errors introduced just from storing a matrix in the computer may change the small eigenvalues or singular values utterly.

Instead, we will represent any nonsingular TN matrix A uniquely as a product of nonnegative bidiagonal matrices

$$A = L^{(1)}L^{(2)} \dots L^{(n-1)}DU^{(n-1)}U^{(n-2)} \dots U^{(1)},$$

where D is diagonal and $L^{(k)}$ and $U^{(k)}$ are lower and upper unit bidiagonal matrices, respectively. We denote this set of parameters as $\mathcal{BD}(A)$ and discuss its properties in detail in section 2.2. A total of n^2 entries in $\mathcal{BD}(A)$ are nontrivial. These n^2 nonnegative entries parameterize the set of TN matrices.

Our second major contribution is to show that $\mathcal{BD}(A)$ determines the eigenvalues and the SVD of A accurately. In particular, the relative condition number of any eigenvalue with respect to perturbations in the entries of $\mathcal{BD}(A)$ is at most $2n^2$ independent of the angle between the left and right eigenvectors. The same structured condition number applies for the singular values. This means that our algorithms compute the eigenvalues and the singular values as accurately as they deserve.

Our algorithms take $\mathcal{BD}(A)$ as an input and perform a subtraction-free reduction of both the eigenvalue and the singular value problems to the bidiagonal singular value problem. The bidiagonal singular value problem is then solved to high relative accuracy using known means [12]. The above reduction uses only two types of matrix transformations, which we call elementary transformations (EETs):

1. subtraction of a positive multiple of a row/column from the next in order to create a zero;

2. addition of a positive multiple of a row/column to the previous one, followed by scaling of both rows/columns.

Rather than performing the EETs directly on the matrix A , we perform them on the bidiagonal decomposition $\mathcal{BD}(A)$ in such a way that subtractions are not required; hence the accuracy is preserved.

To solve the eigenvalue problem we use a well-known process (which we discuss in section 2.3) to implicitly reduce a TN matrix A to a (nonsymmetric) TN tridiagonal form T using similarity EETs. We then implicitly replace $t_{i-1,i}$ and $t_{i,i-1}$ by $(t_{i-1,i}t_{i,i-1})^{1/2}$, making T symmetric without changing its eigenvalues. We compute the eigenvalues of T as the squares of the singular values of its Cholesky factor.

For the singular value problem we perform an implicit Golub–Kahan bidiagonalization using Givens rotations. The trick in preserving the accuracy is to represent each Givens rotation as a sequence of two EETs.

It is a natural question to ask whether an accurate bidiagonal decomposition may be obtained, given the matrix entries. We know of no simple answer for TN matrices, but for (TP) matrices (matrices all of whose minors are positive), we do. In section 3 we prove that if A is TP, then we can compute $\mathcal{BD}(A)$ accurately if and only if we can compute all initial minors of A accurately. The $\dots$ are minors that are contiguous and include the first row or the first column.

In particular, since accurate formulas exist for the initial minors of TP Vandermonde, Cauchy, Cauchy–Vandermonde, and generalized Vandermonde matrices, we can readily compute the eigenvalues and the singular values of these matrices accurately.

Our algorithms and theory apply with little or no change to TP, $\dots$ (some power is TP), and inverses of TN matrices.

The idea of using bidiagonal decompositions to perform accurate matrix computations is not new. Such decompositions are used in every Björck–Pereyra method to obtain accurate solutions to structured TN linear systems (see section 2.4). Also, in [14] Parlett and Dhillon discard the entries of the tridiagonal matrix and choose to work with the bidiagonal Cholesky factors.

Organization of the paper. We review the properties of TN matrices and their bidiagonal decompositions in section 2. We present necessary and sufficient conditions for computing accurate bidiagonal decompositions in section 3. We describe how to perform EETs accurately in section 4. We present our new eigenvalue and singular value algorithms in sections 5 and 6, respectively. The perturbation theory and error analysis are in section 7. Finally, we present numerical experiments in section 8.

2. Background. In this section we review some well-known properties of TN matrices, their bidiagonal decompositions, Neville elimination, Björck–Pereyra methods, and the sensitivity of the SVD of a TN matrix to structured perturbations.

2.1. Properties of TN matrices. Excellent references on TN and related matrices are [2, 16, 21, 22, 25, 31]. We mention only some of the relevant properties.

The eigenvalues of a TN matrix are real and positive. The irreducible TN matrices are called $\dots$, and their eigenvalues are also distinct, as are their singular values. The product of two TN matrices is TN. A bidiagonal matrix with a positive diagonal and only one nonnegative off-diagonal entry is TN. All TN matrices are obtained by forming a product of such simple bidiagonal matrices, as we describe next.

2.2. Bidiagonal decomposition of TN matrices. A dense matrix A can be reduced to an upper triangular form by a variety of elimination methods. A fixed pivot row is used in Gaussian elimination where at each step an entry (say a_{ij}) is set to zero, resulting in the decomposition $A = K \cdot A'$, where $a'_{ij} = 0$ and K differs from the identity matrix only in its (i, j) th entry $l_{ij} = a_{ij}/a_{jj}$:

$$K = \begin{bmatrix} 1 & & & & \\ & \ddots & & & \\ & & \ddots & & \\ & & & \ddots & \\ l_{ij} & & & & 1 \end{bmatrix}.$$

At the end of this elimination process all matrices K are “assembled” in a unit lower triangular matrix L with subdiagonal entries l_{ij} .

If A is TN, then a different elimination approach, called *total nonnegativity elimination*, sheds more light on the structure of A [23, 38]. For completeness of the presentation, we describe this process in detail following [25].

The main difference between Gaussian and Neville eliminations is that the latter uses only adjacent rows or columns for elimination. For example the (i, j) th entry $a_{ij} \neq 0$ is eliminated by subtracting a multiple $m_{ij} = a_{ij}/a_{i-1,j}$ of the $(i-1)$ st row from the i th. In matrix form this results in the decomposition $A = E_i(m_{ij}) \cdot A'$, where E_i differs from the identity only in its $(i, i-1)$ entry:

$$(2.1) \quad E_i(m_{ij}) \equiv \begin{bmatrix} 1 & & & & \\ & \ddots & & & \\ & & 1 & & \\ & & m_{ij} & 1 & \\ & & & \ddots & \\ & & & & 1 \end{bmatrix}.$$

The total nonnegativity is preserved by Neville elimination [23]; thus $m_{ij} \geq 0$. Once A is reduced to upper triangular form U , the same elimination process is applied to U using only adjacent columns. As a result A is factored as a product of $(n^2 - n)/2$ matrices of type (2.1), a diagonal matrix D , and $(n^2 - n)/2$ transposes of matrices of type (2.1).

The reader may have noticed that we bravely divided by $a_{i-1,j}$ in forming m_{ij} . This will never lead to division by zero because A is TN. Indeed, if $a_{i-1,k}$ is any nonzero (thus positive) entry in the $(i-1)$ st row of A , then the 2×2 minor of A consisting of rows $i-1$ and i and columns j and k ($j < k$) must be nonnegative. Therefore $a_{ij} > 0$ implies $a_{i-1,j} > 0$ and in turn $a_{i-1,j} = 0$ implies $a_{ij} = 0$. Equivalently, $m_{ij} = 0$ implies $m_{i+1,j} = \dots = m_{nj} = 0$, and thus

$$(2.2) \quad m_{ij} = 0 \text{ implies } \begin{cases} m_{kj} = 0 \text{ for all } k > i \text{ if } i > j; \\ m_{ik} = 0 \text{ for all } k > j \text{ if } i < j. \end{cases}$$

The matrices E_i have the following important properties:

$$(2.3) \quad \begin{aligned} E_i^{-1}(x) &= E_i(-x), \\ E_i(x)E_j(y) &= E_j(y)E_i(x), \text{ unless } |i-j| = 1 \text{ and } xy \neq 0, \\ E_i(x)E_i(y) &= E_i(x+y). \end{aligned}$$

If we apply Neville elimination to A , eliminating one subdiagonal at a time, starting with the $(n, 1)$ entry we obtain

$$\begin{aligned} A &= (E_n(m_{n1})) \cdot (E_{n-1}(m_{n-1,1})E_n(m_{n2})) \cdots \\ &\quad \cdot (E_2(m_{21})E_3(m_{32}) \cdots E_n(m_{n,n-1})) \\ &\quad \cdot D \cdot (E_n^T(m_{n-1,n}) \cdots E_3^T(m_{23})E_2^T(m_{12})) \cdots \\ &\quad \cdot (E_n^T(m_{2n})E_{n-1}^T(m_{1,n-1})) \cdot (E_n^T(m_{1n})). \end{aligned}$$

We “assemble” the E_i ’s inside the parentheses above into bidiagonal matrices

$$\begin{aligned} L^{(k)} &\equiv E_{n-k+1}(m_{n-k+1,1})E_{n-k+2}(m_{n-k+2,2}) \cdots E_n(m_{nk}) \\ &= \begin{bmatrix} 1 & & & & & \\ & \ddots & & & & \\ & & m_{n-k+1,1} & & & \\ & & & 1 & & \\ & & & m_{n-k+2,2} & 1 & \\ & & & & \ddots & \ddots \\ & & & & & m_{nk} & 1 \end{bmatrix}, \end{aligned}$$

where $l_{i-1}^{(k)} \equiv (L^{(k)})_{i,i-1} = m_{i,k+i-n}$. We analogously form unit upper bidiagonal matrices $U^{(k)}$. We know of no universally convenient way to reference the off-diagonal entries in the bidiagonal matrices $L^{(k)}$. We will use either $l_{i-1}^{(k)}$ or m_{ij} , depending on whether the reference is in connection with $L^{(k)}$ or the entry (a_{ij}) being eliminated.

We now present a fundamental theorem from [25] which describes the unique bidiagonal decomposition of a TN matrix (see also [3, 6, 7, 18, 19] for equivalent combinatorial interpretations).

THEOREM 2.1 (Gasca and Peña [25]).

$$(2.4) \quad A = L^{(1)} \cdots L^{(n-1)} \cdot D \cdot U^{(n-1)} \cdots U^{(1)},$$

$$D = \text{diag}(d_1, d_2, \dots, d_n) \quad L^{(k)} \quad U^{(k)}$$

1. $d_i > 0$, $i = 1, \dots, n$.
2. $l_i^{(k)} = u_i^{(k)} = 0$, $i < n - k$.
3. $l_i^{(k)} \geq 0$, $u_i^{(k)} \geq 0$, $i \geq n - k$.
4. $l_i^{(k)} = 0$, $l_{i+s}^{(k-s)} = 0$, $s = 1, \dots, k - 1$, $u_i^{(k)} = 0$, $u_{i+s}^{(k-s)} = 0$, $s = 1, \dots, k - 1$.

The bidiagonal decomposition (2.4) (which we denote by $\mathcal{BD}(A)$) will be the ultimate unique representation of a TN matrix in this paper. In our algorithms, EETs will introduce “bulges” in $\mathcal{BD}(A)$. The bulges will be chased away to obtain the unique bidiagonal decomposition of the transformed matrix. We will refer to this theorem to confirm if a particular bidiagonal decomposition of a TN matrix is its unique $\mathcal{BD}(A)$.

2.3. Reduction of a (nonsymmetric) matrix to tridiagonal form. In our eigenvalue algorithm in section 5 we will use similarity EETs to implicitly reduce a (nonsymmetric) TN matrix to tridiagonal form using this well-known process [26].

We start by introducing a zero in the $(n, 1)$ entry by subtracting a multiple of the $(n - 1)$ st row from the n th. Then we add the same multiple of the n th column to the

$(n-1)$ st to complete the similarity transformation. In the next step, we subtract a multiple of the $(n-1)$ st column from the n th to create a zero in position $(1, n)$; then again, we complete the similarity by adding the same multiple of the n th row to the $(n-1)$ st. The zero in position $(n, 1)$ is not disturbed by the last operation. Next, we introduce zeros in positions $(n-1, 1)$ and $(1, n-1)$ in a similar manner, which does not disturb the zeros already created in positions $(n, 1)$ and $(1, n)$. We continue the same process until all but the first two entries of the first row and column are zero. Then we apply the same process to the trailing $(n-1) \times (n-1)$ principal submatrix and so on until the matrix is reduced to tridiagonal form.

This process may be unstable for general nonsymmetric matrices, but it is our contribution to show that it is always stable for TN matrices.

Other analogous schemes for reduction to tridiagonal form also exist [37].

2.4. Björck–Pereyra methods. In 1970 Björck and Pereyra presented an algorithm for computing the solution to a Vandermonde system of equations $Vx = b$ in $O(n^2)$ time [4]. Although rooted in the process of Newton interpolation, their algorithm, in essence, computes an accurate decomposition of V^{-1} as a product of bidiagonal matrices and applies it to b . When the nodes of interpolation are nonnegative and increasing (i.e., when V is TN) and b has alternating sign pattern, there is no subtractive cancellation in forming x , and every entry of x is computed accurately. This phenomenon was first observed by Kahan in 1963 [30]; Higham later proved that it applies essentially to all Björck–Pereyra methods [27, 28].

Björck–Pereyra methods were also derived for Cauchy [5], Cauchy–Vandermonde [34], and generalized Vandermonde [13] matrices.

The decomposition $\mathcal{BD}(V)$ is readily and accurately obtainable from this representation of V^{-1} using the formulas (2.3) and inverting D (see also [25]). Therefore by using the algorithms from sections 5 and 6 one can compute accurate eigenvalues and SVD of any TN matrix for which Björck–Pereyra methods exist.

2.5. If A is TN, then $\mathcal{BD}(A)$ determines the SVD of A accurately. By using the Cauchy–Binet identity, Demmel et al. established in [11, section 9] that all minors of a TN matrix are determined accurately by the entries of $\mathcal{BD}(A)$. Thus so is its LDU decomposition resulting from Gaussian elimination with complete pivoting (every element of L , D , and U is a quotient of minors), and in turn so is the SVD. Using these results and ignoring δ^2 and higher-order terms, one can conclude that a δ relative perturbation in the n^2 nontrivial entries of $\mathcal{BD}(A)$ can cause a relative perturbation of at most $O(n^2\delta)$ in any minor (see section 7) and at most $O(n^2\kappa\delta)$ in any singular value. Here $\kappa \equiv \max\{\text{cond}(L), \text{cond}(U)\}$ is a bound on the condition number of L and U , which in practice never exceeds $O(n)$ [39, p. 213]. The angle $\theta(u_i, \hat{u}_i)$ between the i th singular vector and its perturbed counterpart is bounded by

$$\sin \theta(u_i, \hat{u}_i) \leq \frac{O(n^2\kappa\delta)}{\min_{j \neq i} |\sigma_i - \sigma_j| / \sigma_i}.$$

In section 7 we prove a slightly tighter perturbation bound for the singular values.

3. Necessary and sufficient conditions for computing accurate $\mathcal{BD}(A)$.

When can we compute an accurate $\mathcal{BD}(A)$, given A ? Neville elimination is the obvious choice, but it involves subtractions, and therefore likely loss of accuracy. On the other side, if A is TP, the entries of $\mathcal{BD}(A)$ are products and quotients of initial minors of A and can be computed accurately if these minors can be. We recall that a minor is $\mathbf{v}_i \mathbf{v}_j$, if it is contiguous and includes the first row or the first column.

We will use MATLAB [35] notation for submatrices: $A(i : j, k : l)$ will denote the submatrix of A consisting of rows i through j and columns k through l .

PROPOSITION 3.1. A is TP if and only if A is TP and $\mathcal{BD}(A)$.

$$(3.1) \quad l_i^{(k)} = \frac{\det A(q+1 : i+1, 1 : i-q+1)}{\det A(q+1 : i, 1 : i-q)} \cdot \frac{\det A(q : i-1, 1 : i-q)}{\det A(q : i, 1 : i-q+1)},$$

$$(3.2) \quad u_i^{(k)} = \frac{\det A(1 : i-q+1, q+1 : i+1)}{\det A(1 : i-q, q+1 : i)} \cdot \frac{\det A(1 : i-q, q : i-1)}{\det A(1 : i-q+1, q : i)},$$

$$(3.3) \quad d_i = \frac{\det A(1 : i, 1 : i)}{\det A(1 : i-1, 1 : i-1)},$$

where $q = n - k$; $i \geq q = n - k$, (3.1) and (3.2) hold, $l_i^{(k)} = u_i^{(k)} = 0$, $i < n - k$. Expressions (3.1), (3.2), and (3.3) follow directly from Lemma 2.6(1) in [23]. For the second part “sufficiency” is obvious, as is “necessity” for the leading principal minors: $A(1 : i, 1 : i) = d_1 d_2 \cdots d_i$. Next we write $m \equiv i - q$ in (3.2) to obtain

$$(3.4) \quad \det A(1 : m+1, q+1 : q+m+1) = \det A(1 : m+1, q : q+m) \times \left[u_i^{(n-q)} \cdot \frac{\det A(1 : m, q+1 : q+m)}{\det A(1 : m, q : q+m-1)} \right].$$

The expression in the brackets involves only initial minors of order m . We now use (3.4) and induction on m from 0 to $n-1$ and q from 1 to $n-m-1$ to conclude that all initial minors of A that include the first row are computable accurately (since (3.4) involves only multiplications and divisions) at only three arithmetic operations per initial minor. Analogously we draw the same conclusion about the initial minors that include the first column by starting with (3.1). $\square$

Proposition 3.1 allows us to compute accurate bidiagonal decompositions of TP structured matrices. For example, the Vandermonde and Cauchy matrices

$$V = \left[x_i^{j-1} \right]_{i,j=1}^n \quad \text{and} \quad C = \left[\frac{1}{x_i + y_j} \right]_{i,j=1}^n$$

are TP when $0 < x_1 < x_2 < \cdots < x_n$, and $x_1 < x_2 < \cdots < x_n$, $y_1 < y_2 < \cdots < y_n$, $x_1 + y_1 > 0$, respectively. For $\mathcal{BD}(V)$ we have

$$(3.5) \quad d_i = \prod_{j=1}^{i-1} (x_i - x_j), \quad l_i^{(k)} = \prod_{j=n-k}^{i-1} \frac{x_{i+1} - x_{j+1}}{x_i - x_j}, \quad u_i^{(k)} = x_{i+k-n+1},$$

where $l_i^{(k)} = u_i^{(k)} = 0$ for $i < n - k$ and for $\mathcal{BD}(C)$ we have

$$(3.6) \quad d_i = \prod_{k=1}^{i-1} \frac{(x_i - x_k)(y_i - y_k)}{(x_i + y_k)(y_i + x_k)},$$

$$l_i^{(k)} = \frac{x_{n-k} + y_{i-n+k+1}}{x_i + y_{i-n+k+1}} \prod_{l=n-k}^{i-1} \frac{x_{i+1} - x_{l+1}}{x_i - x_l} \cdot \prod_{l=1}^{i-n+k-1} \frac{x_i + y_l}{x_{i+1} + y_l},$$

$$u_i^{(k)} = \frac{y_{n-k} + x_{i-n+k+1}}{y_i + x_{i-n+k+1}} \prod_{l=n-k}^{i-1} \frac{y_{i+1} - y_{l+1}}{y_i - y_l} \cdot \prod_{l=1}^{i-n+k-1} \frac{y_i + x_l}{y_{i+1} + x_l},$$

where $l_i^{(k)} = u_i^{(k)} = 0$ for $i < n - k$.

The subtractions in formulas (3.5) and (3.6) involve only initial (thus exact) data and will not lead to subtractive cancellation. In section 7 we prove that $\mathcal{BD}(V)$ and $\mathcal{BD}(C)$, and in turn the eigenvalues and the singular values of V and C , will be computed accurately. Similar accurate formulas for $\mathcal{BD}(A)$ may be obtained when A is a TP generalized Vandermonde [13] or a Cauchy–Vandermonde [33] matrix.

The formulas (3.1)–(3.3) are not, in general, valid for TN matrices (which are the closure of TP matrices). We do not know of necessary and sufficient conditions for being able to compute $\mathcal{BD}(A)$ accurately when A is TN, but not TP.

4. Computing with $\mathcal{BD}(A)$. In this section we show how to accurately apply an EET to $\mathcal{BD}(A)$. In other words, if A is TN and A' is obtained from A by applying an EET, we will show how to compute $\mathcal{BD}(A')$ directly from $\mathcal{BD}(A)$ without performing any subtractions.

It is convenient to store the n^2 nontrivial entries of $\mathcal{BD}(A)$ as an $n \times n$ square array. In position (i, j) we store the multiplier m_{ij} used to eliminate the (i, j) th entry in A , and on the diagonal we store D :

$$(\mathcal{BD}(A))_{ij} = \begin{cases} d_i, & i = j, \\ m_{ij}, & i \neq j; \end{cases} \quad \text{equivalently} \quad (\mathcal{BD}(A))_{ij} = \begin{cases} d_i, & i = j, \\ l_{i-1}^{(n-i+j)}, & i > j, \\ u_{j-1}^{(n-j+i)}, & i < j. \end{cases}$$

From now on we will use $\mathcal{BD}(A)$ to refer to either the actual bidiagonal decomposition (2.4) or the $n \times n$ array used for its compact storage, but it should cause no confusion.

4.1. The bidiagonal decomposition of the 3×3 TN matrix

$$\begin{bmatrix} 1 & 2 & 6 \\ 4 & 13 & 69 \\ 28 & 131 & 852 \end{bmatrix} = \begin{bmatrix} 1 & & \\ & 1 & \\ & 7 & 1 \end{bmatrix} \begin{bmatrix} 1 & & \\ 4 & 1 & \\ & 8 & 1 \end{bmatrix} \begin{bmatrix} 1 & & \\ & 5 & \\ & & 9 \end{bmatrix} \begin{bmatrix} 1 & 2 & \\ & 1 & 6 \\ & & 1 \end{bmatrix} \begin{bmatrix} 1 & & \\ & 1 & 3 \\ & & 1 \end{bmatrix}$$

is stored as

$$\mathcal{BD} \left(\begin{bmatrix} 1 & 2 & 6 \\ 4 & 13 & 69 \\ 28 & 131 & 852 \end{bmatrix} \right) = \begin{Bmatrix} 1 & 2 & 3 \\ 4 & 5 & 6 \\ 7 & 8 & 9 \end{Bmatrix}.$$

We can formally transpose $\mathcal{BD}(A)$ to obtain $\mathcal{BD}(A^T)$:

$$A^T = (U^{(1)})^T (U^{(2)})^T \dots (U^{(n-1)})^T D (L^{(n-1)})^T (L^{(n-2)})^T \dots (L^{(1)})^T,$$

which we write as $\mathcal{BD}(A^T) = (\mathcal{BD}(A))^T$.

4.1. Subtracting a positive multiple of a row from the next to create a zero. We now show that in an elimination process of reducing a matrix to upper triangular form, subtracting a row from the next in order to create a zero is equivalent to simply setting an entry in $\mathcal{BD}(A)$ to zero and performing

PROPOSITION 4.2. Let A be a TN matrix. If $A(k-1:n, 1:l) = 0$ and $a_{k-1,l} > 0$, $a_{kl} > 0$, $k > l$, then $A' = A$ with a_{kl} replaced by $a_{kl} - a_{k-1,l}$.

4.2. Adding a positive multiple of a column/row to the previous one.

In matrix form this is equivalent to forming the product $AJ_i(x, y)$, where

$$J_i(x, y) \equiv \begin{bmatrix} 1 & & & & \\ & \ddots & & & \\ & & y & & \\ & & x & 1/y & \\ & & & & \ddots \\ & & & & & 1 \end{bmatrix}$$

Adding a multiple of a row to the previous one followed by scaling of both rows is analogous and computed as $\mathcal{BD}(J_i^T(x, y) \cdot A) = \mathcal{BD}(A^T \cdot J_i(x, y))^T$.

$$(4.1) \quad AJ_i(x, y) = L^{(1)}L^{(2)}L^{(3)} \dots L^{(n-1)}DU^{(n-1)} \dots U^{(2)}U^{(1)} \cdot J_i(x, y).$$
$$(4.2) AJ_i = L^{(1)} \dots L^{(i-2)} L^{(i-1)} L^{(i)} \dots L^{(n-2)} L^{(n-1)} DU^{(n-1)} \dots U^{(3)} U^{(2)} U^{(1)} J_i$$

$$(4.3) \quad = L^{(1)} \dots L^{(i-2)} L^{(i-1)} L^{(i)} \dots L^{(n-2)} L^{(n-1)} D U^{(n-1)} \dots U^{(3)} U^{(2)} J_i^{(1)} \mathcal{U}^{(1)}$$

$$= L^{(1)} \dots L^{(i-2)} L^{(i-1)} L^{(i)} \dots L^{(n-2)} L^{(n-1)} DU^{(n-1)} \dots \underline{U^{(3)} J_i^{(2)} \mathcal{U}^{(2)} \mathcal{U}^{(1)}}$$

$$(4.4) \quad = L^{(1)} \dots L^{(i-2)} L^{(i-1)} L^{(i)} \dots L^{(n-2)} L^{(n-1)} D J_i^{(n-1)} \mathcal{U}^{(n-1)} \dots \mathcal{U}^{(2)} \mathcal{U}^{(1)}$$

$$(4.5) \quad = L^{(1)} \dots L^{(i-2)} L^{(i-1)} L^{(i)} \dots L^{(n-2)} L^{(n-1)} J_i^{(n)} \mathcal{D}U^{(n-1)} \dots U^{(2)} U^{(1)}$$

$$\begin{aligned}
&= L^{(1)} \dots L^{(i-2)} L^{(i-1)} L^{(i)} \dots L^{(n-3)} \underline{L^{(n-2)} J_{i+1}^{(n+1)} \mathcal{L}^{(n-1)} \mathcal{D}\mathcal{U}^{(n-1)} \dots \mathcal{U}^{(2)} \mathcal{U}^{(1)}} \\
&= L^{(1)} \dots L^{(i-2)} L^{(i-1)} L^{(i)} \dots L^{(n-3)} \underline{J_{i+2}^{(n+2)} \mathcal{L}^{(n-2)} \mathcal{L}^{(n-1)} \mathcal{D}\mathcal{U}^{(n-1)} \dots \mathcal{U}^{(2)} \mathcal{U}^{(1)}} \\
&= \dots
\end{aligned}$$

$$(4.6) \quad = L^{(1)} \dots L^{(i-2)} L^{(i-1)} J_n^{(2n-i)} \mathcal{L}^{(i+1)} \dots \mathcal{L}^{(n-2)} \mathcal{L}^{(n-1)} \mathcal{D}\mathcal{U}^{(n-1)} \dots \mathcal{U}^{(2)} \mathcal{U}^{(1)}$$

$$(4.7) \quad = L^{(1)} \dots L^{(i-2)} \mathcal{L}^{(i-1)} \mathcal{L}^{(i)} \dots \mathcal{L}^{(n-2)} \mathcal{L}^{(n-1)} \mathcal{D} \mathcal{U}^{(n-1)} \dots \mathcal{U}^{(2)} \mathcal{U}^{(1)}.$$

The factors that are transformed at each step of the above transformations are underlined. The matrix $J_m^{(k)}$ equals $J_m(x_k, y_k)$ for some x_k and y_k . The matrices $\mathcal{L}^{(k)}$, $\mathcal{D}$, and $\mathcal{U}^{(k)}$, $k = 1, 2, \dots, n-1$, are unit lower bidiagonal, diagonal and unit upper diagonal, respectively, with nonzero pattern consistent with Theorem 2.1.

We will now explain how to obtain (4.3) from (4.2) and so on until we obtain (4.7). Then we will prove that (4.7) is in fact $\mathcal{BD}(AJ_i(x, y))$.

We start with the transformations (4.2)–(4.4). At each step we are given matrices

$$U = \begin{bmatrix} 1 & u_1 & & & \\ & 1 & u_2 & & \\ & & \ddots & \ddots & \\ & & & 1 & u_{n-1} \\ & & & & 1 \end{bmatrix}, \quad u_j \geq 0, \quad j = 1, 2, \dots, n-1,$$

and $J_i(x, y)$, $x > 0, y > 0$. We need to compute matrices

$$\mathcal{U} = \begin{bmatrix} 1 & u'_1 & & & \\ & 1 & u'_2 & & \\ & & \ddots & \ddots & \\ & & & 1 & u'_{n-1} \\ & & & & 1 \end{bmatrix}, \quad u'_j \geq 0, \quad j = 1, 2, \dots, n-1,$$

and $J_i(x', y')$ such that

$$(4.8) \quad J_i(x', y') \cdot \mathcal{U} = U \cdot J_i(x, y).$$

By comparing the entries on both sides of (4.8), we obtain

$$(4.9) \quad \begin{aligned} x' &= x, \\ y' &= y + u_{i-1}x, \\ u'_{i-2} &= u_{i-2}y, \\ u'_{i-1} &= u_{i-1}/(yy'), \\ u'_i &= u_i y', \\ u'_j &= u_j \text{ for } j \notin \{i-2, i-1, i\}. \end{aligned}$$

The expressions (4.9) involve no subtractions and require only six arithmetic operations. No division by 0 can occur because $y > 0$. The inequalities $u_j \geq 0$, $x > 0$, $y > 0$ imply $u'_j \geq 0$, $x' > 0$, $y' > 0$. Also $u'_j = 0$ only when $u_j = 0$, $j = 1, 2, \dots, n-1$.

All transformations (4.2)–(4.4) are performed according to (4.8).

The reader familiar with the LR algorithm may have noticed that the equation (4.8) is equivalent to one step of LR, performed accurately using differential quotient-difference recurrences [17, 36, 39].

Now we turn our attention to the transformation from (4.4) to (4.5):

$$(4.10) \quad J_i(x', 1) \cdot \mathcal{D} = D \cdot J_i(x, y).$$

This transformation is straightforward and subtraction-free:

$$(4.11) \quad \begin{aligned} \mathcal{D} &= \text{diag}(d_1, d_2, \dots, d_{i-2}, d_{i-1}y, d_i/y, d_{i+1}, \dots, d_n), \\ x' &= d_i x / (d_{i-1}y). \end{aligned}$$

No division by zero can occur in (4.11) because $yd_{i-1} > 0$.

In the transformations (4.5)–(4.7) we are given $x > 0$ and a matrix

$$L = \begin{bmatrix} 1 & & & & \\ l_1 & 1 & & & \\ & \ddots & \ddots & & \\ & & l_{n-2} & 1 & \\ & & & l_{n-1} & 1 \end{bmatrix}, \quad l_j \geq 0, \quad j = 1, 2, \dots, n-1.$$

We need to compute $x' \geq 0$ and a matrix

$$\mathcal{L} = \begin{bmatrix} 1 & & & & \\ l'_1 & 1 & & & \\ & \ddots & \ddots & & \\ & & l'_{n-2} & 1 & \\ & & & l'_{n-1} & 1 \end{bmatrix}, \quad l'_j \geq 0, \quad j = 1, 2, \dots, n-1,$$

such that

$$(4.12) \quad J_{k+1}(x', 1) \cdot \mathcal{L} = L \cdot J_k(x, 1).$$

By convention we set $J_{n+1} \equiv I$. The bulge chasing is over once $J_{k+1} = I$. This happens when $k = n$ or possibly earlier if x' becomes zero.

By comparing the entries on both sides of (4.12) we obtain

$$(4.13) \quad l'_j = l_j \quad \text{if } j \notin \{k-1, k\},$$

$$(4.14) \quad l'_{k-1} = l_{k-1} + x,$$

$$(4.15) \quad l'_k = l_{k-1}l_k/l'_{k-1} \quad \text{if } k \leq n-1,$$

$$(4.16) \quad x' = xl_k/l'_{k-1} \quad \text{if } k \leq n-1.$$

The expressions (4.13)–(4.16) are subtraction-free and require only four arithmetic operations (we only need compute l_k/l'_{k-1} once in (4.15)–(4.16)). The inequalities $x > 0$ and $l_j \geq 0$ imply $x' \geq 0$ and $l'_j \geq 0$; thus division by 0 cannot occur.

All transformations (4.5)–(4.7) are performed according to (4.12).

To complete the proof we need to show that (4.7) is the unique $\mathcal{BD}(A \cdot J_i(x, y))$. It suffices to verify that the conditions of Theorem 2.1 are satisfied.

Conditions 1, 2, and 3 clearly are satisfied. The second part of condition 4 ($u_j^{(k)} = 0$ implies $u_{j+s}^{(k-s)} = 0$ for $s = 1, 2, \dots, k-1$) is also satisfied, because no zeros or nonzeros are introduced in the upper bidiagonal factors $\mathcal{U}^{(k)}$ in steps (4.2) to (4.5).

We will now establish that the first part of condition 4 holds as well. Let $l_j^{(m)}, j = 1, 2, \dots, n-1$, be the off-diagonal entries in $\mathcal{L}^{(m)}$, and assume that some $l_j^{(m)} = 0$. We need to prove that $l_{j+s}^{(m-s)} = 0$ for $s = 1, \dots, m-1$. It suffices to show that $l_{j+1}^{(m-1)} = 0$, since the rest will follow by induction on s .

The entry $l_j^{(m)}$ could not have been computed using (4.14) because $x > 0$. It must have been computed using (4.13) or (4.15).

If (4.13) was used, then (4.13) will also be used to compute $l_{j+1}^{(m-1)} = l_{j+1}^{(m)}$ on the next step of (4.12). Condition 4 of Theorem 2.1 along with $l_j^{(m)} = 0$ implies $l_{j+1}^{(m-1)} = 0$. Therefore $l_{j+1}^{(m-1)} = 0$.

If (4.15) was used to compute $l_j^{(m)}$, then $0 = l_j^{(m)} = l_{j-1}^{(m)} l_j^{(m)} / (l_{j-1}^{(m)} + x)$. Thus $l_{j-1}^{(m)} = 0$ or $l_j^{(m)} = 0$. Condition 4 for $\mathcal{BD}(A)$ now implies $l_j^{(m-1)} = 0$ or $l_{j+1}^{(m-1)} = 0$. Either way, on the next step of (4.12), (4.15) gives $l_{j+1}^{(m-1)} = l_j^{(m-1)} l_{j+1}^{(m-1)} / l_j^{(m-1)} = 0$.

To complete the proof it remains only to count the number of arithmetic operations. Since $u_j^{(k)} = 0$ for $j < n - k$, the matrix J_i commutes with $U^{(1)}, U^{(2)}, \dots, U^{(n-i-1)}$. Therefore (4.9) are performed at most i times; (4.11) are performed once, and (4.14)–(4.16) are performed at most $n - i + 1$ times. The total number of arithmetic operations does not exceed $6i + 4 + 4(n - i + 1) = 4n + 2i + 8$. $\square$

ALGORITHM 4.4. $\mathcal{BD}(A)$

4.3 $\mathcal{BD}(A \cdot J_i(x, y))$ $4n + 2i + 8$

```

function B = TNAddToPrevious(B, x, y, i)
n = size(B, 1)
if i < n then
    B(1, i + 1) = B(1, i + 1) * y
end
for j = 1 : i - 1
    z = y + B(j, i) * x
    if j > 1 then
        B(j - 1, i - 1) = B(j - 1, i - 1) * y
    end
    B(j, i) = B(j, i) / y / z
    if i < n then
        B(j + 1, i + 1) = B(j + 1, i + 1) * z
    end
    y = z
end
B(i - 1, i - 1) = B(i - 1, i - 1) * y
x = B(i, i) * x / B(i - 1, i - 1)
B(i, i) = B(i, i) / y
j = i
while (j < n) and (x > 0)
    z = B(j, i - 1)
    B(j, i - 1) = z + x
    y = B(j + 1, i) / B(j, i - 1)
    B(j + 1, i) = y * z
    x = y * x
    j = j + 1
end
B(n, i - 1) = B(n, i - 1) + x

```

4.3. Givens rotations. Let A be TN and G be a Givens rotation intended to create a zero in the process of Golub–Kahan bidiagonalization of A . We will now demonstrate how to compute $\mathcal{BD}(GA)$, given $\mathcal{BD}(A)$.

The trick is to represent G as a sequence of two EETs. Indeed, applying a Givens rotation to create a zero in position (say $(n, 1)$) is equivalent to (1) subtracting a (positive) multiple of the $(n - 1)$ st row from the n th in order to create a zero in position $(n, 1)$, followed by (2) adding a positive multiple of the n th row to the $(n - 1)$ st, and finally (3) scaling the last two rows.

Let

$$G = \begin{bmatrix} c & s \\ -s & c \end{bmatrix},$$

where $s^2 + c^2 = 1$, be a Givens rotation used to set the $(n, 1)$ entry of A to zero. Then

$$G \cdot \begin{bmatrix} a_{n-1,1} & a_{n-1,2} \\ a_{n1} & a_{n2} \end{bmatrix} = \begin{bmatrix} c & s \\ -s & c \end{bmatrix} \cdot \begin{bmatrix} a_{n-1,1} & a_{n-1,2} \\ a_{n1} & a_{n2} \end{bmatrix} = \begin{bmatrix} a_{n-1,1}^{(1)} & a_{n-1,2}^{(1)} \\ 0 & a_{n2}^{(1)} \end{bmatrix}.$$

A simple calculation shows that $c = 1/\sqrt{1+x^2}$ and $s = x/\sqrt{1+x^2}$, where

$$x = a_{n1}/a_{n-1,1} = l_{n-1}^{(1)} = (\mathcal{BD}(A))_{n1}$$

is the only nonzero off-diagonal entry in $L^{(1)}$ of $\mathcal{BD}(A)$. Now write G as

$$G = \begin{bmatrix} 1/c & cx \\ 0 & c \end{bmatrix} \cdot \begin{bmatrix} 1 & 0 \\ -x & 1 \end{bmatrix}.$$

Therefore applying a Givens rotation to the last two rows of A in order to create a zero in position $(n, 1)$ is equivalent to the multiplication

$$A^{(1)} = J_n^T(cx, 1/c) \cdot E_n(-x) \cdot A.$$

Forming $A' = E_n(-x) \cdot A$ is equivalent to subtracting a positive multiple of the $(n-1)$ st row from the n th in order to create a zero in position $(n, 1)$. Therefore we obtain $\mathcal{BD}(A')$ from $\mathcal{BD}(A)$ by setting $(\mathcal{BD}(A))_{n1}$ to zero. The multiplication $A^{(1)} = J_n^T(cx, 1/c) \cdot A'$ is performed using Algorithm 4.4 to compute

$$\mathcal{BD}(A^{(1)}) = (\mathcal{BD}((A')^T \cdot J_n(cx, 1/c)))^T.$$

Applying a Givens rotation on the right is analogous, as is the application of subsequent Givens rotations in the reduction of A to bidiagonal form.

5. The eigenvalue problem. In this section we present our algorithm for computing accurate eigenvalues of a TN matrix A , given $\mathcal{BD}(A)$.

We first implicitly reduce A to tridiagonal form by using similarity EETs as described in section 2.3. We preserve the accuracy in this reduction by applying the EETs to $\mathcal{BD}(A)$ as described in section 4. We obtain the decomposition $\mathcal{BD}(T)$ of a tridiagonal matrix T , which has only three nontrivial factors:

$$T = L^{(n-1)} \cdot D \cdot U^{(n-1)}.$$

The matrix T need not be symmetric and may not have a full set of eigenvectors. It does, however, have the same eigenvalues as the tridiagonal matrix $\bar{T} \equiv \bar{L}^{(n-1)} D \bar{U}^{(n-1)}$, where

$$\bar{l}_i^{(n-1)} = \bar{u}_i^{(n-1)} \equiv \sqrt{l_i^{(n-1)} u_i^{(n-1)}}, \quad i = 1, 2, \dots, n-1.$$

($\bar{T}$ and T have the same characteristic polynomial; they are, however, only similar if $\bar{l}_i^{(n-1)} = 0$ whenever $u_i^{(n-1)} = 0$ and vice versa.) The eigenvalues of $\bar{T}$ are computed accurately as the squares of the singular values of its Cholesky factor $C = D^{1/2} \bar{U}^{(n-1)}$

using the LAPACK [1] routine **DLASQ1** [12, 17]. (C is accurately obtained from $\mathcal{BD}(T)$ at the cost of $2n - 2$ multiplications and $2n - 1$ square roots.)

The matrix C is thus formed without performing any subtractions at the cost of at most

$$2 \sum_{i=1}^{n-2} \sum_{j=i+2}^n (4n + 2j + 8) + 4n - 3 = \frac{16}{3}n^3 + O(n^2)$$

arithmetic operations. The routine **DLASQ1** costs an additional $O(n^2)$ operations; thus the cost of Algorithm 5.1 is at most $\frac{16}{3}n^3 + O(n^2)$.

In section 7 we prove that all eigenvalues of A are computed accurately independent of the angle between the left and right eigenvectors.

The eigenvectors (when they all exist) may certainly be accumulated, although questions remain about their accuracy in the nonsymmetric case—see section 7.

ALGORITHM 5.1 (accurate eigenvalues of a TN matrix). $B = \mathcal{BD}(A)$.

Input: A real or complex $n \times n$ TN matrix A with $n \geq 1$.

```

function TNEigenvalues(B)
    n = size(B,1)
    for i = 1 : n - 2
        for j = n : -1 : i + 2
            x = B(j,i)
            B(j,i) = 0
            B = TNAddToPrevious(B, x, 1, j)
            x = B(i,j)
            B(i,j) = 0
            B = (TNAddToPrevious(B^T, x, 1, j))^T
        end
    end
    C = diag(sqrt(diag(B)))
    for i = 1 : n - 1
        C(i, i + 1) = sqrt(B(i, i + 1) * B(i + 1, i) * B(i, i))
    end
    TNEigenvalues = (DLASQ1(C))^2
end

```

6. The singular value problem. In this section we present our algorithm for computing accurate singular values of a TN matrix A , given $\mathcal{BD}(A)$.

We first perform an implicit Golub–Kahan bidiagonalization on A by starting with $\mathcal{BD}(A)$ and performing Givens rotations as described in section 4.3. We implicitly introduce zeros in the first column of A below the main diagonal, then in the first row to the right of the (1,2) entry. We then proceed by induction until $\mathcal{BD}(A)$ is reduced to $\mathcal{BD}(F)$, where F is bidiagonal. The decomposition $\mathcal{BD}(F)$ contains only two nontrivial factors: D and $U^{(n-1)}$, so we recover F from $\mathcal{BD}(F)$ at the cost of $n - 1$ multiplications. The computation so far is subtraction-free. Finally, we compute the singular values of F accurately using the LAPACK [1] routine **DLASQ1**.

By performing complexity analysis similar to that of section 5, we conclude that Algorithm 6.1 costs at most $\frac{16}{3}n^3 + O(n^2)$ arithmetic operations for the reduction to bidiagonal form F and $O(n^2)$ operations for **DLASQ1**.

The singular vectors can be computed by accumulating the Givens rotations on the left and the right with the singular vector matrices of F .

In section 7 we prove that the computed singular values and singular vectors are accurate.

ALGORITHM 6.1 (accurate singular values of a TN matrix). $\mathbf{v}_i, \mathbf{u}_i, B = \mathcal{BD}(A), \mathbf{v}_i, \mathbf{u}_i, A$

```

function TNSingularValues(B)
    n = size(B, 1)
    for i = 1 : n - 1
        for j = n : -1 : i + 1
            x = B(j, i)
            B(j, i) = 0
            c = sqrt(1 + x^2)
            B = (TNAddToPrevious(B^T, x/c, c, j))^T
        end
        for j = n : -1 : i + 2
            x = B(i, j)
            B(i, j) = 0
            c = sqrt(1 + x^2)
            B = TNAddToPrevious(B, x/c, c, j)
        end
    end
    F = diag(diag(B))
    for i = 1 : n - 1
        F(i, i + 1) = F(i, i) * B(i, i + 1)
    end
    TNSingularValues = DLASQ1(F)
    
```

7. Perturbation theory and error analysis. In this section we prove that if a TN matrix A is represented as a product of nonnegative bidiagonal matrices, then small relative perturbations in the entries of the bidiagonal factors cause only small relative perturbations in the eigenvalues and the SVD of A .

In particular, $\mathcal{BD}(A)$ determines the eigenvalues and the SVD of A accurately, which justifies computing them accurately. We believe that the perturbation result for the eigenvalues is new; the result for the SVD is known (see section 2.5 and Demmel et al. [11, section 9]), but we improve the perturbation bounds for the singular values.

We then use these perturbation results to prove that Algorithms 5.1 and 6.1 compute the eigenvalues and the singular values of a TN matrix accurately in floating point arithmetic; the singular vectors that may be accumulated by Algorithm 6.1 will also be accurate in the appropriate sense. The eigenvector problem remains open in the nonsymmetric case.

We finish the section with the perturbation theory and error analysis for the eigenvalue and singular value problems for TP Vandermonde and TP Cauchy matrices.

For our error analysis we use the standard model of floating point arithmetic [29, section 2.2]:

$$fl(x \odot y) = (x \odot y)(1 + \delta)^{\pm 1}, \quad \text{where } |\delta| \leq \epsilon, \quad \text{and } \odot \in \{+, -, \times, /\}.$$

We will accumulate relative errors or relative perturbations in the style of Higham [29]: If $|\delta_i| \leq \delta \ll 1$ and $\rho_i = \pm 1$, then

$$\left| \prod_{i=1}^k (1 + \delta_i)^{\rho_i} - 1 \right| \leq \frac{k\delta}{1 - k\delta}.$$

Also, if $|\delta_1| \leq m\delta/(1 - m\delta)$ and $|\delta_2| \leq k\delta/(1 - k\delta)$, then

$$|(1 + \delta_1)(1 + \delta_2) - 1| \leq \frac{(m + k)\delta}{1 - (m + k)\delta}.$$

Eigenvalues and singular values. We will extract information about the perturbation of the eigenvalues and the singular values of a TN matrix A from its k th compound or k th order matrix $\mathcal{A}^{(k)}$ [20, 21]. The entries of $\mathcal{A}^{(k)}$ are all k th order minors of A . The eigenvalues of $\mathcal{A}^{(k)}$ are all possible products of k eigenvalues of A [21, Thm. 23, p. 65]. We will now prove that the same is true for the singular values.

LEMMA 7.1. *Let $\sigma_1 \geq \sigma_2 \geq \dots \geq \sigma_n \geq 0$ be the singular values of A . Then the singular values of $\mathcal{A}^{(k)}$ are $\sigma_1 \sigma_2 \dots \sigma_n$ taken k at a time.*

Proof. Let $A = USV^T$ be the SVD of A , where $S = \text{diag}(\sigma_1, \sigma_2, \dots, \sigma_n)$. The k th compound matrix of a product of matrices equals the product of the k th compound matrices of the factors [21, p. 64]. Therefore $\mathcal{U}^{(k)}$ and $\mathcal{V}^{(k)}$ are orthogonal ($U^T U = I$ implies $(\mathcal{U}^{(k)})^T \mathcal{U}^{(k)} = \mathcal{I}^{(k)}$), and the SVD of $\mathcal{A}^{(k)}$ is

$$\mathcal{A}^{(k)} = \mathcal{U}^{(k)} \mathcal{S}^{(k)} (\mathcal{V}^{(k)})^T.$$

Finally, the matrix $\mathcal{S}^{(k)}$ is diagonal, and its diagonal entries are all possible products of the numbers $\sigma_1, \sigma_2, \dots, \sigma_n$ taken k at a time. $\square$

THEOREM 7.2. *Let $B_1, B_2, \dots, B_s$ be $n \times n$ matrices with $B_r = x(1 + \delta)B_r$, $1 \leq r \leq s$, and $|\delta| \ll 1$. Let $A = B_1 B_2 \dots B_s$ and $\hat{A} = x(1 + \delta)A$. Then the singular values of A and $\hat{A}$ satisfy $\lambda_1 \geq \lambda_2 \geq \dots \geq \lambda_n$ and $\hat{\lambda}_1 \geq \hat{\lambda}_2 \geq \dots \geq \hat{\lambda}_n$ with $\lambda_k \geq \hat{\lambda}_k \geq \lambda_k(1 - 2|\delta|)$ for $k = 1, 2, \dots, n$.*

$$|\hat{\lambda}_k - \lambda_k| \leq \frac{2|\delta|}{1 - 2|\delta|} \lambda_k, \quad |\hat{\sigma}_k - \sigma_k| \leq \frac{2|\delta|}{1 - 2|\delta|} \sigma_k.$$

Proof. The values of s and r are of no importance in this theorem; we introduce them only for clarity. Also, any B_j may be upper or lower bidiagonal, or simply diagonal.

The matrix A is TN since it is a product of TN bidiagonal matrices. We will first prove the theorem by assuming that A is TP and then extend the result to the TN case by continuity.

Since $A = B_1 B_2 \dots B_s$, the Cauchy–Binet identity [20, p. 9] implies that any entry $a_{ij}^{(k)}$ of $\mathcal{A}^{(k)}$ is a linear function of x : $a_{ij}^{(k)} = ax + b$ for some $a, b \geq 0$. Similarly, $\hat{a}_{ij}^{(k)} = ax(1 + \delta) + b$. Then for all i, j

$$(7.1) \quad \left| \hat{a}_{ij}^{(k)} - a_{ij}^{(k)} \right| = |\delta|ax \leq |\delta|a_{ij}^{(k)}.$$

The matrix $\mathcal{A}^{(k)}$ is positive with Perron root $\lambda_1 \lambda_2 \dots \lambda_k$ and two-norm $\sigma_1 \sigma_2 \dots \sigma_k$. Since small perturbations in the entries of a positive matrix cause small perturbations in its Perron root [15, Thm. 1] and two-norm [9, Cor. 5.1] the inequality (7.1) implies

$$(7.2) \quad \hat{\lambda}_1 \hat{\lambda}_2 \dots \hat{\lambda}_k = (1 + \delta_1) \lambda_1 \lambda_2 \dots \lambda_k, \quad \hat{\sigma}_1 \hat{\sigma}_2 \dots \hat{\sigma}_k = (1 + \delta_2) \sigma_1 \sigma_2 \dots \sigma_k,$$

and by analogy

(7.3)

$$\hat{\lambda}_1 \hat{\lambda}_2 \cdots \hat{\lambda}_{k-1} = (1 + \delta_3) \lambda_1 \lambda_2 \cdots \lambda_{k-1}, \quad \hat{\sigma}_1 \hat{\sigma}_2 \cdots \hat{\sigma}_{k-1} = (1 + \delta_4) \sigma_1 \sigma_2 \cdots \sigma_{k-1},$$

where $|\delta_i| \leq \delta$ for $i = 1, 2, 3, 4$. The claim now follows directly by substituting (7.3) into (7.2) and using that $|(1 + \delta_1)/(1 + \delta_3) - 1| \leq 2|\delta|/(1 - 2|\delta|)$.

Let A and $\hat{A}$ now be TN. Let the TP matrices A_θ and $\hat{A}_\theta$ be obtained from A and $\hat{A}$ by replacing the zero entries in $\mathcal{BD}(A)$ and $\mathcal{BD}(\hat{A})$, respectively, by a tiny quantity $\theta > 0$. Since the eigenvalues of a matrix are continuous functions of the matrix entries, by taking the limit as $\theta \rightarrow 0$ we conclude that the claim of this theorem is true when A is TN as well.

We considered the TP and TN cases separately because we needed $\mathcal{A}^{(k)}$ to be nonnegative and, in order to invoke Theorem 1 from [15]. $\square$

If k entries in the factors of the product $A = B_1 \cdots B_s$ are perturbed and the relative perturbation in each entry is bounded by $0 < \delta \ll 1$, then we can apply Theorem 7.2 repeatedly and accumulate the relative perturbations to conclude that the relative perturbation in each eigenvalue and each singular value of the perturbed product is bounded by $2k\delta/(1 - 2k\delta)$.

COROLLARY 7.3. Let $A, \hat{A}$ be TP matrices with $\mathcal{BD}(A)_{ij} = \lambda_i \delta_{ij}$ and $\mathcal{BD}(\hat{A})_{ij} = \sigma_i \delta_{ij}$, $i = 1, 2, \dots, n$. If $|\mathcal{BD}(\hat{A})_{ij} - \mathcal{BD}(A)_{ij}| \leq \delta(\mathcal{BD}(\hat{A}))_{ij}$ for all i, j and $0 < \delta \ll 1$, then

$$|\hat{\lambda}_i - \lambda_i| \leq \frac{2n^2\delta}{1 - 2n^2\delta} \lambda_i, \quad |\hat{\sigma}_i - \sigma_i| \leq \frac{2n^2\delta}{1 - 2n^2\delta} \sigma_i.$$

We will now use Theorem 7.2 for the error analysis of our eigenvalue and singular value algorithms, Algorithms 5.1 and 6.1. These algorithms reduce the eigenvalue and the singular value problem of a TN matrix to the singular value problem of a bidiagonal matrix. In this reduction the only source of error is in computing the entries of a certain representation of an intermediate TN matrix A as a product of nonnegative bidiagonal matrices $A = B_1 \cdots B_s$ (e.g., any of the representations (4.2) to (4.7)). All arithmetic is subtraction-free; thus every single floating point operation will cause at most ϵ relative perturbation in at most one entry of some B_j . In turn, the relative perturbation of any eigenvalue and singular value of A will be at most $2\epsilon/(1 - 2\epsilon)$. Accumulating the error in each eigenvalue and singular value in the style of Higham as described above, we conclude that the relative perturbation in each eigenvalue and singular value of A after at most $\frac{16}{3}n^3 + O(n^2)$ such arithmetic operations (which is a bound on the cost of reduction to the bidiagonal singular value problem for both algorithms) will not exceed $(\frac{32}{3}n^3 + O(n^2))\epsilon/(1 - (\frac{32}{3}n^3 + O(n^2))\epsilon)$.

COROLLARY 7.4. Let A be a TN matrix with $\mathcal{BD}(A)_{ij} = \lambda_i \delta_{ij}$ and $\mathcal{BD}(\hat{A})_{ij} = \sigma_i \delta_{ij}$, $i = 1, 2, \dots, n$. If $|\mathcal{BD}(\hat{A})_{ij} - \mathcal{BD}(A)_{ij}| \leq \epsilon(\mathcal{BD}(\hat{A}))_{ij}$ for all i, j and $0 < \epsilon \ll 1$, then

$$|\lambda_i - \tau_i^2| \leq \frac{(\frac{32}{3}n^3 + O(n^2))\epsilon}{1 - (\frac{32}{3}n^3 + O(n^2))\epsilon} \lambda_i, \quad |\sigma_i - \theta_i| \leq \frac{(\frac{32}{3}n^3 + O(n^2))\epsilon}{1 - (\frac{32}{3}n^3 + O(n^2))\epsilon} \sigma_i.$$

The final step of computing the singular values of the bidiagonal matrix using DLASQ1 is known to introduce only a small additional relative error [17, 36].

COROLLARY 7.5.

Let A be a TN matrix of size n with singular values $\sigma_1 \geq \sigma_2 \geq \dots \geq \sigma_n$ and let ϵ be the relative error in the singular values of the bidiagonal matrix computed by DLASQ1. Then the singular values $\hat{\sigma}_i$ computed by Algorithm 6.1 satisfy

$$|\hat{\sigma}_i - \sigma_i| \leq \frac{32}{3} n^3 + O(n^2) \epsilon.$$

Singular vectors. The singular vectors of a TN matrix A are accurately determined by $\mathcal{BD}(A)$ and may be computed accurately by accumulating all Givens rotations in Algorithm 6.1.

PROPOSITION 7.6. Let $A = U\Sigma V^T$ be a TN matrix of size n with singular values $\sigma_1 \geq \sigma_2 \geq \dots \geq \sigma_n$ and let $\hat{U}, \hat{\Sigma}, \hat{V}$ be the matrices computed by Algorithm 6.1. Then the singular vectors u_i and $\hat{u}_i$ satisfy

the bound

$$\sin \theta(u_i, \hat{u}_i) \leq \frac{O(n^3 \kappa \epsilon)}{\min_{j \neq i} |(\sigma_j - \sigma_i)/\sigma_i|},$$

$$(7.4) \quad \sin \theta(u_i, \hat{u}_i) \leq \frac{O(n^3 \kappa \epsilon)}{\min_{j \neq i} |(\sigma_j - \sigma_i)/\sigma_i|},$$

where κ is the condition number of A , L is the maximum row sum of U , and D is the maximum column sum of V . The modest constant hidden in the above big- O notation can be found in [11].

The modest constant hidden in the above big- O notation can be found in [11].

We conjecture that a slightly stronger result is true, namely we can omit the factor κ from the bound (7.4).

The result follows directly from [11] (see also the discussion in section 2.5). First consider the effect of a single δ relative perturbation in any one entry of $\mathcal{BD}(A)$. As in the proof of Theorem 7.2, from the Cauchy–Binet identity we know that the relative perturbation in any minor of A will be at most δ . The relative perturbation in any entry of L, D , or U will be at most 2δ , since every entry of L, D , and U is a quotient of minors of A . Following [11] we obtain the bound

$$(7.5) \quad \sin \theta(u_i, \hat{u}_i) \leq \frac{O(\kappa \delta)}{\min_{j \neq i} |(\sigma_j - \sigma_i)/\sigma_i|}.$$

Accumulating the perturbations in Algorithm 6.1 using (7.5) and using that for small acute angles $\sin x + \sin y \approx \sin(x + y)$, we obtain (7.4). $\square$

Eigenvectors. In the symmetric case, the eigenvector matrix of a TN matrix A may be accumulated in Algorithm 5.1 as a product of highly accurate EETs, a diagonal matrix, and the right singular vector matrix of a bidiagonal matrix. Alternatively the eigenvector matrix may be computed using Algorithm 6.1 as the right singular vector matrix of A . Either way, the computed eigenvectors will satisfy the error bound (7.4).

In the nonsymmetric case, however, we know of no relative perturbation results for eigenvectors of nonsymmetric TN matrices, and we make no claims as to the accuracy of the eigenvectors of a nonsymmetric TN matrix that may be computed using Algorithm 5.1.

A nonsymmetric TN matrix may not even have a complete set of eigenvectors. For example a Jordan block corresponding to a positive eigenvalue is TN. If a nonsymmetric TN matrix is irreducible, then it is diagonalizable, and a full set of eigenvectors exists. Although the factors of the eigenvector matrix—EETs, a diagonal matrix, and

the right singular vector of a bidiagonal matrix—are very accurate in the appropriate sense, forming the product will likely involve subtractive cancellation and loss of accuracy in the individual eigenvector components.

The eigenvector matrix of an irreducible TN matrix A is a γ -TN (i.e., it has an LU decomposition where L and U^{-1} are TN [24]), and the j th eigenvector has $j - 1$ sign changes [21, Thm. 6, p. 87].

Finding algorithms that will guarantee accuracy in the computed eigenvectors and/or the two properties above is still an open problem and a topic of current research.

Vandermonde and Cauchy matrices. We conclude this section with a result about the accuracy and stability of the eigenvalues and singular values of TP Vandermonde and TP Cauchy matrices

$$V(x) = \left[x_i^{j-1} \right]_{i,j=1}^n \quad \text{and} \quad C(x, y) = \left[\frac{1}{x_i + y_j} \right]_{i,j=1}^n,$$

where $0 < x_1 < \dots < x_n$, and $x_1 < \dots < x_n$, $y_1 < \dots < y_n$, and $x_1 + y_1 > 0$, respectively. Define

$$\begin{aligned} \text{rel\_gap}_x &\equiv \min_{i \neq j} \frac{|x_i - x_j|}{|x_i| + |x_j|}, \\ \text{rel\_gap}_{xy} &\equiv \min_{i,j} \frac{|x_i - y_j|}{|x_i| + |y_j|}, \\ \kappa_V &\equiv \frac{1}{\text{rel\_gap}_x}, \\ \kappa_C &\equiv \frac{1}{\text{rel\_gap}_x} + \frac{1}{\text{rel\_gap}_y} + \frac{1}{\text{rel\_gap}_{xy}}. \end{aligned}$$

PROPOSITION 7.7. *If $V = V(x)$, $\hat{V} = V(\hat{x})$, $C = C(x, y)$, $\hat{C} = C(\hat{x}, \hat{y})$, $\lambda_i, \mu_i, \hat{\lambda}_i, \hat{\mu}_i$ are the eigenvalues of $V, C, \hat{V}, \hat{C}$, respectively, $\hat{x}_i = x_i(1 + \delta_i)$, $\hat{y}_i = y_i(1 + \eta_i)$, and $|\delta_i|, |\eta_i| \leq \delta \ll 1$, then*

$$(7.6) \quad |\hat{\lambda}_i - \lambda_i| \leq O(n^3 \kappa_V \delta) \lambda_i, \quad |\hat{\mu}_i - \mu_i| \leq O(n^3 \kappa_C \delta) \mu_i.$$

Proof. Let λ be an eigenvalue of V with eigenvector x , and $\hat{\lambda}$ be an eigenvalue of $\hat{V}$ with eigenvector $\hat{x}$. Then λ is an eigenvalue of C with eigenvector x , and $\hat{\lambda}$ is an eigenvalue of $\hat{C}$ with eigenvector $\hat{x}$. Let μ be an eigenvalue of C with eigenvector y , and $\hat{\mu}$ be an eigenvalue of $\hat{C}$ with eigenvector $\hat{y}$. Then μ is an eigenvalue of V with eigenvector y , and $\hat{\mu}$ is an eigenvalue of $\hat{V}$ with eigenvector $\hat{y}$.

It suffices to prove the result for the eigenvalues of C , the rest being analogous. We have

$$|(\hat{x}_i - \hat{x}_j) - (x_i - x_j)| = |\delta_i x_i - \delta_j x_j| \leq \delta |x_i - x_j| / \text{rel\_gap}_x.$$

Similarly

$$|(\hat{x}_i - \hat{y}_j) - (x_i - y_j)| = |\delta_i x_i - \eta_j y_j| \leq \delta |x_i - y_j| / \text{rel\_gap}_{xy}.$$

Accumulating the perturbations in $\mathcal{BD}(C)$ and $\mathcal{BD}(\hat{C})$ using the formulas (3.6) we obtain

$$|(\mathcal{BD}(\hat{C}))_{ij} - (\mathcal{BD}(C))_{ij}| \leq \frac{n \kappa_C \delta}{1 - n \kappa_C \delta} \mathcal{BD}(C)_{ij}.$$

Corollary 7.3 now implies (7.6). $\square$

PROPOSITION 7.8.

Let V and C be $n \times n$ matrices with eigenvalues λ_i and μ_i and singular values $\hat{\lambda}_i$ and $\hat{\mu}_i$, respectively. Let $\mathcal{BD}(V)$ and $\mathcal{BD}(C)$ be the bidiagonal decompositions of V and C computed using formulas (3.5) and (3.6) with ϵ relative perturbation. Then

$$(7.7) \quad |\hat{\lambda}_i - \lambda_i| \leq \frac{(12n^3 + O(n^2))\epsilon}{1 - (12n^3 + O(n^2))\epsilon} \lambda_i, \quad |\hat{\mu}_i - \mu_i| \leq \frac{(18n^3 + O(n^2))\epsilon}{1 - (18n^3 + O(n^2))\epsilon} \mu_i.$$

The decompositions $\mathcal{BD}(V)$ and $\mathcal{BD}(C)$ can be computed using formulas (3.5) and (3.6) in $\frac{2}{3}n^3 + O(n^2)$ and $\frac{8}{3}n^3 + O(n^2)$ arithmetic operations, respectively. Every floating point operation in evaluating (3.5) or (3.6) can cause at most ϵ relative perturbation in an entry of $\mathcal{BD}(V)$ or $\mathcal{BD}(C)$, respectively. By applying Theorem 7.2 repeatedly, we conclude that the relative perturbations in the eigenvalues and/or singular values of V and C resulting from rounding errors in just computing $\mathcal{BD}(V)$ and $\mathcal{BD}(C)$ will not exceed

$$(7.8) \quad \frac{(\frac{4}{3}n^3 + O(n^2))\epsilon}{1 - (\frac{4}{3}n^3 + O(n^2))\epsilon} \quad \text{and} \quad \frac{(\frac{16}{3}n^3 + O(n^2))\epsilon}{1 - (\frac{16}{3}n^3 + O(n^2))\epsilon},$$

respectively. According to Corollary 7.5, Algorithms 5.1 and 6.1 will introduce at most

$$(7.9) \quad \frac{(\frac{32}{3}n^3 + O(n^2))\epsilon}{1 - (\frac{32}{3}n^3 + O(n^2))\epsilon}$$

additional relative error in each eigenvalue and singular value of V and C . Combining (7.8) and (7.9), we obtain (7.7). $\square$

The n^3 factor in the error bounds above is only the worst case error bound, which does not appear in numerical experiments we conducted.

8. Numerical results. We performed extensive numerical tests and confirmed the correctness and cost of our algorithms. To confirm the correctness we chose various TN structured matrices (Cauchy, Vandermonde, generalized Vandermonde, etc.) with random nodes and compared the output to the output of other $O(n^3)$ accurate algorithms (for the SVD [10, 11]) and to the output from Mathematica [40] in extended precision arithmetic. Most test matrices had condition numbers well in excess of 10^{16} , so conventional algorithms failed to get any correct digits in the smallest singular values and eigenvalues, whereas Algorithms 5.1 and 6.1 always agreed with the output from Mathematica to at least 14 digits in all singular values and eigenvalues, even the tiniest ones. We present a couple of our test results here.

The Hilbert matrix. In our first numerical example we computed the singular values of the 20×20 Hilbert matrix

$$H = \left[\frac{1}{i+j-1} \right]_{i,j=1}^{20},$$

which is also Cauchy and its bidiagonal decomposition is easily and accurately computed in $O(n^3)$ time using (3.6). The Hilbert matrix is an attractive example because it is a very ill-conditioned TN matrix for which other accurate $O(n^3)$ SVD algorithms exist [10, Alg. 2], which we can compare against.

We computed the singular values of H using the following algorithms:

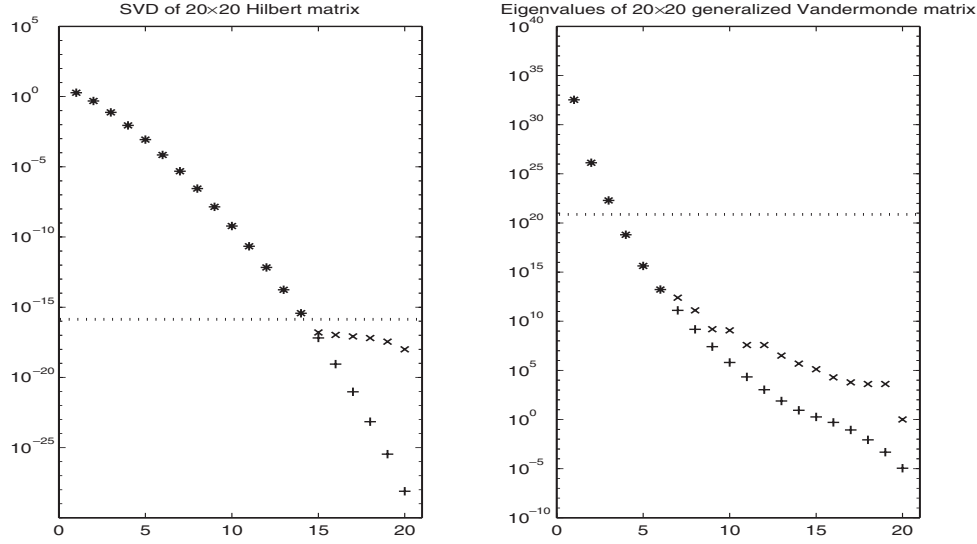


FIG. 8.1. Plots of the singular values of the 20×20 Hilbert matrix and the eigenvalues of a 20×20 generalized Vandermonde matrix; “+” = accurate, “x” = traditional. Data below the dotted lines may be inaccurate for the traditional algorithms.

- Mathematica [40] with 60-digit arithmetic;
- Algorithm 6.1;
- Algorithm 2 from [10];
- traditional SVD algorithm as implemented in MATLAB’s function `svd`.

Since $\kappa(H) \approx 10^{30}$, Mathematica with 60-digit arithmetic computed all singular values with about 30 significant digits of accuracy. Algorithm 2 from [10] guarantees high relative accuracy (close to 16 digits since we are using double precision) in the computed singular values, and its output agreed with the output from Algorithm 6.1 and the output from Mathematica to at least 14 decimal places. In contrast, the traditional algorithm—MATLAB’s function `svd`—returned only the largest singular values accurately with the accuracy gradually decreasing until the singular values smaller than $O(\epsilon)\sigma_1 \approx 10^{-16}$ were computed with no correct digits at all. All these results are in the left plot of Figure 8.1.

Generalized Vandermonde matrix. Next, consider the eigenvalue problem for a 20×20 generalized Vandermonde matrix

$$G = \left[x_i^{j-1+\lambda_{n-j+1}} \right]_{i,j=1}^{20}$$

with nodes $x_i = i$, $i = 1, 2, \dots, 20$, and corresponding to partition $\lambda = (6, 3, 2, 1)$. This Vandermonde matrix is generalized in the sense that the exponents increase nonlinearly—the i th row of G is $(1, x_i, x_i^2, \dots, x_i^{15}, x_i^{17}, x_i^{19}, x_i^{21}, x_i^{25})$. It is typical for the generalized Vandermonde matrices to be described in terms of corresponding partitions λ [13, 32].

We chose this example of a generalized Vandermonde matrix because it incorporates both our technique for computing an accurate bidiagonal decomposition of such a matrix [13] and the eigenvalue algorithm described in this paper. The bidiagonal decomposition of the $\mathbf{v}_i, \dots, \mathbf{v}_n$ G^{-1} is computable accurately and efficiently using a

Björck–Pereyra method [13], and we permute the elements in the $L^{(i)}$ and $U^{(i)}$ and invert D in the bidiagonal decomposition of G^{-1} in order to obtain $\mathcal{BD}(G)$ [25]. As a benchmark we again used Mathematica. Since G is nonsymmetric, the accuracy of the computed eigenvalues is also affected by the angle between the left (y_i) and right (x_i) eigenvectors. The relative error in each eigenvalue λ_i is bounded by

$$\text{macheps} \cdot \frac{\|A\|_2}{\lambda_i |y_i^T x_i|},$$

where `macheps` is the machine precision in Mathematica. Running Mathematica’s `Eigenvalues` routine in 60-digit arithmetic (`macheps` = 10^{-60}) ensured that each eigenvalue is computed with at least 16 correct leading digits (because $\lambda_{\min} \approx 10^{-5}$, $\min_i(\lambda_i |y_i^T x_i|) \approx 10^{-7}$, and $\|A\|_2 \approx 10^{32}$). The results from Algorithm 5.1 agreed with the results from Mathematica to at least 14 decimal digits, even in the tiniest eigenvalues. We also compared our results to the results of the traditional eigenvalue algorithm QR as implemented by MATLAB’s routine `eig`. As expected only the largest eigenvalues are computed accurately. MATLAB also returned some negative eigenvalues and eigenvalues with nonzero imaginary part, so in Figure 8.1 we have plotted only their magnitudes.

9. Conclusions and open problems. We have presented new $O(n^3)$ algorithms for computing accurate eigenvalues and SVD of a TN matrix A , given its bidiagonal decomposition $\mathcal{BD}(A)$. With this we settle an open question from [11] not only for the singular value problem but for the eigenvalue problem as well. To the best of our knowledge Algorithm 5.1 is the first algorithm to compute all eigenvalues of a nonsymmetric matrix to high relative accuracy. The question of computing accurate eigenvectors of nonsymmetric TN matrices remains open.

We excluded the class of singular TN matrices because they no longer possess a unique bidiagonal decomposition; the requirement for nonsingularity in Theorem 2.1 is important. The problem of finding accurate eigenvalue and SVD algorithms for singular TN matrices also remains open.

Acknowledgments. I thank James Demmel for suggesting this problem to me and for many helpful discussions in the process of preparing this work; Froilán Dopico and Juan Manuel Peña for making useful suggestions, which lead to improvements in the presentation; and J. Martínez for pointing out the relevance of [33].

REFERENCES

- [1] E. ANDERSON, Z. BAI, C. BISCHOF, S. BLACKFORD, J. DEMMEL, J. DONGARRA, J. DU CROZ, A. GREENBAUM, S. HAMMARLING, A. MCKENNEY, AND D. SORENSSEN, *LAPACK Users’ Guide*, 3rd ed., Software Environ. Tools 9, SIAM, Philadelphia, 1999.
- [2] T. ANDO, *Totally positive matrices*, Linear Algebra Appl., 90 (1987), pp. 165–219.
- [3] A. BERENSTEIN, S. FOMIN, AND A. ZELEVINSKY, *Parametrizations of canonical bases and totally positive matrices*, Adv. Math., 122 (1996), pp. 49–149.
- [4] Å. BJÖRCK AND V. PEREYRA, *Solution of Vandermonde systems of equations*, Math. Comp., 24 (1970), pp. 893–903.
- [5] T. BOROS, T. KAILATH, AND V. OLSHEVSKY, *A fast parallel Björck–Pereyra-type algorithm for solving Cauchy linear equations*, Linear Algebra Appl., 302/303 (1999), pp. 265–293.
- [6] F. BRENTI, *Combinatorics and total positivity*, J. Combin. Theory Ser. A, 71 (1995), pp. 175–218.
- [7] F. BRENTI, *The applications of total positivity to combinatorics, and conversely*, in Total Positivity and Its Applications, Math. Appl. 359, Kluwer Academic Publishers, Dordrecht, The Netherlands, 1996, pp. 451–473.
- [8] Y. CHEN, *Singular values in electrical impedance tomography*, private communication, 2003.

- [9] J. DEMMEL, *Applied Numerical Linear Algebra*, SIAM, Philadelphia, 1997.
- [10] J. DEMMEL, *Accurate singular value decompositions of structured matrices*, SIAM J. Matrix Anal. Appl., 21 (1999), pp. 562–580.
- [11] J. DEMMEL, M. GU, S. EISENSTAT, I. SLAPNIČAR, K. VESELIĆ, AND Z. DRMAČ, *Computing the singular value decomposition with high relative accuracy*, Linear Algebra Appl., 299 (1999), pp. 21–80.
- [12] J. DEMMEL AND W. KAHAN, *Accurate singular values of bidiagonal matrices*, SIAM J. Sci. Statist. Comput., 11 (1990), pp. 873–912.
- [13] J. DEMMEL AND P. KOEV, *The accurate and efficient solution of a totally positive generalized Vandermonde linear system*, SIAM J. Matrix Anal. Appl., to appear.
- [14] I. S. DHILLON, *A New $O(n^2)$ Algorithm for the Symmetric Tridiagonal Eigenvalue/Eigenvector Problem*, Ph.D. thesis, University of California, Berkeley, CA, 1997.
- [15] L. ELSNER, I. KOLTRACHT, M. NEUMANN, AND D. XIAO, *On accurate computations of the Perron root*, SIAM J. Matrix Anal. Appl., 14 (1993), pp. 456–467.
- [16] S. M. FALLAT, *Bidiagonal factorizations of totally nonnegative matrices*, Amer. Math. Monthly, 108 (2001), pp. 697–712.
- [17] K. FERNANDO AND B. PARLETT, *Accurate singular values and differential qd algorithms*, Numer. Math., 67 (1994), pp. 191–229.
- [18] S. FOMIN AND A. ZELEVINSKY, *Double Bruhat cells and total positivity*, J. Amer. Math. Soc., 12 (1999), pp. 335–380.
- [19] S. FOMIN AND A. ZELEVINSKY, *Total positivity: Tests and parametrizations*, Math. Intelligencer, 22 (2000), pp. 23–33.
- [20] F. GANTMACHER, *The Theory of Matrices*, AMS Chelsea, Providence, RI, 1998.
- [21] F. GANTMACHER AND M. KREIN, *Oscillation Matrices and Kernels and Small Vibrations of Mechanical Systems*, revised edition, AMS Chelsea, Providence, RI, 2002.
- [22] M. GASCA AND C. A. MICCHELLI, EDS., *Total Positivity and Its Applications*, Math. Appl. 359, Kluwer Academic Publishers, Dordrecht, The Netherlands, 1996.
- [23] M. GASCA AND J. M. PEÑA, *Total positivity and Neville elimination*, Linear Algebra Appl., 165 (1992), pp. 25–44.
- [24] M. GASCA AND J. M. PEÑA, *Total positivity, QR factorization, and Neville elimination*, SIAM J. Matrix Anal. Appl., 14 (1993), pp. 1132–1140.
- [25] M. GASCA AND J. M. PEÑA, *On factorizations of totally positive matrices*, in Total Positivity and Its Applications, Kluwer Academic Publishers, Dordrecht, The Netherlands, 1996, pp. 109–130.
- [26] G. A. GEIST, G. W. HOWELL, AND D. S. WATKINS, *The BR eigenvalue algorithm*, SIAM J. Matrix Anal. Appl., 20 (1999), pp. 1083–1098.
- [27] N. J. HIGHAM, *Error analysis of the Björck-Pereyra algorithms for solving Vandermonde systems*, Numer. Math., 50 (1987), pp. 613–632.
- [28] N. J. HIGHAM, *Stability analysis of algorithms for solving confluent Vandermonde-like systems*, SIAM J. Matrix Anal. Appl., 11 (1990), pp. 23–41.
- [29] N. J. HIGHAM, *Accuracy and Stability of Numerical Algorithms*, 2nd ed., SIAM, Philadelphia, 2002.
- [30] W. KAHAN AND I. FARKAS, *Algorithms 167–169*, Comm. ACM, 6 (1963), pp. 164–165. See also the *Certification*, Comm. ACM, 6 (1963), p. 523.
- [31] S. KARLIN, *Total Positivity*, Vol. I, Stanford University Press, Stanford, CA, 1968.
- [32] I. G. MACDONALD, *Symmetric Functions and Hall Polynomials*, 2nd ed., Oxford University Press, New York, 1995.
- [33] J. J. MARTÍNEZ AND J. M. PEÑA, *Factorizations of Cauchy-Vandermonde matrices*, Linear Algebra Appl., 284 (1998), pp. 229–237.
- [34] J. J. MARTÍNEZ AND J. M. PEÑA, *Fast algorithms of Björck-Pereyra type for solving Cauchy-Vandermonde linear systems*, Appl. Numer. Math., 26 (1998), pp. 343–352.
- [35] THE MATHWORKS, INC., *MATLAB Reference Guide*, Natick, MA, 1992.
- [36] B. PARLETT, *The new qd algorithms*, in Acta Numerica, Cambridge University Press, Cambridge, UK, 1995, pp. 459–491.
- [37] J. W. RAINEY AND G. J. HABETLER, *Tridiagonalization of completely nonnegative matrices*, Math. Comp., 26 (1972), pp. 121–128.
- [38] A. M. WHITNEY, *A reduction theorem for totally positive matrices*, J. Analyse Math., 2 (1952), pp. 88–92.
- [39] J. H. WILKINSON, *The Algebraic Eigenvalue Problem*, Oxford University Press, Oxford, UK, 1965.
- [40] S. WOLFRAM, *Mathematica: A System for Doing Mathematics by Computer*, Addison-Wesley, Reading, MA, 1988.

ON OPTIMAL REORDERINGS OF SPARSE MATRICES FOR PARALLEL CHOLESKY FACTORIZATIONS*

WEN-YANG LIN[†] AND CHUEN-LIANG CHEN[‡]

Abstract. The height of the elimination tree has long acted as the only criterion for deriving a suitable fill-preserving sparse matrix ordering for parallel factorization. Although the deficiency in adopting height as the criterion for all circumstances was well recognized, no research has succeeded in alleviating this constraint. In this paper, we extend the unit-cost fill-preserving ordering into a generalized class that can adopt various aspects in parallel factorization, such as computation, communication, and algorithmic diversity. We recognize and show that if any cost function satisfies two mandatory properties, called the independence and conservation properties, a greedy ordering scheme then generates an optimal ordering with minimum completion cost. We also present an efficient implementation of the proposed ordering algorithm. Incorporating various techniques, the complexity can be improved from $O(n \log n + e)$ to $O(q \log q + \kappa)$, where n denotes the number of nodes, e the number of edges, q the number of maximal cliques, and κ the sum of all maximal clique sizes in the filled graph. Empirical results show that the proposed algorithm can significantly reduce the parallel factorization cost without sacrificing much in terms of time efficiency.

Key words. elimination tree, fill-preserving ordering, Jess–Kees algorithm, minimum completion cost, parallel Cholesky factorization, sparse matrix

AMS subject classifications. 65F05, 65F50, 65Y05, 68R10

DOI. 10.1137/S0895479803386354

1. Introduction. In this paper, we consider the problem of finding fill-preserving sparse matrix orderings for parallel factorization, which arises during the exploitation of parallelism in the direct solution of large sparse symmetric positive definite systems [10]. Given a large sparse symmetric and positive definite matrix A , we want to determine an ordering that is appropriate in terms of preserving the sparsity and minimizing the cost to perform its Cholesky factorization in parallel.

Jess and Kees [12] were the first to propose a fill-preserving ordering strategy for parallel factorization. Their method has the desirable property of minimizing the height of the elimination tree over the class of fill-preserving orderings. Some effective implementations of the Jess–Kees algorithm have been proposed [15], [23]. In [22], Liu used a tree rotation heuristic to find orderings that, though not optimal, could substantially reduce the height of the elimination tree. Aspvall and Heggernes [2] also proposed an efficient algorithm to generate orderings for a minimum height elimination tree. Their work was limited to interval graphs, a subclass of chordal graphs.

Thus far, all approaches to this problem have been devoted to minimizing the height of the elimination tree. This criterion is based on the unit-cost assumption for node elimination, which, though simple in nature, is improper for exhibiting various factors that affect the cost of parallel factorization, such as communication overhead and the algorithmic diversity of the Cholesky form. In [16], [17], we produced some

*Received by the editors March 17, 2003; accepted for publication (in revised form) by E. G. Ng November 16, 2004; published electronically June 22, 2005. A preliminary version of this paper appeared in the Proceedings of the 2000 IEEE International Parallel and Distributed Processing Symposium, pp. 799–805.

<http://www.siam.org/journals/simax/27-1/38635.html>

[†]Department of Computer Science and Information Engineering, National University of Kaohsiung, 700 Kaohsiung University Road, Nan-Tzu District, Kaohsiung 811, Taiwan (wylin@nuk.edu.tw).

[‡]Department of Computer Science and Information Engineering, National Taiwan University, Taipei 10764, Taiwan (clchen@csie.ntu.edu.tw).

examples to illustrate how the cost discrepancy may grow indefinitely; we also solved the problem that specifically adopts the operation count or communication message count for factoring a column/row to generate a parallel pivoting sequence with the property that it minimizes the critical completion cost for parallel factorization.

In this study, we extend the elimination tree height criterion to a more general class that is categorized as nodal cost functions satisfying two properties called the *independent* and *conservative* properties. In terms of such a class of ordering criteria, we propose a greedy algorithm to find the optimal orderings for parallel factorization. Empirical results show that, on the average, the proposed minimum completion cost ordering (MinCP) reduces the parallel completion cost by up to 25% more than the well-known minimum height ordering (Jess–Kees), and by up to 66% in the best case.

The rest of this paper is organized as follows. In section 2, we provide background material and introduce related notation. In section 3, we introduce the elimination-tree-based computation model on which all discussions are based. We then define the independent and conservative properties and specify the generalized class of ordering criteria. Section 4 describes our greedy algorithm for finding optimal fill-preserving ordering. The optimality proof and the implementation details are also discussed there. Section 5 presents the experimental results on our test set. Finally, the conclusions and future work are summarized in section 6.

2. Background.

2.1. Sparse Cholesky factorization. Consider a system of linear equations

$$Ax = b,$$

where A is an $n \times n$ symmetric positive definite matrix, b is a given vector, and x is the unknown vector to be solved. In the direct solution of such linear systems, A is usually first decomposed into LL^T , which is known as Cholesky factorization, where L is a lower triangular matrix. The solution vector x is then computed by solving two triangular systems $Ly = b$ and $L^T x = y$. When A is sparse, some initially zero entries in A may become nonzero in L , which are called *fill* or *new nonzeros*. In order to reduce time and storage requirements, only the nonzero positions of L are stored and operated on during sparse Cholesky factorization [6]. Hence, an ordering stage is typically applied first to reduce the fill in the factor L [7]. Finding an ordering that gives the least fill is *NP-hard* [28]. Two frequently used heuristics are *minimum height ordering* and *minimum completion cost ordering* [7].

From the study by Dongarra, Gustavson, and Karp [4], there are six algorithmic forms of Cholesky factorization obtained by permuting the three nested loops that execute the following single statement:

$$a_{ij} \leftarrow a_{ij} - l_{ik}l_{jk}.$$

Depending on which of the three indices is placed in the outermost loop, they can be classified into three basic forms [10]: *column-major*, *row-major*, and *block-major*.

Here, we consider the parallel versions of these three sparse Cholesky factorizations and a more complicated variant of sparse submatrix-Cholesky called *submatrix-Cholesky factorization* [5]. These four algorithms are described in Figures 1, 2, 3, and 4, respectively. In Figure 4, we let $\mathcal{F}^{(k)}$ be the frontal matrix associated with column k and $\overline{\mathcal{F}}^{(k)}$ the remaining frontal matrix after the removal of the first column. The symbols A_{*k} and L_{*k} denote column k of A and L , respectively.

```

for  $j = 1$  to  $n$  do
  for  $k = 1$  to  $j - 1$  with  $l_{jk} \neq 0$  do
    for  $i = j$  to  $n$  with  $l_{ik} \neq 0$  do
       $a_{ij} \leftarrow a_{ij} - l_{ik}l_{jk}$ ;
     $l_{jj} \leftarrow \sqrt{a_{jj}}$ ;
    for  $i = j + 1$  to  $n$  with  $a_{ij} \neq 0$  do
       $l_{ij} \leftarrow a_{ij}/l_{jj}$ ;
endfor

```

FIG. 1. *Sparse column-Cholesky factorization.*

```

for  $i = 1$  to  $n$  do
  for  $j = 1$  to  $i$  do
    for  $k = 1$  to  $j - 1$  with  $l_{ik}l_{jk} \neq 0$  do
       $a_{ij} \leftarrow a_{ij} - l_{ik}l_{jk}$ ;
    if  $i = j$  then  $l_{ii} \leftarrow \sqrt{a_{ii}}$ ;
    else if  $a_{ij} \neq 0$  then  $l_{ij} \leftarrow a_{ij}/l_{jj}$ ;
  endfor
endfor

```

FIG. 2. *Sparse row-Cholesky factorization.*

```

for  $k = 1$  to  $n$  do
   $l_{kk} \leftarrow \sqrt{a_{kk}}$ ;
  for  $j = k + 1$  to  $n$  with  $a_{jk} \neq 0$  do
     $l_{jk} \leftarrow a_{jk}/l_{kk}$ ;
  for  $j = k + 1$  to  $n$  with  $l_{jk} \neq 0$  do
    for  $i = j$  to  $n$  with  $l_{ik} \neq 0$  do
       $a_{ij} \leftarrow a_{ij} - l_{ik}l_{jk}$ ;
    endfor
  endfor

```

FIG. 3. *Sparse submatrix-Cholesky factorization.*

```

for  $k = 1$  to  $n$  do
  assemble column  $A_{*k}$  into  $\mathcal{F}^{(k)}$ ;
  for each child  $l$  of  $k$  in the assembly tree do
    assemble the remaining frontal matrix  $\overline{\mathcal{F}}^{(l)}$  into  $\mathcal{F}^{(k)}$ ;
  apply one step Cholesky to factor the first column  $\mathcal{F}_{*1}^{(k)}$  of  $\mathcal{F}^{(k)}$ ;
  strip off  $\mathcal{F}_{*1}^{(k)}$  from  $\mathcal{F}^{(k)}$  to form  $\overline{\mathcal{F}}^{(k)}$ ;
  store  $\mathcal{F}_{*1}^{(k)}$  into  $L_{*k}$ ;
endfor

```

FIG. 4. *Multifrontal method.*

2.2. Graph notation. The nonzero structure of matrix A can be conveniently represented by an undirected graph $G = (V, E)$. The n nodes, $v_1, v_2, \dots, v_n$, correspond to the n columns/rows of the matrix, and an edge connects v_i and v_j if and only if the corresponding entry a_{ij} is nonzero. A graph G is called an “ordered graph” if a bijection

$$\alpha : \{1, 2, \dots, n\} \rightarrow V$$

is defined. The bijection α is called the “ordering” of G . An ordering of G corresponds to a symmetric permutation of the rows and columns of A . Hereafter, to simplify the discussion, we assume that the sparse matrix A is irreducible and has been permuted by a fill-reducing ordering, such as a minimum degree or nested dissection ordering.

For a node v in G , we denote its adjacent set as $\text{adj}(v, G)$ and its degree as $\text{deg}(v, G)$. A node i is *lower* (*higher*) adjacent to v if $i < v$ ($i > v$). $\text{Padj}(v, G)$ ($\text{Madj}(v, G)$) is the set of all nodes adjacent to and numbered lower (higher) than v , and $\text{Pdeg}(v, G)$ ($\text{Mdeg}(v, G)$) is the size of $\text{Padj}(v, G)$ ($\text{Madj}(v, G)$). The next equalities follow immediately from the connection of the notation with the structure of the matrix:

$$\begin{aligned}\text{Padj}(v_j, G) &\equiv \{i < j \mid a_{ij} \neq 0\}, \\ \text{Madj}(v_j, G) &\equiv \{i > j \mid a_{ij} \neq 0\}.\end{aligned}$$

The structural effect of Gaussian elimination on the matrix can easily be modeled by a sequence of graphs $G^{(0)}, G^{(1)}, \dots, G^{(n)}$ [7]: Initially, let $G^{(0)} = G$. At the i th step, where $1 \leq i \leq n$, a node v along with its incident edges are eliminated from $G^{(i-1)}$ and edges are added such that all nodes in $\text{adj}(v, G^{(i-1)})$ are pairwise adjacent. The graph structure of the resulting Cholesky factor L of A , G^* , is called the *elimination graph* of A and denoted by $G^* = \bigcup_{i=0}^{n-1} G^{(i)}$.

A clique is a set of nodes with the property that all of its members are pairwise adjacent. If no other node can be added while preserving the pairwise adjacent property, the clique is called *maximal*. It has been shown that a filled graph G^* can be represented as a set of maximal cliques. Assume that G^* comprises the maximal cliques $K_1, K_2, \dots, K_q$. Following [15], the term *residual clique* refers to the resulting structure of a maximal clique that has undergone some elimination steps. We denote the residual clique of K_j after i elimination steps as $K_j^{(i)}$. Note that the residual clique may not be maximal. Indeed, when a residual clique becomes a proper subset of another residual clique, the former residual clique is not maximal and should be absorbed to maintain the clique representation of the elimination graph. Figure 5 illustrates the elimination graphs and clique structures for the first two elimination steps applied to a filled graph. Initially, $K_1^{(0)} = \{a, d\}$, $K_2^{(0)} = \{b, e, f\}$, $K_3^{(0)} = \{c, e, f\}$, $K_4^{(0)} = \{d, e, f\}$. After eliminating node a , the residual clique of K_1 is left with only one node d , which is clearly a proper subset of and absorbed by K_4 . The next elimination of node b leaves nodes e and f behind in clique K_2 , which is a subset of K_3 and K_4 , and shall be absorbed by either of these two cliques.

A node v in a graph G is *simplicial* if $\text{adj}(v, G)$ is a clique. A maximal clique that contains simplicial nodes is called a *maximal simplicial clique*. Two nodes, u and v , are *non-adjacent* if $\{u\} \cup \text{adj}(u, G) \neq \{v\} \cup \text{adj}(v, G)$. Two nodes are *non-adjacent* if there is no edge between them. The next results follow immediately from these definitions.

LEMMA 2.1. *If u and v are non-adjacent nodes in G , then u and v are non-adjacent in G^* .*

LEMMA 2.2. *If u and v are adjacent nodes in G , then u and v are adjacent in G^* .*

LEMMA 2.3. *If u and v are non-adjacent nodes in G , then u and v are non-adjacent in G^* .*

A *reordering* of A is a permutation such that the reordered matrix will suffer the same fill and require the same number of arithmetic operations for factorization. In the literature [21], [22], such orderings have also been called *fill-reducing*.

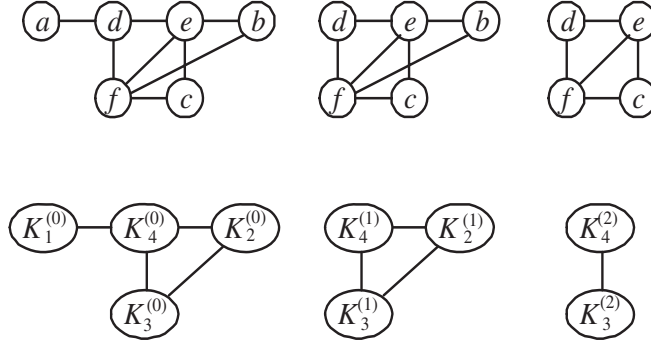


FIG. 5. *Clique representation of the elimination graph for the first two steps, where $K_1^{(0)} = \{a, d\}$, $K_2^{(0)} = \{b, e, f\}$, $K_3^{(0)} = \{c, e, f\}$, $K_4^{(0)} = \{d, e, f\}$. Two cliques are connected if their intersection is not empty.*

In terms of graph theory, a fill-preserving ordering of A retains the structure of the filled graph G^* .

Rose [25] has shown that a filled graph is a chordal graph and always has at least one perfect ordering, i.e., an ordering with no fill. For example, the ordering used to generate the filled graph is a perfect ordering of the filled graph. It is interesting to note that if A is reordered by a perfect ordering on G^* , the resulting filled graph $\tilde{G}^*$ is a subset of G^* [21]. Some of the filled edges in G^* may not appear in $\tilde{G}^*$. That is, a perfect ordering on G^* , as an ordering of A , is at least as good as the original fill-preserving ordering of A in terms of fill and computation. Therefore, finding a fill-preserving ordering for matrix A can be regarded as finding a perfect ordering of the filled graph G^* .

It also has been shown that eliminating a simplicial node creates no filled edge. A perfect ordering thus can be obtained by finding simplicial nodes in the reduced graphs during the elimination process.

LEMMA 2.4. *If $(v_1, v_2, \dots, v_n)$ is a perfect ordering of G^* , then v_j is simplicial in $G^{*(j-1)}$ for $1 \leq j \leq n$.*

3. Computation model and cost specification.

3.1. The elimination tree model. The elimination tree model, which was introduced in [12], has been shown to be useful in exploiting the parallelism that exists in sparse matrix factorization. An elimination tree $T(A)$ of matrix A is a tree containing the same node set as G and has an edge between two nodes v_i and v_j if $v_j = \text{parent}(v_i)$, where $\text{parent}(v_i) = \min\{v_k \mid k > i \text{ and } l_{ki} \neq 0\}$. The elimination tree captures the column/row dependencies of the Cholesky factor [20]. Short elimination trees imply less interdependency and thus more parallelism. Liu [20] illustrated that the elimination tree model is appropriate for analyzing the three versions of Cholesky factorization. Duff and Reid [5] also used the elimination tree as a structure in their multifrontal factorization. We thus concentrate on this computation model.

Let $T[v]$ denote the subtree of $T(A)$ rooted at node v . Some useful properties of the elimination trees in [12] are given below.

LEMMA 3.1. *If $l_{jk} \neq 0$ and $k < j$, then v_k is an ancestor of v_j in $T(A)$.*

This lemma implies that the root vertex is v_n .

LEMMA 3.2. $\dots T[v] \dots T[u] \dots G^* \dots$
 $\dots (x, y) \dots x \in T[v] \dots y \in T[u] \dots$

Consider a perfect ordering α on G^* . Let $\Phi_\alpha(v)$ specify the cost of eliminating a node v in G_α^* with respect to the Cholesky factorization. Since the elimination tree exhibits the column dependencies of the Cholesky factor, it is natural to define the nonnegative cost (called $\dots$) to complete the parallel elimination at node v as the critical weighted path on the subtree $T[v]$. More precisely, let $CP_\alpha(\cdot)$ denote the completion cost function. We have the following recursive definition:

$$CP_\alpha(v) = \begin{cases} \Phi_\alpha(v) & \text{if } v \text{ is a leaf node,} \\ \Phi_\alpha(v) + \max\{CP_\alpha(u) \mid u \text{ is a child of } v\} & \text{otherwise.} \end{cases}$$

The total completion cost, CP_α , to eliminate the entire graph or factorize the corresponding matrix is then equal to $CP_\alpha(v_n)$. We omit the “ordering” subscript when it is clear from the context.

3.2. Independence and conservation properties. Given a filled graph, the work of finding an optimal perfect ordering to minimize the parallel factorization time is not at all trivial; current approaches do not take into consideration various algorithmic forms of factorization. Our previous study [17] provides us with a foundation to work from. We observed that a greedy approach seems promising to yield an optimal perfect ordering under various considerations. We further found that the conditions assuring the optimality of a greedy ordering algorithm can be clarified as two important properties on the cost specification.

Let ρ be a perfect ordering on G^* ,

$$\rho : v_1, v_2, \dots, v_n,$$

such that nodes v_j and v_{j+1} are both simplicial in $G_\rho^{*(j-1)}$. Then the ordering

$$\tilde{\rho} : v_1, v_2, \dots, v_{j-1}, v_{j+1}, v_j, v_{j+2}, \dots, v_n$$

that differs from ρ only in positions j and $j+1$ is also a perfect ordering. We define a class Ω of cost functions that satisfies the following independence and conservation properties, i.e., $\Omega = \{\Phi : \Phi \text{ satisfies the independence and conservation properties}\}$.

DEFINITION 3.3. $\dots v_j \dots v_{j+1} \dots G_\rho^{*(j-1)} \dots$
 $\dots v_j \dots v_{j+1} \dots$

$$\Phi_{\tilde{\rho}}(v_{j+1}) = \Phi_\rho(v_{j+1}) \quad \Phi_{\tilde{\rho}}(v_j) = \Phi_\rho(v_j).$$

DEFINITION 3.4. $\dots v_j \dots v_{j+1} \dots G_\rho^{*(j-1)} \dots$
 $\dots v_j \dots v_{j+1} \dots$

$$\Phi_\rho(v_j) + \Phi_\rho(v_{j+1}) = \Phi_{\tilde{\rho}}(v_j) + \Phi_{\tilde{\rho}}(v_{j+1}).$$

It is worthwhile to point out that when the two simplicial nodes v_j and v_{j+1} belong to different maximal cliques, the two subtrees $T[v_j]$ and $T[v_{j+1}]$ are disjoint. The cost independence property thus constrains the class Ω to preserve the subtree independence property. The conservation property further restricts our attention to the functions that reflect the graph indistinguishability of any nodes within the same simplicial clique, a principle useful in many aspects of the sparse matrix computations [10].

TABLE 1
A category of node cost functions.

		Column	Row	Submatrix	Multifrontal
Computation	Unit	$\Phi 1$	$\Phi 1$	$\Phi 1$	$\Phi 1$
	Multiplications	$\Phi 2$	$\Phi 3$	$\Phi 4$	$\Phi 4$
Communication	Number of messages	$\Phi 5$	$\Phi 5$	$\Phi 5$	$\Phi 6$
	Volume	$\Phi 7$	$\Phi 8$	$\Phi 7$	$\Phi 9$

TABLE 2
Specification of the cost functions in Table 1.

$\Phi 1$	1
$\Phi 2$	$\sum_{v_j \in Padj(v_i) \cup \{v_i\}} K_{v_j}^{(i-1)} $
$\Phi 3$	$\sum_{v_j \in Padj(v_i) \cup \{v_i\}} (K_{v_j}^{(j-1)} - K_{v_j}^{(i)})$
$\Phi 4$	$\frac{1}{2} K_{v_i}^{(i-1)} (K_{v_i}^{(i-1)} + 1)$
$\Phi 5$	$Pdeg(v_i)$
$\Phi 6$	$\sum_{v_j \in child(v_i)} Mdeg(v_j)$
$\Phi 7$	$\sum_{v_j \in Padj(v_i)} K_{v_j}^{(i-1)} $
$\Phi 8$	$\sum_{v_j \in Padj(v_i)} (K_{v_j}^{(j-1)} - K_{v_j}^{(i-1)})$
$\Phi 9$	$\sum_{v_j \in child(v_i)} \frac{1}{2} K_{v_j}^{(i-1)} (K_{v_j}^{(i-1)} + 1)$

3.3. Example cost functions in Ω . There are various factors affecting the specification of the cost function for eliminating nodes in parallel. Some typical factors are parallel architecture: distributed- or shared-memory, the number of processors available, the task granularity and scheduling strategy, and the algorithmic form of the Cholesky factorization. On the basis of the elimination tree model, we considered the parallel sparse column-, row-, submatrix-Cholesky and multifrontal methods on a distributed-memory multiprocessor with the following assumptions:

(1) There is an unlimited number of processors as well as unlimited number of memory modules connected via an interconnection network of sufficiently wide bandwidth.

(2) The column-oriented distribution is used for column-Cholesky, submatrix-Cholesky, and multifrontal, and row-oriented distribution is used for row-Cholesky. Each processor is solely responsible for the task of maintaining and updating its column or row.

(3) For simplicity, we ignore the underlying interconnection and routing topology.

It is difficult to define one cost function that is generally good for most cases. Instead, we chose to specify a cost function for each case. As an example, we considered the typical factors, computation and communication, with respect to different algorithmic forms of Cholesky factorization. Table 1 summarizes this category. The specification of each cost function is described in Table 2, where the notation $K_{v_j}^{(i)}$ denotes the residual set of the maximal clique K_{v_j} in which a node v_j becomes simplicial. The derivation of these specifications is omitted here. We show in Figure 6 an example for $\Phi 2$ only. The details can be found in [18].

Note that function $\Phi 1$ corresponds to the case for unit cost. That is, the criterion for using elimination tree height is a special case of the proposed class. It is not hard to verify that each function satisfies the independence and conservation properties.

$$\begin{pmatrix} a & & & & & \\ & b & & & & \\ & & c & & & \\ \times & & & d & & \\ & \boxed{\times} \boxed{\times} \boxed{\times} \boxed{e} & & & & \\ & \boxed{\times} \boxed{\times} \boxed{\times} \boxed{\times} & f & & & \end{pmatrix}$$

$K_2^{(4)} K_3^{(4)} K_4^{(4)} K_4^{(4)}$

FIG. 6. Illustration of nodal cost $\Phi 2$.

$$\begin{pmatrix} a & & & & \\ & b & & & \\ & & \boxed{c} & & \\ \otimes & & & \otimes d & \\ & \times \boxed{\times} \otimes & e & & \\ & \times \boxed{\times} \otimes \times & f & & \end{pmatrix} \quad \begin{pmatrix} a & & & & \\ & b & & & \\ \otimes & & \otimes d & & \\ & & & \boxed{c} & \\ & \times \otimes \boxed{\times} & e & & \\ & \times \otimes \boxed{\times} \times & f & & \end{pmatrix}$$

FIG. 7. Illustration of the cost-independent property.

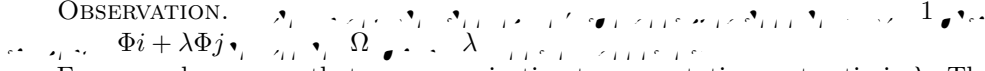
$$\begin{pmatrix} a & & & & \\ & b & & & \\ & & c & & \\ \otimes & & & \otimes d & \\ & \boxed{\times} \boxed{\times} \boxed{\times} e & & & \\ & \boxed{\times} \boxed{\times} \boxed{\times} \times & f & & \end{pmatrix} \quad \begin{pmatrix} a & & & & \\ & b & & & \\ & & c & & \\ & \boxed{\times} \boxed{\times} & e & & \\ \otimes & & & \boxed{\times} \otimes d & \\ & \boxed{\times} \boxed{\times} \boxed{\times} \otimes & f & & \end{pmatrix}$$

FIG. 8. Illustration of the cost-conservative property.

When v_j and v_{j+1} are independent in $G_\rho^{*(j-1)}$, the assertion is evident because the interchange of v_j and v_{j+1} does not change their prior (monotone) adjacent sets as well as the structure of the elimination tree. On the other hand, when v_j and v_{j+1} are indistinguishable in $G_\rho^{*(j-1)}$, the interchange of v_j and v_{j+1} makes v_{j+1} become a member of v_j 's prior adjacent set and v_j a member of v_{j+1} 's monotone adjacent set. Although this would change the nodal costs of v_j and v_{j+1} , the cost variations are the same; thus the sum of the cost remains unchanged.

As an illustration, consider the filled graph in Figure 5 again. Figures 7 and 8 illustrate the independence and conservation properties for the nodal cost function $\Phi 2$, which measures the multiplicative operations for column-Cholesky factorization. In Figure 7, c and d are two independent simplicial nodes after the first two elimination steps. The nonzeros surrounded by squares or circles represent the contribution to

$\Phi 2(c)$ or $\Phi 2(d)$, respectively. As the picture shows, the effect of interchanging c and d is nothing more than a relabeling. In Figure 8, d and e are indistinguishable simplicial nodes after elimination step 3. It can be observed that $\Phi 2(d) + \Phi 2(e)$ equals $4 + 8 = 12$ before the interchange and remains $5 + 7 = 12$ after the interchange.

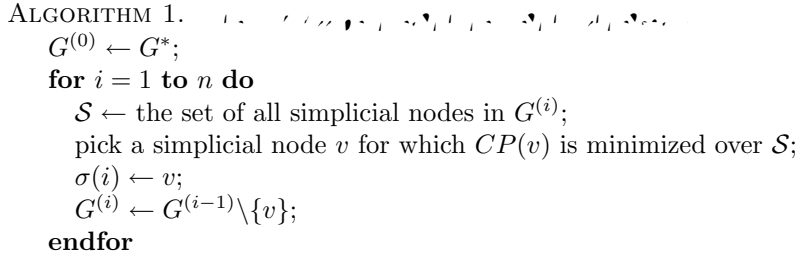
OBSERVATION.  $\Phi i + \lambda \Phi j$, Ω , λ

For example, assume that a communication-to-computation cost ratio is λ . The cost function for eliminating a node under column-Cholesky factorization can be denoted as $\Phi 2 + \lambda \Phi 5$ or $\Phi 2 + \lambda \Phi 7$, a more realistic form.

4. A generic ordering algorithm.

4.1. General description. Assume that the sparse matrix A has been ordered by a fill-reducing ordering and the filled graph is G^* . Our intention is to find a perfect ordering $\sigma : \{1, 2, \dots, n\} \rightarrow V$ on G^* to minimize the completion cost.

Lemma 2.4 gives us the basic guideline—use the elimination process model. In each elimination step, the nodes that are simplicial are identified. Then based on a greedy paradigm, one of the simplicial nodes with the minimum completion cost as defined in section 3 is chosen to be labeled and eliminated next. This process continues until all nodes are eliminated. This greedy ordering strategy is summarized in Algorithm 1.

ALGORITHM 1. 
 $G^{(0)} \leftarrow G^*$;
for $i = 1$ **to** n **do**
 $\mathcal{S} \leftarrow$ the set of all simplicial nodes in $G^{(i)}$;
pick a simplicial node v for which $CP(v)$ is minimized over $\mathcal{S}$;
 $\sigma(i) \leftarrow v$;
 $G^{(i)} \leftarrow G^{(i-1)} \setminus \{v\}$;
endfor

Consider the step for picking the next eliminated node. There are some points that deserve further clarification. First, when node v is chosen to be eliminated in the i th elimination step, we have to calculate its completion cost with the knowledge of an incomplete ordering $\sigma(1), \sigma(2), \dots, \sigma(i)$. This seems to be inconsistent with the definition given in section 3.1, where the completion cost function is determined from the elimination tree corresponding to a complete ordering. But recall that the completion cost of node v is equal to the critical weighted path on the subtree rooted at v , and all descendants of v are ordered earlier than v . Thus, knowing the incomplete ordering $\sigma(1), \sigma(2), \dots, \sigma(i)$ is sufficient to determine the structure of subtree $T[v]$.

Second, to choose the next node for elimination, we have to evaluate the completion cost of each simplicial node, which, however, cannot be worked out until the node is actually eliminated. To resolve the uncertainty about node ordering, we tentatively regard each simplicial node as the next eliminated node and figure out its completion cost. The computed cost thus is a provisional completion cost. The node with the minimum provisional completion cost is eliminated next and its provisional completion cost becomes its actual completion cost.

For illustration, consider the graph in Figure 5. After eliminating nodes a , b , and c , the remaining three nodes, d , e , and f , in the graph are simplicial. Assume the nodal cost is $\Phi 2$. If d is regarded as the next eliminated node, the completion cost of d is $CP(a) + \Phi 2(d) = 2 + 3 = 5$. Similar calculations yield that both e and f have the same provisional completion cost, $3 + 3 = 6$. Hence, node d is eliminated next.

4.2. Minimum completion cost property. In this subsection, we will show that given a filled graph G^* , the ordering obtained with Algorithm 1 minimizes the completion cost over all perfect orderings of G^* .

To facilitate the discussion, we consider two cases: the relative order between independent simplicial nodes and that between indistinguishable simplicial nodes. More specifically, let $\rho : v_1, v_2, \dots, v_n$ be any perfect ordering in which nodes v_j and v_{j+1} are both simplicial in $G^{*(j-1)}$. Then a reordering of ρ , which differs only in the j th and $(j+1)$ st positions, is also a perfect ordering, that is, $\tilde{\rho} : v_1, v_2, \dots, v_{j-1}, v_{j+1}, v_j, v_{j+2}, \dots, v_n$. We will show the effect the interchange of v_j and v_{j+1} has on the completion cost, for each of the two different cases that v_j and v_{j+1} are independent or indistinguishable, and extend the result to a more general case for interchanging a set of consecutive simplicial nodes $v_j, v_{j+1}, \dots, v_{j+s}$ in $G^{*(j-1)}$.

LEMMA 4.1. *Let v_j, v_{j+1} be independent simplicial nodes in $G^{*(j-1)}$. Then $CP_{\tilde{\rho}} = CP_{\rho}$.*

Since v_j and v_{j+1} are independent, $T[v_j]$ and $T[v_{j+1}]$ are disjoint, and the elimination tree remains the same after the interchanging of v_j and v_{j+1} . The statement then follows from the assumption that Φ satisfies the independence property. $\square$

COROLLARY 4.2. *Let $v_j, v_{j+1}, \dots, v_{j+s}$ be consecutive simplicial nodes in $G^{*(j-1)}$. Then $CP_{\tilde{\rho}} = CP_{\rho}$.*

We next consider the case for indistinguishable nodes. To facilitate the discussion and simplify the notation, we introduce the *degree* of a node v in $G^{*(j)}$, $1 \leq j \leq n$, as

$$D_j(v) = \max\{CP(u) \mid u \in \text{adj}(v, G^*) \cap \{v_1, v_2, \dots, v_j\}\},$$

and $D_j(v) = 0$ when $\text{adj}(v, G^*) \cap \{v_1, v_2, \dots, v_j\} = \emptyset$. We do not refer to the children of v because this set is not determined until v is eliminated.

LEMMA 4.3. *Let v_j, v_{j+1} be indistinguishable simplicial nodes in $G^{*(j-1)}$. Then $D_{j-1}(v_j) \geq D_{j-1}(v_{j+1})$ and $CP_{\tilde{\rho}} \leq CP_{\rho}$.*

Clearly, an edge exists between v_j and v_{j+1} . By the elimination tree definition, v_{j+1} must be the parent of v_j in the case of ordering ρ . Therefore,

$$\begin{aligned} CP_{\rho}(v_{j+1}) &= \Phi_{\rho}(v_{j+1}) + \max\{CP_{\rho}(v_j), D_{j-1}(v_{j+1})\} \\ &= \Phi_{\rho}(v_{j+1}) + \max\{\Phi_{\rho}(v_j) + D_{j-1}(v_j), D_{j-1}(v_{j+1})\} \\ &= \Phi_{\rho}(v_{j+1}) + \Phi_{\rho}(v_j) + D_{j-1}(v_j). \end{aligned}$$

In ordering $\tilde{\rho}$, v_j becomes the parent of v_{j+1} . We have

$$\begin{aligned} CP_{\tilde{\rho}}(v_j) &= \Phi_{\tilde{\rho}}(v_j) + \max\{CP_{\tilde{\rho}}(v_{j+1}), D_{j-1}(v_j)\} \\ &= \Phi_{\tilde{\rho}}(v_j) + \max\{\Phi_{\tilde{\rho}}(v_{j+1}) + D_{j-1}(v_{j+1}), D_{j-1}(v_j)\}. \end{aligned}$$

It is immediately clear that $CP_{\tilde{\rho}}(v_j)$ will be $\Phi_{\tilde{\rho}}(v_j) + D_{j-1}(v_j)$ or $\Phi_{\tilde{\rho}}(v_j) + \Phi_{\tilde{\rho}}(v_{j+1}) + D_{j-1}(v_{j+1})$. Both are no greater than $CP_{\rho}(v_{j+1})$ due to the conservation property and the condition that $D_{j-1}(v_j) \geq D_{j-1}(v_{j+1})$. The lemma then follows. $\square$

COROLLARY 4.4. *Let $v_j, v_{j+1}, \dots, v_{j+s}$ be consecutive indistinguishable simplicial nodes in $G^{*(j-1)}$. Then $D_{j-1}(v_j) \leq D_{j-1}(v_{j+1}) \leq \dots \leq D_{j-1}(v_{j+s})$ and $CP_{\tilde{\rho}} \leq CP_{\rho}$.*

COROLLARY 4.5.

Let ρ be an ordering. For any $j, s \geq 1$, we have $D_{j-1}(v_j) = D_{j-1}(v_{j+1}) = \dots = D_{j-1}(v_{j+s})$.

THEOREM 4.6.

Let σ be an ordering. Then σ is a perfect ordering if and only if σ is a simplicial ordering in G^* .

Let $\sigma : u_1, u_2, \dots, u_n$ be the ordering obtained by Algorithm 1 and let $\rho : v_1, v_2, \dots, v_n$ be an optimal ordering. Clearly, both σ and ρ are perfect orderings. We will show that ρ can be transformed into σ without increasing the completion cost. Hence, σ also is optimal.

Assume that $\sigma \neq \rho$ and k is the least index such that $u_k \neq v_k$. Since both σ and ρ are perfect orderings, u_k and v_k must be simplicial in $G_{\sigma}^{*(k-1)}$ as well as $G_{\rho}^{*(k-1)}$ by Lemma 2.4. Furthermore, let $v_j = u_k$ for some j , $k < j \leq n$, and let $\tilde{\rho}$ be a reordering of ρ such that v_j is ordered immediately before v_k , whereas other nodes remain unchanged, i.e., $\tilde{\rho} : v_1 = u_1, \dots, v_{k-1} = u_{k-1}, v_j, v_k, \dots, v_{j-1}, v_{j+1}, \dots, v_n$. We want to show that $\tilde{\rho}$ can be obtained from a sequence of reorderings of ρ such that v_j is shifted left one position each time. The sequence is defined as follows:

$$\begin{aligned} \omega^{(0)}(\rho) &= \rho, \\ \omega^{(i)}(\rho) &= \omega(\omega^{(i-1)}(\rho)) \\ &= \omega(v_1, \dots, v_{j-i-1}, \mathbf{v}_{j-i}, \mathbf{v}_j, v_{j-i+1}, \dots, v_{j-1}, v_{j+1}, \dots, v_n) \\ &= v_1, \dots, v_{j-i-1}, \mathbf{v}_j, \mathbf{v}_{j-i}, v_{j-i+1}, \dots, v_{j-1}, v_{j+1}, \dots, v_n, \quad 1 \leq i \leq j-k. \end{aligned}$$

Note that v_j is simplicial in $G^{*(k-1)}$. It follows from Lemma 2.3 that v_j remains simplicial in $G^{*(l)}$ for $l \geq k$. Hence the sequence of reorderings of ρ defined above produce a perfect ordering.

Consider $\omega^{(i)}(\rho)$. For simplicity, let $\gamma = \omega^{(i-1)}(\rho)$ and $\tilde{\gamma} = \omega^{(i)}(\rho)$. To show that $CP_{\tilde{\gamma}} \leq CP_{\gamma}$, we only have to deliberate, according to Lemma 4.3, on the case that v_j and v_{j-i} belong to the same maximal clique in $G_{\gamma}^{*(j-i-1)}$ and show that $D_{j-i-1}(v_j) \leq D_{j-i-1}(v_{j-i})$.

There are two cases to consider. Suppose first that v_{j-i} is simplicial in $G_{\sigma}^{*(k-1)}$. Since the algorithm chooses v_j at step k , it follows that $D_{k-1}(v_j) \leq D_{k-1}(v_{j-i})$. Since both v_j and v_{j-i} are simplicial in the same maximal clique of $G_{\sigma}^{*(k-1)}$, it follows that $D_{j-i-1}(v_{j-i}) \geq D_{j-i-1}(v_j)$, which is the desired result. On the other hand, suppose that v_{j-i} is not simplicial in $G_{\sigma}^{*(k-1)}$. Then v_{j-i} has at least one descendant in $G_{\sigma}^{*(k-1)}$ and at least one of these descendants, say w , has to be simplicial in $G_{\sigma}^{*(k-1)}$. Again, from the choice of the algorithm at step k , we know that $D_{k-1}(v_j) \leq D_{k-1}(w)$. The ancestor of w right below v_{j-i} is adjacent to v_{j-i} . Since cost increases as one goes up the elimination tree, $D_{k-1}(v_j) \leq D_{j-i-1}(v_{j-i})$. Clearly, if $D_{j-i-1}(v_j) = D_{k-1}(v_j)$, then $D_{j-i-1}(v_{j-i}) \geq D_{j-i-1}(v_j)$. If $D_{j-i-1}(v_j) > D_{k-1}(v_j)$, then some simplicial node u in the same maximal clique containing v_j and v_{j-i} is ordered earlier than $j-i$ and after $k-1$, and $CP(u) = D_{j-i-1}(v_j)$. Since v_{j-i} is also adjacent to u , it follows that $D_{j-i-1}(v_{j-i}) \geq D_{j-i-1}(v_j)$, which completes the proof of the result. $\square$

4.3. Enhancements. Though Algorithm 1 generates a minimum completion cost ordering, it takes more time than necessary to obtain the result. The bottleneck lies in the step that determines the simplicial node with minimum completion cost. From the time a node becomes simplicial, we have to reevaluate its provisional

completion cost in all successive steps that eliminate a neighboring simplicial node until the node is eliminated. We will show in this subsection that the completion cost reevaluation is not essential. Here we will also demonstrate how we can derive a supernode-based ordering strategy.

LEMMA 4.7. *Let v be a nonsimplicial node in $\mathcal{S}$. Let $u_1, u_2, \dots, u_s$ be the simplicial nodes adjacent to v .*

The lemma follows from the fact that a node v deleted from $\mathcal{S}$ has the minimum completion cost among all nodes in $\mathcal{S}$ and the new simplicial nodes to be inserted into $\mathcal{S}$ are ancestors of v . $\square$

THEOREM 4.8. *Let G^* be a graph with n nodes and m edges. Let v be a nonsimplicial node in G^* . Let $u_1, u_2, \dots, u_s$ be the simplicial nodes adjacent to v . Let K be the set of nodes in G^* that are adjacent to v and are not adjacent to any of $u_1, u_2, \dots, u_s$. Let $G^{*(j-1)}$ be the graph obtained from G^* by deleting v and inserting $u_1, u_2, \dots, u_s$. Let $G^{*(j)}$ be the graph obtained from $G^{*(j-1)}$ by deleting $u_1, u_2, \dots, u_s$ and inserting v .*

- (1) $u_1, u_2, \dots, u_s$ are ordered before or between u_{i-1} and u_i for $1 \leq i \leq s$ and $u_0 = v$.
- (2) $u_1, u_2, \dots, u_s$ are ordered before or between u_{i-1} and u_i for $1 \leq i \leq s$ and $u_0 = v$.

To prove the first statement, we assume that a nonsimplicial node w in $K - \{u_1, u_2, \dots, u_s\}$ is ordered before or between u_{i-1} and u_i for $1 \leq i \leq s$ and $u_0 = v$. More specifically, let w be eliminated in step k and ordered between u_{i-1} and u_i for $1 \leq i \leq s$ and $u_0 = v$. The fact that w is nonsimplicial implies that w will have some descendant ordered later than the children of u_i excluding w . It follows that $D_{k-1}(w) \geq D_{k-1}(u_i)$. Hence, w can be ordered after u_i according to Lemma 4.3. This interchange can be repeated until w is ordered after $u_1, u_2, \dots, u_s$. Hence the first statement follows.

Next, note that the completion costs of all nodes adjacent to $u_1, u_2, \dots, u_s$ and eliminated before v are, according to Lemma 4.7, no greater than that of v . It follows that $u_1, u_2, \dots, u_s$ share the same descendant cost immediately after the elimination of v . The only neighbors of $u_1, u_2, \dots, u_s$ eliminated subsequently are simplicial nodes in K . It follows that $u_1, u_2, \dots, u_s$ continue to share the same descendant cost until the first u_i is eliminated. At that point, it follows from Corollary 4.5 that the relative order between $u_1, u_2, \dots, u_s$ is irrelevant. $\square$

As a result, during the progress of greedy elimination, the relative order for indistinguishable simplicial nodes is determined once they became simplicial. The elimination step at which nodes become simplicial thus divides a maximal clique into different classes: several sets of simplicial nodes and one set of nonsimplicial nodes, and the simplicial nodes classified into the same set can be regarded as a “supernode.” This relative ordering posed on the simplicial nodes and nonsimplicial nodes also resolves the uncertainty about the children set of a simplicial node, knowledge of which is required for evaluating its completion cost. Algorithm 2 describes a refinement of Algorithm 1 incorporating this supernode-based elimination without reevaluating the completion cost.

ALGORITHM 2. *Let G^* be a graph with n nodes and m edges. Let v be a nonsimplicial node in G^* . Let $u_1, u_2, \dots, u_s$ be the simplicial nodes adjacent to v . Let K be the set of nodes in G^* that are adjacent to v and are not adjacent to any of $u_1, u_2, \dots, u_s$. Let $G^{*(j-1)}$ be the graph obtained from G^* by deleting v and inserting $u_1, u_2, \dots, u_s$. Let $G^{*(j)}$ be the graph obtained from $G^{*(j-1)}$ by deleting $u_1, u_2, \dots, u_s$ and inserting v .*

```

 $G^{(0)} \leftarrow G^*$ ;
let  $K_1, K_2, \dots, K_q$  be the maximal cliques of  $G^*$ ;
for each  $K_j$ ,  $1 \leq j \leq q$ , do
     $U \leftarrow$  the set of simplicial nodes in  $K_j$ ;
    denote  $U$  as a supernode and compute its completion cost;
     $\mathcal{S} \leftarrow \mathcal{S} \cup \{U\}$ ;
endfor
 $i \leftarrow 1$ ;
while  $\mathcal{S} \neq \emptyset$  do

```

```

pick a supernode  $U$  from  $\mathcal{S}$  with  $\min_{U \in \mathcal{S}} CP(U)$ ;
for each node  $v \in U$  do
   $\sigma(i) \leftarrow v$ ;
   $G^{(i)} \leftarrow G^{(i-1)} \setminus \{v\}$ ;
   $i \leftarrow i + 1$ ;
endfor
 $\mathcal{S} \leftarrow \mathcal{S} \setminus \{U\}$ ;
 $U \leftarrow$  the set of new simplicial nodes in  $G^{(i)}$ ;
denote  $U$  as a supernode and compute its completion cost;
 $\mathcal{S} \leftarrow \mathcal{S} \cup \{U\}$ ;
endwhile

```

4.4. Implementation. The realization of Algorithm 2 requires at least the following considerations: (1) the representation of the elimination graph and the testing of the simplicial nodes; (2) the implementation of the simplicial set $\mathcal{S}$; and (3) the calculation of the completion cost of a simplicial node.

Elimination graph representation and simplicial node detection. As stated previously, any elimination graph can be represented as a clique structure. To simplify the discussions, we omit this material. The detailed procedure for creating the maximal cliques composing G^* , the clique representation of the elimination graph, and the testing of simplicial nodes in each elimination step can be found in [15].

Simplicial set implementation. The primary requirement is an effective way to find the simplicial supernode with the minimum completion cost. A simple solution is using a heap structure in the sense of a heapsort. The root of the heap is the node with the desired property. Note that the result in Theorem 4.8 suggests the possibility of mass elimination. Thus, as a node v is eliminated in step j , the set of the new simplicial nodes $\{u_1, u_2, \dots, u_s\}$ can be regarded as a supernode. Hence, only the representative node, say u_s , is inserted into the heap structure of $\mathcal{S}$. This greatly alleviates the cost of maintaining $\mathcal{S}$.

Completion cost calculation. The completion cost of a simplicial node is composed of two parts: the maximum descendant cost and the nodal cost. The nodal cost depends on the specification of the cost function, which is not hard to calculate but, if not simplified, will dominate the computation of the whole algorithm. In particular, consider the nodal cost functions Φ_2 , Φ_3 , Φ_5 , Φ_7 , and Φ_8 . Calculating these functions involves an accumulation over the prior (or monotone) adjacent set, yielding a complexity of $O(e)$ for all simplicial nodes throughout the algorithm, where e is the number of edges in G^* . The computation can be reduced if the cost functions are specified in terms of the simplicial clique notion. For this purpose, we introduce the notation $\eta_K^{(j)}$, which denotes the set of nodes ordered no later than v_j and whose simplicial clique is K . That is,

$$\eta_K^{(j)} = \{v_i \mid K_{v_i} = K \text{ for } i \leq j\}.$$

As an illustration, consider the graph in Figure 9(a). There are two maximal cliques $K_1 = \{a, b, e\}$ and $K_2 = \{c, d, e, f\}$. Assuming that the nodes are ordered as

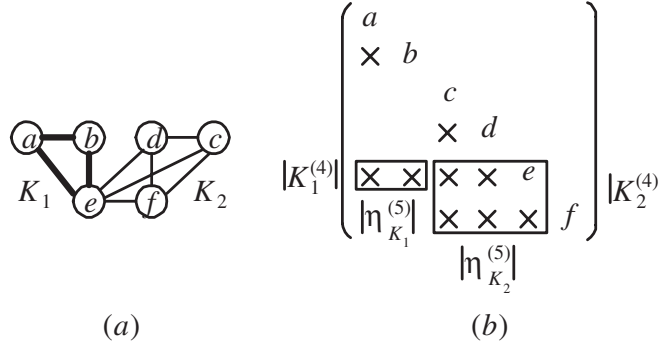


FIG. 9. An example filled graph and the corresponding filled matrix.

a, b, c, d, e, f , we can derive

$$\begin{aligned}
 \eta_{K_1}^{(1)} &= \{a\}, & \eta_{K_2}^{(1)} &= \emptyset, \\
 \eta_{K_1}^{(2)} &= \{a, b\}, & \eta_{K_2}^{(2)} &= \emptyset, \\
 \eta_{K_1}^{(3)} &= \{a, b\}, & \eta_{K_2}^{(3)} &= \{c\}, \\
 \eta_{K_1}^{(4)} &= \{a, b\}, & \eta_{K_2}^{(4)} &= \{c, d\}, \\
 \eta_{K_1}^{(5)} &= \{a, b\}, & \eta_{K_2}^{(5)} &= \{c, d, e\}, \\
 \eta_{K_1}^{(6)} &= \{a, b\}, & \eta_{K_2}^{(6)} &= \{c, d, e, f\}.
 \end{aligned}$$

Let $\mathcal{M}(v_i)$ denote the set of maximal cliques containing v_i in G^* . Note that $\mathcal{M}(v_i)$ also represents the set of simplicial cliques of the nodes in $\text{Padj}(v_i) \cup \{v_i\}$, i.e.,

$$\mathcal{M}(v_i) = \bigcup_{v_j \in \text{Padj}(v_i) \cup \{v_i\}} K_{v_j}.$$

In short, the nodes having the same simplicial cliques form a supernode. Using this representation, the nodal cost $\Phi 2$ can be rewritten as

$$\sum_{K \in \mathcal{M}(v_i)} |\eta_K^{(i)}| |K^{(i-1)}|.$$

For example, consider the nodal cost $\Phi 2$ of node e in Figure 9(a). The set of maximal cliques containing e is $\{K_1, K_2\}$, $\eta_{K_1}^{(5)} = \{a, b\}$, $\eta_{K_2}^{(5)} = \{c, d, e\}$, $K_1^{(4)} = \{e\}$, and $K_2^{(4)} = \{e, f\}$. Hence, $\Phi 2(e) = |\eta_{K_1}^{(5)}| |K_1^{(4)}| + |\eta_{K_2}^{(5)}| |K_2^{(4)}| = 8$, which is depicted in Figure 9(b). A similar transformation can be applied to $\Phi 3$, $\Phi 5$, $\Phi 7$, and $\Phi 8$. As for $\Phi 6$ and $\Phi 9$, the set $\text{child}(v_i)$ can be transformed to the set $\mathcal{CM}(v_i)$ so as to facilitate the supernode representation, where $\mathcal{CM}(v_i)$ represents the set of simplicial cliques of the nodes in $\text{child}(v_i)$. Table 3 shows the modified specifications for the cost functions in Table 2. Using this modification, the computation for nodal cost calculation will be reduced to $O(\kappa)$, where κ represents the sum of the sizes of all of the maximal cliques in G^* .

The completion cost, as stated in Theorem 4.8, can be determined as the node becomes simplicial. Let the new simplicial nodes, as a node v is eliminated in step j ,

TABLE 3
Modified specifications for cost functions in Table 2.

$\Phi 2$	$\sum_{K \in \mathcal{M}(v_i)} \eta_K^{(i)} K^{(i-1)} $
$\Phi 3$	$\sum_{K \in \mathcal{M}(v_i)} \frac{1}{2} \eta_K^{(i)} (2(K^{(0)} - K^{(i)}) - \eta_K^{(i)} + 1)$
$\Phi 5$	$\sum_{K \in \mathcal{M}(v_i)} \eta_K^{(i-1)} $
$\Phi 6$	$\sum_{K \in \mathcal{CM}(v_i)} \eta_K^{(i-1)} $
$\Phi 7$	$\sum_{K \in \mathcal{M}(v_i)} \eta_K^{(i-1)} K^{(i-1)} $
$\Phi 8$	$\sum_{K \in \mathcal{M}(v_i)} \frac{1}{2} \eta_K^{(i-1)} (2(K^{(0)} - K^{(i-1)}) - \eta_K^{(i-1)} + 1)$
$\Phi 9$	$\sum_{K \in \mathcal{CM}(v_i)} \frac{1}{2} K^{(i-1)} (K^{(i-1)} + 1)$

be $u_1, u_2, \dots, u_s$ and all of them are contained within a maximal clique $K^{(j)}$. Note that the relative order between $u_1, u_2, \dots, u_s$ is irrelevant. Hence, it is appropriate to consider the natural order of $u_1, u_2, \dots, u_s$. Let w denote in $K^{(j)}$ the simplicial node that has the maximum completion cost among all nodes in $K^{(j)}$ that became simplicial earlier than $u_1, u_2, \dots, u_s$. Then it is immediately clear that $D_j(u_1) = D_j(u_2) = \dots = D_j(u_s) = \max\{CP(v), CP(w)\}$. Instead of keeping track of the maximum descendant cost of each simplicial node, we simply maintain this cost as the provisional cost $CP(K)$ of the maximal clique to which these simplicial nodes belong. In addition to inserting the simplicial node u_s into $\mathcal{S}$, we update $CP(K)$. With this concept we only have to compare $CP(v)$ and $CP(K)$ to obtain the maximum descendant cost without knowing w .

THEOREM 4.9. *Let G be a graph with n nodes and e edges. Let $\mathcal{S}$ be a simplicial set of G with q nodes. Let G^* be the graph obtained by adding all nodes in $\mathcal{S}$ to G .*

It has been shown in [15] that the creation of maximal cliques and the representation of G^* can be completed in $O(n + \kappa)$ time, where $\kappa = \sum_{i=1}^q |K_i|$ and n denotes the number of nodes in G^* . Since each supernode becomes new simplicial only upon the absorption of a maximal clique and each insertion to the simplicial set $\mathcal{S}$ takes $O(\log q)$ time, the maintenance of the simplicial set $\mathcal{S}$ through the whole process consumes at most $O(q \log q)$ time. The cost for computing the nodal cost is constant or proportional to $Pdeg(v)$ or $|\text{child}(v)|$, depending on the specification of the cost function. For n nodes, the total cost would be $O(e)$ in the worst case, where e denotes the number of edges in G^* . Note that in our implementation the calculation is indeed performed via the supernode concept. The cost is thus reduced to $O(\kappa)$. Hence, the overall complexity is $O(q \log q + \kappa)$. $\square$

5. Experimental results. In this section, we present the experimental results on some test matrices from the collection at the University of Florida.¹ Table 4 lists the name, a description, the order, and the number of nonzeros in the lower triangular part of each test matrix.

All experiments were performed on an SGI Origin 3800 system with 32 processors. We first used Liu's multiple minimum degree (MMD) algorithm [19] to reorder each of the matrices and performed symbolic factorization to obtain the corresponding filled matrix. Thereafter we applied the Jess-Kees algorithm (following the leading implementation in [15]) and our MinCP algorithm, respectively, to each of the reordered

¹Available from <http://www-pub.cise.ufl.edu/~davis/sparse>.

TABLE 4
Test matrices from the University of Florida Sparse Matrix Collection.

Key	Description	Order	Nonzeros in $A(10^3)$
3DTUBE	3D pressure tube	45330	1629
BARTH4	Structural engineering problem	6019	23
BARTH5	Structural engineering problem	15606	61
BCSSTK30	Structural eng., off-shore platform	28924	1036
BCSSTK31	Structural eng., automobile component	35588	608
BCSSTK32	Structural eng., automobile chassis	44609	1029
BCSSTK35	Structural eng., automobile seat frame	30237	740
BCSSTK36	Structural eng., automobile shock absorber	23052	583
BCSSTK37	Structural eng., track ball	25503	583
BCSSTK39	Shuttle solid rocket booster	46772	1068
CFD1	CFD, symmetric pressure matrix	70656	949
CFD2	CFD, symmetric pressure matrix	123440	1605
CRYSTK02	Structural eng., crystal vibration	19365	497
CRYSTK03	Structural eng., crystal vibration	24696	863
CT20STIF	Structural eng., CT20 engine block	52329	1323
FINAN512	Economics, portfolio optimization	74752	336
GEARBOX	ZF aircraft flap actuator	153746	4617
LI	3D FEM, magnetohydrodynamic problem	22695	686
MSC10848	Aircraft structure	10848	620
MSC23052	Naval destroyer structure	23052	588
NASASRB	Shuttle rocket booster	54870	1366
PWT	Structural engineering problem	36519	181
SHUTTLE	Structural engineering problem	10429	57
SKIRT	Structural engineering problem	12598	104
STRUCT3	L-shaped regular grid	53570	613
STRUCT4	Full three ocean model	4350	121
VIBROBOX	Vibroacoustic problem	12328	177

matrices. Since the minimum degree ordering is sensitive to tie breaking and in turn will affect the cost for parallel factorization, we randomly permuted all test matrices 100 times (including the original one). All methods were run on the same randomized matrices. For our purpose, we are concerned with the completion cost difference when using the Jess–Kees algorithm as the ordering method under any criterion. This value will justify the effectiveness of our method. Every randomized matrix was tested to collect the difference.

The mean and the best of the differences are reported in Tables 5 and 6, respectively. For most of the test matrices, the mean difference is less than 10%, whereas it reaches 18% for BCSSTK30, BCSSTK35, and CFD1 in the case of Φ_4 and Φ_9 . The result for best difference, however, is larger than 15% for most matrices, and reaches 41% and 66% for BCSSTK35 and SKIRT, respectively. Note that the results in the first three columns are quite similar to those in the last three columns. This is because the formulations of Φ_2 , Φ_3 , and Φ_4 are similar to Φ_7 , Φ_8 , and Φ_9 , respectively. It is also interesting to note that the multifrontal factorization benefits the most from our

TABLE 5
Completion cost increase (mean) of the Jess-Kees algorithm to MinCP, initially ordered by MMD.

Key	$\Phi 2$	$\Phi 3$	$\Phi 4$	$\Phi 5$	$\Phi 6$	$\Phi 7$	$\Phi 8$	$\Phi 9$
3DTUBE	+0%	+0%	+1%	+0%	+0%	+0%	+0%	+1%
BARTH4	+10%	+6%	+13%	+5%	+5%	+10%	+6%	+13%
BARTH5	+2%	+1%	+5%	+1%	+1%	+2%	+1%	+5%
BCSSTK30	+16%	+11%	+18%	+4%	+8%	+16%	+11%	+18%
BCSSTK31	+2%	+1%	+5%	+1%	+1%	+2%	+1%	+5%
BCSSTK32	+10%	+6%	+14%	+5%	+7%	+10%	+6%	+14%
BCSSTK35	+11%	+3%	+18%	+2%	+7%	+11%	+3%	+18%
BCSSTK36	+1%	+1%	+4%	+1%	+1%	+1%	+1%	+4%
BCSSTK37	+5%	+1%	+14%	+1%	+6%	+5%	+1%	+14%
BCSSTK39	+6%	+5%	+8%	+2%	+3%	+6%	+5%	+8%
CFD1	+12%	+7%	+18%	+6%	+9%	+12%	+7%	+18%
CFD2	+1%	+1%	+2%	+1%	+1%	+1%	+1%	+2%
CRYSTK02	+1%	+1%	+2%	+1%	+1%	+1%	+1%	+2%
CRYSTK03	+2%	+2%	+3%	+1%	+1%	+2%	+2%	+3%
CT20STIF	+6%	+3%	+14%	+4%	+8%	+6%	+3%	+14%
FINAN512	+0%	+0%	+1%	+0%	+0%	+0%	+0%	+1%
GEARBOX	+1%	+0%	+2%	+0%	+1%	+1%	+0%	+2%
LI	+1%	+0%	+2%	+0%	+1%	+1%	+0%	+2%
MSC10848	+3%	+2%	+3%	+1%	+1%	+3%	+2%	+3%
MSC23052	+1%	+1%	+4%	+0%	+1%	+1%	+1%	+4%
NASASRB	+4%	+4%	+4%	+2%	+1%	+4%	+4%	+4%
PWT	+2%	+5%	+1%	+3%	+0%	+2%	+5%	+1%
SHUTTLE	+3%	+2%	+3%	+1%	+1%	+3%	+3%	+3%
SKIRT	+6%	+7%	+8%	+3%	+2%	+6%	+8%	+8%
STRUCT3	+1%	+1%	+1%	+0%	+0%	+1%	+1%	+1%
STRUCT4	+0%	+0%	+1%	+0%	+0%	+0%	+0%	+1%
VIBROBOX	+0%	+0%	+1%	+0%	+0%	+0%	+0%	+1%

ordering algorithm (see columns $\Phi 4$ and $\Phi 9$), then the submatrix-Cholesky, column-Cholesky, and last the row-Cholesky factorization.

It is well known that MMD will lead to elimination trees that are not well balanced. So, most good orderings used today are hybrids of nested dissection (ND) and minimum degree (MD) [1], [11], [14], [24]. We thus conducted another experiment, using one of the leading hybrid ND/MD orderings, METIS [13], in place of MMD, to evaluate the gains of our ordering. The results are shown in Tables 7 and 8.

It is not surprising that the gains decrease for most of the test matrices, which corroborates the result in the literature that ND produces more balanced elimination trees than MMD and leads to more parallelism. For matrices derived from complex geometries, however, it is known that ND does not necessarily produce good separators and/or result in well-balanced elimination trees. In this case, one can expect large reductions in completion cost through our algorithm. Matrices SHUTTLE, STRUCT3, and FINAN512 highlight this phenomenon; all exhibit more cost reductions when they are initially ordered by METIS than by MMD, measured by either the mean or the

TABLE 6

Completion cost increase (best) of the Jess–Kees algorithm to MinCP, initially ordered by MMD.

Key	$\Phi 2$	$\Phi 3$	$\Phi 4$	$\Phi 5$	$\Phi 6$	$\Phi 7$	$\Phi 8$	$\Phi 9$
3DTUBE	+11%	+5%	+16%	+4%	+7%	+11%	+5%	+16%
BARTH4	+16%	+12%	+20%	+12%	+14%	+16%	+12%	+20%
BARTH5	+16%	+9%	+29%	+6%	+12%	+16%	+9%	+29%
BCSSTK30	+34%	+26%	+33%	+14%	+16%	+34%	+26%	+33%
BCSSTK31	+22%	+13%	+26%	+11%	+16%	+22%	+13%	+26%
BCSSTK32	+37%	+23%	+39%	+16%	+23%	+37%	+23%	+39%
BCSSTK35	+27%	+9%	+41%	+7%	+21%	+27%	+9%	+41%
BCSSTK36	+18%	+8%	+33%	+9%	+19%	+18%	+8%	+33%
BCSSTK37	+22%	+8%	+36%	+11%	+22%	+22%	+8%	+36%
BCSSTK39	+17%	+15%	+21%	+10%	+10%	+17%	+16%	+21%
CFD1	+27%	+17%	+37%	+12%	+19%	+27%	+17%	+37%
CFD2	+14%	+11%	+20%	+11%	+8%	+14%	+11%	+20%
CRYSTK02	+13%	+9%	+17%	+7%	+12%	+13%	+9%	+17%
CRYSTK03	+21%	+27%	+16%	+16%	+13%	+21%	+27%	+16%
CT20STIF	+27%	+14%	+35%	+16%	+28%	+27%	+14%	+35%
FINAN512	+8%	+2%	+21%	+2%	+10%	+8%	+2%	+21%
GEARBOX	+19%	+10%	+24%	+10%	+16%	+19%	+10%	+24%
LI	+12%	+3%	+23%	+5%	+15%	+12%	+3%	+23%
MSC10848	+22%	+15%	+29%	+13%	+11%	+22%	+15%	+29%
MSC23052	+15%	+10%	+29%	+9%	+20%	+15%	+10%	+29%
NASASRB	+15%	+14%	+17%	+8%	+8%	+15%	+14%	+17%
PWT	+14%	+29%	+9%	+14%	+5%	+14%	+29%	+9%
SHUTTLE	+9%	+10%	+10%	+7%	+6%	+9%	+10%	+10%
SKIRT	+55%	+66%	+37%	+36%	+24%	+55%	+66%	+37%
STRUCT3	+12%	+8%	+18%	+8%	+7%	+12%	+8%	+18%
STRUCT4	+5%	+1%	+15%	+2%	+8%	+5%	+1%	+15%
VIBROBOX	+3%	+1%	+20%	+2%	+6%	+3%	+1%	+20%

best difference. Other matrices in our test set that partially exhibit this phenomenon include BARTH4, BARTH5, BCSSTK30, BCSSTK37, and PWT. For these matrices, we observe that, in the case of multifrontal factorization, the performance gains obtained with initial orderings from METIS are larger than those from MMD. Similar to the case when the initial orderings are produced by MMD, the multifrontal factorization benefits the most from our ordering algorithm, then the submatrix-Cholesky, column-Cholesky, and last the row-Cholesky factorization.

As one might expect, the overhead spent on ordering should not exceed the performance gain in the factorization. Table 9 reports the CPU times for the minimum height ordering and our minimum completion ordering. As the results show, our minimum completion cost ordering takes approximately 2–4 times more time than minimum height ordering. The overhead incurred by replacing minimum height ordering with our scheme is small as compared with the prospective improvement in the most time-consuming step—factorization.

TABLE 7

Completion cost increase (mean) of the Jess-Kees algorithm to MinCP, initially ordered by METIS.

Key	$\Phi 2$	$\Phi 3$	$\Phi 4$	$\Phi 5$	$\Phi 6$	$\Phi 7$	$\Phi 8$	$\Phi 9$
3DTUBE	+0%	+0%	+0%	+0%	+0%	+0%	+0%	+0%
BARTH4	+2%	+0%	+8%	+0%	+2%	+2%	+0%	+8%
BARTH5	+0%	+0%	+3%	+0%	+0%	+0%	+0%	+3%
BCSSTK30	+4%	+0%	+25%	+0%	+4%	+4%	+0%	+25%
BCSSTK31	+2%	+1%	+5%	+1%	+2%	+2%	+1%	+5%
BCSSTK32	+0%	+0%	+2%	+0%	+0%	+0%	+0%	+1%
BCSSTK35	+4%	+1%	+7%	+1%	+4%	+4%	+1%	+7%
BCSSTK36	+1%	+0%	+4%	+0%	+1%	+1%	+0%	+4%
BCSSTK37	+3%	+2%	+7%	+1%	+3%	+3%	+2%	+7%
BCSSTK39	+0%	+0%	+1%	+0%	+0%	+0%	+0%	+1%
CFD1	+3%	+1%	+7%	+1%	+3%	+3%	+1%	+7%
CFD2	+0%	+0%	+0%	+0%	+0%	+0%	+0%	+0%
CRYSTK02	+0%	+0%	+0%	+0%	+0%	+0%	+0%	+0%
CRYSTK03	+0%	+0%	+6%	+0%	+0%	+0%	+0%	+6%
CT20STIF	+1%	+0%	+2%	+0%	+1%	+1%	+0%	+2%
FINAN512	+2%	+2%	+5%	+1%	+2%	+2%	+2%	+5%
GEARBOX	+0%	+0%	+0%	+0%	+0%	+0%	+0%	+0%
LI	+0%	+0%	+5%	+0%	+0%	+0%	+0%	+5%
MSC10848	+1%	+0%	+2%	+0%	+1%	+1%	+0%	+2%
MSC23052	+1%	+0%	+3%	+0%	+1%	+1%	+0%	+3%
NASASRB	+0%	+0%	+1%	+0%	+0%	+0%	+0%	+1%
PWT	+0%	+0%	+2%	+0%	+0%	+0%	+0%	+2%
SHUTTLE	+1%	+1%	+4%	+0%	+2%	+1%	+1%	+4%
SKIRT	+3%	+2%	+7%	+1%	+4%	+3%	+2%	+7%
STRUCT3	+2%	+0%	+8%	+0%	+2%	+2%	+0%	+8%
STRUCT4	+0%	+0%	+1%	+0%	+0%	+0%	+0%	+1%
VIBROBOX	+0%	+0%	+0%	+0%	+0%	+0%	+0%	+0%

6. Conclusions. The height of the elimination tree has long acted as the only criterion in deriving a suitable fill-preserving sparse matrix ordering for parallel factorization. Although it is well known that adopting height as the criterion for all circumstances is deficient, this has never been successfully addressed by any research. This study is the first one to expand the ordering criterion into a more general class that reflects the various aspects of parallel factorization. We recognize that if any cost function satisfies the proposed independent and conservative properties, a greedy ordering scheme then generates an optimal ordering with minimum completion cost.

It should be noted that our emphasis in this paper has been on pursuing optimal equivalent reorderings for parallel Cholesky factorizations from various theoretical perspectives. To justify the results obtained in this study, we still need to conduct a comprehensive experiment on diverse actual parallel Cholesky factorization codes.

Recently, more effective results for parallel factorization were achieved through a larger task model, such as the clique or supernodal tree [9], [26], [27]. In such a case,

TABLE 8

Completion cost increase (best) of the Jess-Kees algorithm to MinCP, initially ordered by METIS.

Key	$\Phi 2$	$\Phi 3$	$\Phi 4$	$\Phi 5$	$\Phi 6$	$\Phi 7$	$\Phi 8$	$\Phi 9$
3DTUBE	+0%	+0%	+0%	+0%	+0%	+0%	+0%	+0%
BARTH4	+22%	+1%	+61%	+0%	+23%	+22%	+1%	+60%
BARTH5	+5%	+1%	+36%	+0%	+7%	+5%	+1%	+35%
BCSSTK30	+30%	+13%	+50%	+13%	+31%	+30%	+13%	+50%
BCSSTK31	+1%	+1%	+12%	+1%	+0%	+1%	+1%	+12%
BCSSTK32	+15%	+7%	+33%	+10%	+15%	+15%	+7%	+33%
BCSSTK35	+13%	+8%	+21%	+7%	+12%	+13%	+8%	+21%
BCSSTK36	+9%	+9%	+15%	+9%	+12%	+9%	+9%	+15%
BCSSTK37	+23%	+12%	+48%	+12%	+19%	+23%	+12%	+48%
BCSSTK39	+1%	+1%	+21%	+0%	+1%	+1%	+1%	+21%
CFD1	+15%	+11%	+28%	+5%	+17%	+15%	+11%	+28%
CFD2	+0%	+0%	+6%	+0%	+0%	+0%	+0%	+6%
CRYSTK02	+0%	+0%	+0%	+0%	+0%	+0%	+0%	+0%
CRYSTK03	+0%	+0%	+21%	+0%	+0%	+0%	+0%	+21%
CT20STIF	+9%	+2%	+19%	+3%	+10%	+9%	+2%	+19%
FINAN512	+13%	+13%	+13%	+9%	+10%	+13%	+13%	+13%
GEARBOX	+2%	+2%	+9%	+2%	+4%	+2%	+2%	+9%
LI	+0%	+2%	+23%	+0%	+5%	+0%	+2%	+23%
MSC10848	+11%	+3%	+18%	+1%	+9%	+11%	+3%	+18%
MSC23052	+6%	+7%	+15%	+8%	+9%	+6%	+7%	+15%
NASASRB	+0%	+0%	+8%	+2%	+4%	+0%	+0%	+8%
PWT	+9%	+8%	+23%	+6%	+13%	+9%	+8%	+23%
SHUTTLE	+14%	+21%	+26%	+15%	+15%	+14%	+21%	+26%
SKIRT	+13%	+14%	+34%	+10%	+16%	+13%	+14%	+33%
STRUCT3	+14%	+6%	+34%	+3%	+15%	+14%	+6%	+34%
STRUCT4	+4%	+1%	+13%	+0%	+6%	+4%	+1%	+13%
VIBROBOX	+1%	+1%	+1%	+0%	+0%	+1%	+1%	+1%

ordering not only is the permutation of the nodes in the filled graph, but also affects the construction of each supernode. Different ordering may lead to a totally different set of supernodes. To our knowledge, recent literature [3], [8] only achieved finding an ordering to minimize the height of the clique tree. Methods for finding optimal orderings in terms of other more general and realistic concerns remain unanswered. We are currently engaged in this investigation and expect to have some results in the near future.

Acknowledgments. We would like to thank John G. Lewis for many valuable suggestions and comments, especially on improving the conservation property for the nodal cost function. We thank the referees for many thoughtful suggestions, particularly in pointing out some errors of our previous implementation of the MinCP algorithm. We are also grateful for the support from the National Center for High-performance Computing (NCHC) in using SGI Origin 3800.

TABLE 9

Time (CPU seconds) to execute the minimum height (Jess-Kees algorithm) and minimum completion cost orderings (MinCP).

Key	Jess-Kees	$\Phi 2$	$\Phi 3$	$\Phi 4$	$\Phi 5$	$\Phi 6$	$\Phi 7$	$\Phi 8$	$\Phi 9$
3DTUBE	0.22	0.56	0.55	0.39	0.48	0.51	0.53	0.53	0.52
BARTH4	0.02	0.04	0.04	0.03	0.03	0.04	0.04	0.04	0.04
BARTH5	0.05	0.11	0.11	0.09	0.10	0.10	0.10	0.11	0.11
BCSSTK30	0.08	0.18	0.17	0.13	0.15	0.16	0.17	0.17	0.16
BCSSTK31	0.12	0.28	0.27	0.20	0.24	0.25	0.26	0.27	0.26
BCSSTK32	0.13	0.31	0.30	0.22	0.26	0.28	0.29	0.29	0.28
BCSSTK35	0.08	0.17	0.17	0.12	0.15	0.16	0.16	0.16	0.16
BCSSTK36	0.06	0.13	0.13	0.09	0.11	0.12	0.12	0.12	0.12
BCSSTK37	0.07	0.15	0.15	0.11	0.13	0.14	0.14	0.14	0.14
BCSSTK39	0.14	0.33	0.32	0.24	0.28	0.30	0.31	0.31	0.31
CFD1	0.44	1.22	1.21	0.86	1.05	1.12	1.16	1.16	1.16
CFD2	0.77	2.18	2.23	1.59	2.00	2.08	2.19	2.17	2.18
CRYSTK02	0.06	0.15	0.15	0.11	0.13	0.14	0.14	0.14	0.14
CRYSTK03	0.12	0.29	0.28	0.20	0.25	0.26	0.27	0.27	0.27
CT20STIF	0.17	0.42	0.42	0.31	0.37	0.39	0.41	0.41	0.40
FINAN512	0.34	0.92	1.02	0.71	0.91	0.94	0.94	0.98	0.93
GEARBOX	0.63	1.64	1.64	1.18	1.43	1.53	1.58	1.59	1.52
LI	0.21	0.57	0.56	0.40	0.48	0.51	0.53	0.53	0.52
MSC10848	0.03	0.07	0.07	0.05	0.06	0.06	0.07	0.06	0.06
MSC23052	0.06	0.13	0.13	0.10	0.11	0.12	0.13	0.13	0.13
NASASRB	0.19	0.49	0.48	0.35	0.42	0.45	0.46	0.46	0.46
PWT	0.12	0.31	0.32	0.24	0.28	0.29	0.30	0.31	0.30
SHUTTLE	0.03	0.08	0.08	0.06	0.07	0.07	0.08	0.08	0.08
SKIRT	0.04	0.10	0.09	0.07	0.08	0.09	0.09	0.09	0.09
STRUCT3	0.19	0.50	0.50	0.37	0.44	0.47	0.48	0.49	0.48
STRUCT4	0.04	0.08	0.08	0.06	0.07	0.07	0.08	0.07	0.07
VIBROBOX	0.07	0.16	0.16	0.12	0.14	0.15	0.15	0.15	0.15

REFERENCES

- [1] C. ASHCRAFT AND J.W.H. LIU, *Robust ordering of sparse matrices using multisection*, SIAM J. Matrix Anal. Appl., 19 (1998), pp. 816–832.
- [2] B. ASPVALL AND P. HEGGERNES, *Finding minimum height elimination tree for interval graphs in polynomial time*, BIT, 34 (1994), pp. 484–509.
- [3] J.R.S. BLAIR AND B.W. PEYTON, *On finding minimum-diameter clique trees*, Nordic J. Comput., 1 (1994), pp. 173–201.
- [4] J.J. DONGARRA, F.G. GUSTAVSON, AND A. KARP, *Implementing linear algebra algorithms for dense matrices on a vector pipeline machine*, SIAM Rev., 26 (1984), pp. 91–112.
- [5] I.S. DUFF AND J.K. REID, *The multifrontal solution of indefinite sparse symmetric linear equations*, ACM Trans. Math. Software, 9 (1983), pp. 302–325.
- [6] I.S. DUFF, R.G. GRIMES, AND J.G. LEWIS, *Sparse matrix test problems*, ACM Trans. Math. Software, 15 (1989), pp. 1–14.
- [7] A. GEORGE AND J.W.H. LIU, *Computer Solution of Large Sparse Positive Definite Systems*, Prentice-Hall, Englewood Cliffs, NJ, 1981.
- [8] J.R. GILBERT AND R. SCHREIBER, *Highly parallel sparse Cholesky factorization*, SIAM J. Sci. Statist. Comput., 13 (1992), pp. 1151–1172.

- [9] A. GUPTA AND V. KUMAR, *Optimally scalable parallel sparse Cholesky factorization*, in Proceedings of the Seventh SIAM Conference on Parallel Processing for Scientific Computing, SIAM, Philadelphia, 1995, pp. 442–447.
- [10] M.T. HEATH, E. NG, AND B.W. PEYTON, *Parallel algorithms for sparse linear systems*, SIAM Rev., 33 (1991), pp. 420–460.
- [11] B. HENDRICKSON AND E. ROTHBERG, *Improving the run time and quality of nested dissection ordering*, SIAM J. Sci. Comput., 20 (1998), pp. 468–489.
- [12] J.A.G. JESS AND H.G.M. KEES, *A data structure for parallel L/U decomposition*, IEEE Trans. Comput., 31 (1982), pp. 231–239.
- [13] G. KARYPIS AND V. KUMAR, *METIS—A software package for partitioning unstructured graphs, partitioning meshes, and computing fill-reducing orderings of sparse matrices*, Version 4.0, University of Minnesota, Minneapolis, MN, 1998.
- [14] G. KARYPIS AND V. KUMAR, *A fast and high quality multilevel scheme for partitioning irregular graphs*, SIAM J. Sci. Comput., 20 (1998), pp. 359–392.
- [15] J.G. LEWIS, B.W. PEYTON, AND A. POTHEIN, *A fast algorithm for reordering sparse matrices for parallel factorization*, SIAM J. Sci. Statist. Comput., 10 (1989), pp. 1146–1173.
- [16] W.Y. LIN AND C.L. CHEN, *Minimum completion time criterion for parallel sparse Cholesky factorization*, in Proceedings of the International Conference on Parallel Processing, St. Charles, IL, 1993, pp. 107–114.
- [17] W.Y. LIN AND C.L. CHEN, *Minimum communication cost reordering for parallel sparse Cholesky factorization*, Parallel Comput., 25 (1999), pp. 943–967.
- [18] W.Y. LIN AND C.L. CHEN, *On evaluating elimination tree based parallel sparse Cholesky factorizations*, Int. J. Comput. Math., 74 (2000), pp. 361–377.
- [19] J.W.H. LIU, *Modification of the minimum degree algorithm by multiple elimination*, ACM Trans. Math. Software, 11 (1985), pp. 141–153.
- [20] J.W.H. LIU, *Computational models and task scheduling for parallel sparse Cholesky factorization*, Parallel Comput., 3 (1986), pp. 327–342.
- [21] J.W.H. LIU, *Equivalent sparse matrix reordering by elimination tree rotations*, SIAM J. Sci. Statist. Comput., 9 (1988), pp. 424–444.
- [22] J.W.H. LIU, *Reordering sparse matrices for parallel elimination*, Parallel Comput., 11 (1989), pp. 73–91.
- [23] J.W.H. LIU AND A. MIRZAIAN, *A linear reordering algorithm for parallel pivoting of chordal graphs*, SIAM J. Discrete Math., 2 (1989), pp. 100–107.
- [24] F. PELLEGRINI, J. ROMAN, AND P. AMESTOY, *Hybridizing nested dissection and halo approximate minimum degree for efficient sparse matrix ordering*, Concurrency: Practice Experience, 12 (2000), pp. 69–84.
- [25] D.J. ROSE, *A graph-theoretic study of the numerical solution of sparse positive definite systems of linear equations*, in Graph Theory and Computing, R.C. Read, ed., Academic Press, New York, 1972, pp. 183–217.
- [26] E. ROTHBERG, *Performance of panel and block approaches to sparse Cholesky factorization on the iPSC/860 and Paragon multicomputers*, SIAM J. Sci. Comput., 17 (1996), pp. 699–713.
- [27] E. ROTHBERG AND A. GUPTA, *An efficient block-oriented approach to parallel sparse Cholesky factorization*, SIAM J. Sci. Comput., 15 (1994), pp. 1413–1439.
- [28] M. YANNAKAKIS, *Computing the minimum fill-in is NP-complete*, SIAM J. Algebraic Discrete Methods, 2 (1981), pp. 77–79.

ON THE LOW-RANK APPROXIMATION OF DATA ON THE UNIT SPHERE*

M. CHU[†], N. DEL BUONO[‡], L. LOPEZ[‡], AND T. POLITI[§]

Abstract. In various applications, data in multidimensional space are normalized to unit length. This paper considers the problem of best fitting given points on the m -dimensional unit sphere S^{m-1} by k -dimensional great circles with k much less than m . The task is cast as an algebraically constrained low-rank matrix approximation problem. Using the fidelity of the low-rank approximation to the original data as the cost function, this paper offers an analytic expression of the projected gradient which, on one hand, furnishes the first order optimality condition and, on the other hand, can be used as a numerical means for solving this problem.

Key words. standardized data, linear model, factor analysis, low-rank approximation, latent semantic indexing, projected gradient

AMS subject classifications. 62H25, 68P20, 65K10, 90C52, 65F30

DOI. 10.1137/S0895479803433295

1. Introduction. Given n points on the unit sphere S^{m-1} in $\mathbb{R}^m$, we concern ourselves in this paper with the question of best fitting these points by a “great circle,” which is understood to mean the intersection of S^{m-1} with a certain k -dimensional subspace of $\mathbb{R}^m$. Because the approximation is by points restricted to S^{m-1} , we are not dealing with an ordinary data fitting problem. Denoting the given points as rows of a data matrix A , the task of best fitting can be recast as a constrained low-rank matrix approximation problem as follows:

$$\begin{aligned} & A \in \mathbb{R}^{n \times m} \\ & Z \in \mathbb{R}^{n \times m} \quad A \approx Z \\ & \text{rank}(Z) = k \quad k < \min\{m, n\} \end{aligned}$$

Structured low-rank approximation problems arise in many important applications [3, 9, 11, 12, 33]. Some basic theory and computation schemes, including a generic alternating projection algorithm and a robust equality constrained optimization formulation, can be found in a recent article [8] by the first author and in the references contained therein. The structure in our current problem assumes the form that each row of Z has unit length. A similar setting appears in an application for designing structured tight frames [34] where the alternating projection method is used as an effective tool. When the row number n is much larger than the rank k , however, reformulations of this problem as the two schemes proposed in [8] would involve a significant number of redundancies in the constraints, which, in turn, might cause computational difficulties. The approximation obtained by a linear combination of

*Received by the editors August 11, 2003; accepted for publication (in revised form) by I. S. Dhillon October 25, 2004; published electronically June 22, 2005.

<http://www.siam.org/journals/simax/27-1/43329.html>

[†]Department of Mathematics, North Carolina State University, Raleigh, NC 27695-8205 (chu@math.ncsu.edu). The research of this author was supported in part by the NSF under grants DMS-0073056 and CCR-0204157, and in part by the INDAM under the Visiting Professorship Program when the author visited the Department of Mathematics at the University of Bari.

[‡]Dipartimento di Matematica, Università degli Studi di Bari, Via E. Orabona 4, I-70125 Bari, Italy (delbuono@dm.uniba.it, lopezl@dm.uniba.it).

[§]Dipartimento di Matematica, Politecnico di Bari, Via Amendola 126/B, 70126 Bari, Italy (politi@poliba.it).

concept vectors generated by the spherical k -means method [13], on the other hand, lacks the normalization and is limited only to the subspace spanned by the concept vectors. In this paper we propose using the projected gradient approach as an alternative means of tackling the problem. Three types of measurement of best fitting will be used and their results compared.

The requirement of normalizing data before approximations can be performed emerges from many applications. See, for example, the book [15] about pattern classification, the monograph [22] and the vast literature on principal component analysis, the required goal in positive matrix factorization [26], and the spherical k -means method for concept mining in [13]. It might be illuminating to brief readers on some simple examples of why such a normalized low-rank approximation problem is of importance in practice. To set the stage, let $Y = [y_{ij}] \in \mathbb{R}^{n \times \ell}$ denote the matrix of “observed” data. The entry y_{ij} represents, in a broad sense, the score obtained by entity j on variable i . By a standard score we mean that a raw score per variable has been normalized to have mean 0 and standard deviation 1. After this normalization, the matrix

$$(1.1) \quad R := \frac{1}{\ell} Y Y^\top$$

represents the correlation matrix of all n variables. Note that $r_{ii} = 1$ and $|r_{ij}| \leq 1$ for all $i, j = 1, \dots, n$. In a linear model, it is assumed that the score y_{ij} is a linearly weighted score by entity j based on several factors. We shall temporarily assume that there are m factors, but it is precisely the point that the factors are to be retrieved in the mining process. A linear model, therefore, assumes the relationship

$$(1.2) \quad Y = AF,$$

where $A = [a_{ik}] \in \mathbb{R}^{n \times m}$ is a matrix with a_{ik} denoting the score of entity j on factor k , and $F = [f_{kj}] \in \mathbb{R}^{m \times \ell}$ is a matrix with f_{kj} denoting the score of factor k on variable i . Depending on the application, often only a portion of the matrices in (1.2) is known, and the goal is to retrieve the remaining information. The linear model (1.2) can have different interpretations. We sketch two scenarios below.

1. Assume that each of the ℓ columns of the observed matrix Y represents the transcript of a college student (an entity) at the end of freshman year on n fixed subjects (the variables). It is generally believed that a college freshman’s academic performance depends on a number of preexistent factors. Upon entering the college, each student could be asked to fill out a questionnaire inquiring about his or her background (scores) with respect to these factors. These responses are placed in the corresponding column of the scoring matrix F . What is not clear to the administrators in the first place is how to choose the factors to compose the questionnaire or how each of the chosen factors would be weighted (the loadings) to reflect their effect on each particular subject. In practice, we usually do not have a priori knowledge about the number and character of underlying factors in A . Sometimes we do not even know the factor scores in F . Only the data matrix Y is observable. Explaining the complex phenomena observed in Y with the help of a set of factors extracted from the data matrix is the primary and most important goal of factor analysis.

To collectively gather students’ academic records from wide variations (of grading rules and difficulties of subjects) for the purpose of analyzing some influential factors

in A , it is essential that all these scores be evaluated on the same basis. Otherwise, the factors cannot be correctly revealed. Statistically, the same basis of scores means the standardization (of rows) of Y . For our purpose, it is reasonable to assume that all sets of factors being considered are uncorrelated with each other. (See [22] for the case of stochastic independence.) As such, rows of F should be orthogonal to each other. We should further assume that the scores in F for each factor are normalized. Otherwise, one may complain that the factors being considered are not neutral to begin with and the analysis would be biased and misinformed. Under these assumptions, we have

$$(1.3) \quad \frac{1}{\ell} F F^\top = I_m,$$

where I_m stands for the identity matrix in $\mathbb{R}^{m \times m}$. It follows that the correlation matrix R can be expressed directly in terms of the loading matrix A alone, i.e.,

$$(1.4) \quad R = A A^\top.$$

Factor extraction now becomes a problem of decomposing the correlation matrix R into the product $A A^\top$ using as few factors as possible [19].

The k th column of A may be interpreted as correlations of the data variables with that particular k th factor. Those data variables with high factor loadings are considered to be more influenced by the factor in some sense, and those with zero or near-zero loadings are treated as being unlike the factor. The quality of this likelihood, indicating the k th factor, is measured by the norm of the k th column of A . One basic idea in factor analysis is to rewrite the loadings of variables over some newly selected factors so as to manifest more clearly the correlation between variables and factors [18, 22]. Those factors with smaller significance are likely to be dropped to reduce the number of effectual factors.

2. Assume now that the textual documents are collected in an $n \times m$ matrix $H = [h_{ik}]$ in $\mathbb{R}^{n \times m}$. Each document is represented by one row in H . The entry h_{ik} represents the weight of one particular term k in document i , whereas each term could be defined by just one single word or a string of phrases. A natural choice of the weight h_{ik} is by counting the number of times that the term k occurs in document i . Many more elaborate weighting schemes that yield better performance have been devised in the literature. See, for example, lists in [24, 28] and comparisons in [31]. One commonly used term-weighting scheme to enhance discrimination between various documents and to enhance retrieval effectiveness is to define

$$(1.5) \quad h_{ik} = t_{ik} g_k n_i,$$

where t_{ik} captures the relative importance of term k in document i , g_k weights the overall importance of term k in the entire set of documents, and

$$(1.6) \quad n_i = \left(\sum_{k=1}^m t_{ik} g_k \right)^{-1/2}$$

is the scaling factor for normalization. The normalization by (1.6) is necessary because, otherwise, one could artificially inflate the prominence of document i by padding it with repeated pages or volumes.

One usage of the indexing matrix is for information retrieval, particularly in the context of latent semantic indexing (LSI) application [2, 10, 17]. Each query is represented as a column vector $\mathbf{q}_j^\top = [q_{1j}, \dots, q_{mj}]$ in $\mathbb{R}^m$, where q_{kj} represents the weight of term k in the query j . Similar to (1.5), more elaborate schemes can also be employed to weight terms in a query. To measure how the query $\mathbf{q}_j$ matches the documents, we calculate the column vector

$$(1.7) \quad \mathbf{c}_j = H\mathbf{q}_j$$

and rank the relevance of documents to $\mathbf{q}_j$ according to the ℓ_2 norm in $\mathbf{c}_j$.

To put the notation in the context of linear model (1.2), we observe the following analogies between the two examples:

indexing matrix H	$\longleftrightarrow$	loading matrix A
document i	$\longleftrightarrow$	variable i
term k	$\longleftrightarrow$	factor k
one query $\mathbf{q}_j$	$\longleftrightarrow$	one column in scoring matrix F
weight h_{ik} of term k in document i	$\longleftrightarrow$	loading a_{ik} of factor k on variable i
weights q_{kj} of term k in query $\mathbf{q}_j$	$\longleftrightarrow$	response f_{kj} of entity j to factor k
rank $\mathbf{c}_{ij}$ of document i in query $\mathbf{q}_j$	$\longleftrightarrow$	score y_{ij} of variable i in entity j

It is important to note the different emphasis in the two examples. In Scenario 1 the data matrix Y is given and the factors in A are to be retrieved, while in Scenario 2 the terms in the indexing matrix H are predetermined and the scores $\mathbf{c}_j$ are to be calculated.

The computation involved in the LSI application is more on the vector-matrix multiplication (1.7) than on factor retrieval, provided that H is a “reasonable” representation of the relationship between documents and terms. In practice, however, the matrix H is never exact nor perfect. How to represent the indexing matrix and the queries in a more compact form so as to facilitate the computation of the scores therefore becomes a major challenge in the field [13, 27]. One idea, with the backing of both statistical analysis [6] and empirical testing [2], is to represent H by its low-rank approximation. A constrained low-rank approximation problem therefore arises. We shall point out in our numerical examples that the fidelity to H by merely normalizing a truncated singular value decomposition (SVD) is not as high as what our method can achieve.

One final point connecting Scenario 1 to Scenario 2 is worth mentioning. It is plausible that once we have accumulated enough experiences $C = [\mathbf{c}_1, \dots, \mathbf{c}_\ell]$ from queries $\mathbf{q}_1, \dots, \mathbf{q}_\ell$, we should be able to employ whatever factor retrieval techniques were developed for Scenario 1 to reconstruct terms (factors) for Scenario 2. That is, after the standardization of the scores, the information gathered in C may be used as feedback to the process of selecting newer and more suitable H . Recall from (1.4) that no reference to the queries is needed. This ℓ_2 norm with feedback from C can be iterated to reduce and improve the indexing matrix H .

2. Low-rank approximation. The linear model (1.2) can have a much broader interpretation than the two applications we have demonstrated. Its interpretation in Scenario 1 for factor analysis manifests the critical role that the loading matrix A will play. Our low-rank approximation problem can be seen as the need to retrieve a few principal factors to represent the original A . Likewise, the interpretation in

Scenario 2 for the information retrieval suggests that the indexing matrix H is playing an equally important role as the loading matrix A . Our low-rank approximation problem in the context of LSI has in mind to derive an approximation of H to allow fast computation. In this section, we introduce a notion that can be employed to deal with this algebraically constrained low-rank approximation problem.

The quality of the approximation by a lower rank matrix Z to A can be measured in different ways. In the following, we shall describe three kinds of measurements and use them to form our objective functions. For convenience, the notation $\text{diag}(M)$ henceforth refers to the diagonal matrix of the matrix M .

2.1. Fidelity test. Considering A as the indexing matrix in the LSI, one way to infer whether Z is a good (but low-rank) representation of A is to use documents (rows) in A as queries themselves to inquire about document information from Z . If the document-term relationships embedded in Z adhere to those in A , then the scores of the query $\mathbf{q}_j$ applied to Z , where $\mathbf{q}_j^\top$ is the j th row of A , should point to the most relevant document j . It should be so for each $j = 1, \dots, n$. In other words, in the product $ZA^\top$ the highest score per column should occur precisely at the diagonal entry. This notion of self-examination is referred to as the *self-examination* of Z to A .

Ideally, the fidelity test should produce the maximum value 1 along the diagonal. Since the indexing matrix A is never perfect, we may as well be settled with the collective effect for the sake of easy computation. We say that the approximation Z remains in “good” fidelity to the original A if the trace of $ZA^\top$ is sufficiently close to $\text{trace}(AA^\top) = n$. This low-rank approximation problem therefore becomes the problem of maximizing

$$(2.1) \quad \text{trace}(ZA^\top) = \langle Z, A \rangle,$$

where $\langle M, N \rangle$ stands for the Frobenius inner product of matrices M and N in $\mathbb{R}^{n \times m}$. The objective, in short, is to *maximize* the fidelity of Z to A while maintaining its low rank.

In order to characterize the properties that the matrix Z is of rank k with unit-length rows, we introduce the parametrization that

$$(2.2) \quad Z = (\text{diag}(USS^\top U^\top))^{-1/2} USV^\top,$$

where $U \in \mathcal{O}_n$, $S \in \mathbb{R}^k$, $V \in \mathcal{O}_m$, and $\mathcal{O}_n$ denotes the group of orthogonal matrices of size $n \times n$. The task now is equivalent to maximizing the functional

$$(2.3) \quad F(U, S, V) := \langle (\text{diag}(USS^\top U^\top))^{-1/2} USV^\top, A \rangle,$$

subject to the orthogonal conditions of U and V . In the above, we have used the same symbol $S \in \mathbb{R}^k$ to denote, without causing ambiguity, the “diagonal matrix” in $\mathbb{R}^{n \times m}$ whose first k diagonal entries are those of S . Observe that by limiting S to $\mathbb{R}^k$, the rank of the matrix $X = USV^\top \in \mathbb{R}^{n \times m}$ is guaranteed to be at most k , and is exactly k if S contains no zero entry.

The Fréchet derivative of F at (U, S, V) acting on any $(H, D, K) \in \mathbb{R}^{n \times n} \times \mathbb{R}^k \times \mathbb{R}^{m \times m}$ can be considered as the sum of three operations,

$$(2.4) \quad F'(U, S, V) \cdot (H, D, K) = \frac{\partial F}{\partial U} \cdot H + \frac{\partial F}{\partial S} \cdot D + \frac{\partial F}{\partial V} \cdot K,$$

where the operation $\Lambda \cdot \eta$ denotes the result of the action by the linear operator Λ on the η . For convenience, define

$$(2.5) \quad \alpha(U, S) := (\text{diag}(USS^\top U^\top))^{-1}.$$

We now calculate each action in (2.4) as follows.

First, it is easy to see that

$$(2.6) \quad \frac{\partial F}{\partial V} \cdot K = \langle \alpha(U, S)^{1/2} USK^\top, A \rangle.$$

Using the property that

$$(2.7) \quad \langle MN, P \rangle = \langle M, PN^\top \rangle = \langle N, M^\top P \rangle$$

for any matrices M , N , and P of compatible sizes, it follows from the Riesz representation theorem that, with respect to the Frobenius inner product, the partial gradient can be represented as

$$(2.8) \quad \frac{\partial F}{\partial V} = A^\top \alpha(U, S)^{1/2} US.$$

Observe next that

$$(2.9) \quad \frac{\partial F}{\partial S} \cdot D = \langle \alpha(U, S)^{1/2} UDV^\top, A \rangle + \left\langle \left(\frac{\partial \alpha(U, S)^{1/2}}{\partial S} \cdot D \right) USV^\top, A \right\rangle.$$

By the chain rule we have

$$(2.10) \quad \begin{aligned} \frac{\partial \alpha(U, S)^{1/2}}{\partial S} \cdot D &= \frac{1}{2} \alpha(U, S)^{-1/2} (-\alpha(U, S) \text{diag}(UDS^\top U^\top + USD^\top U^\top) \alpha(U, S)) \\ &= -\frac{1}{2} \alpha(U, S)^{3/2} \text{diag}(UDS^\top U^\top + USD^\top U^\top), \end{aligned}$$

where we have used the fact that diagonal matrices are commutable. Denoting the diagonal matrix

$$(2.11) \quad \omega(U, S, V) := \alpha(U, S)^{3/2} \text{diag}(AVS^\top U^\top),$$

and taking advantage of the fact that $\langle M, \text{diag}(N) \rangle = \langle \text{diag}(M), \text{diag}(N) \rangle$, the action in (2.9) can be expressed as

$$(2.12) \quad \begin{aligned} \frac{\partial F}{\partial S} \cdot D &= \langle D, U^\top \alpha(U, S)^{1/2} AV \rangle - \frac{1}{2} \langle UDS^\top U^\top + USD^\top U^\top, \omega(U, S, V) \rangle \\ &= \langle D, \text{diag}(U^\top \alpha(U, S)^{1/2} AV) \rangle - \langle D, \text{diag}(U^\top \omega(U, S, V) US) \rangle. \end{aligned}$$

We thus conclude that

$$(2.13) \quad \frac{\partial F}{\partial S} = \text{diag}_k(U^\top \alpha(U, S)^{1/2} AV) - \text{diag}_k(U^\top \omega(U, S, V) US),$$

where, for clarity, we use diag_k to denote the first k diagonal elements and to emphasize that the partial gradient $\frac{\partial F}{\partial S}$ is in fact a vector in $\mathbb{R}^k$.

Finally,

$$(2.14) \quad \frac{\partial F}{\partial U} \cdot H = \langle \alpha(U, S)^{1/2} HSV^\top, A \rangle + \left\langle \left(\frac{\partial \alpha(U, S)^{1/2}}{\partial U} \cdot H \right) USV^\top, A \right\rangle,$$

whereas by the chain rule again we know that

$$(2.15) \quad \begin{aligned} \frac{\partial \alpha(U, S)^{1/2}}{\partial U} \cdot H &= \frac{1}{2} \alpha(U, S)^{-1/2} (-\alpha(U, S) \operatorname{diag}(HSS^\top U^\top + USS^\top H^\top) \alpha(U, S)) \\ &= -\frac{1}{2} \alpha(U, S)^{3/2} \operatorname{diag}(HSS^\top U^\top + USS^\top H^\top). \end{aligned}$$

Together, we obtain

$$(2.16) \quad \begin{aligned} \frac{\partial F}{\partial U} \cdot H &= \langle H, \alpha(U, S)^{1/2} AVS^\top \rangle - \frac{1}{2} \langle HSS^\top U^\top + USS^\top H^\top, \omega(U, S, V) \rangle \\ &= \langle H, \alpha(U, S)^{1/2} AVS^\top \rangle - \langle H, \omega(U, S, V) USS^\top \rangle, \end{aligned}$$

and hence

$$(2.17) \quad \frac{\partial F}{\partial U} = \alpha(U, S)^{1/2} AVS^\top - \omega(U, S, V) USS^\top.$$

Formulas (2.8), (2.13), and (2.17) constitute the gradient

$$\nabla F(U, S, V) = \left\langle \frac{\partial F}{\partial U}, \frac{\partial F}{\partial S}, \frac{\partial F}{\partial V} \right\rangle$$

of the objection function F . Because our parameters (U, S, V) must come from the manifold $\mathcal{O}_n \times \mathbb{R}^k \times \mathcal{O}_m$, we want to know further the projection of ∇F into the feasible set.

By taking advantage of the product topology, the tangent space $\mathcal{T}_{(U, S, V)}(\mathcal{O}_n \times \mathbb{R}^k \times \mathcal{O}_m)$ of the product manifold $\mathcal{O}_n \times \mathbb{R}^k \times \mathcal{O}_m$ at $(U, S, V) \in \mathcal{O}_n \times \mathbb{R}^k \times \mathcal{O}_m$ can be decomposed as the product of tangent spaces, i.e.,

$$(2.18) \quad \mathcal{T}_{(U, S, V)}(\mathcal{O}_n \times \mathbb{R}^k \times \mathcal{O}_m) = \mathcal{T}_U \mathcal{O}_n \times \mathbb{R}^k \times \mathcal{T}_V \mathcal{O}_m.$$

The projection of $\nabla F(U, S, V)$ onto $\mathcal{T}_{(U, S, V)}(\mathcal{O}_n \times \mathbb{R}^k \times \mathcal{O}_m)$, therefore, is the product of the projections of $\frac{\partial F}{\partial U}$, $\frac{\partial F}{\partial S}$, and $\frac{\partial F}{\partial V}$ onto $\mathcal{T}_U \mathcal{O}_n$, $\mathbb{R}^k$, and $\mathcal{T}_V \mathcal{O}_m$, respectively. Each of the projections can easily be calculated by using the techniques developed in [7], which we summarize below.

Observe that any matrix $M \in \mathbb{R}^{n \times n}$ has a unique orthogonal splitting,

$$(2.19) \quad M = U \left\{ \frac{1}{2} (U^\top M - M^\top U) \right\} + U \left\{ \frac{1}{2} (U^\top M + M^\top U) \right\},$$

as the sum of elements from the tangent space $\mathcal{T}_U \mathcal{O}_n$ and the normal space $\mathcal{N}_U \mathcal{O}_n$. Thus the projection $\mathcal{P}_{\mathcal{T}_U \mathcal{O}_n}(M)$ of M onto $\mathcal{T}_U \mathcal{O}_n$ is given by the matrix

$$(2.20) \quad \mathcal{P}_{\mathcal{T}_U \mathcal{O}_n}(M) = U \left\{ \frac{1}{2} (U^\top M - M^\top U) \right\}.$$

Replacing M by $\frac{\partial F}{\partial U}$ in (2.20), we thus obtain the explicit formulation of the projection of $\frac{\partial F}{\partial U}$ onto the tangent space $\mathcal{T}_U \mathcal{O}_n$, that is,

$$\begin{aligned} \mathcal{P}_{\mathcal{T}_U \mathcal{O}_n} \left(\frac{\partial F}{\partial U} \right) &= \{ (XX^\top \omega(U, S, V) - \omega(U, S, V) XX^\top) \\ &\quad + (\alpha(U, S)^{1/2} AX^\top - XA^\top \alpha(U, S)^{1/2}) \} \frac{U}{2}. \end{aligned}$$

Similarly, the projection of $\frac{\partial F}{\partial V}$ onto tangent spaces $\mathcal{T}_V \mathcal{O}_m$ can be calculated. The projection of $\frac{\partial F}{\partial S}$ onto $\mathbb{R}^k$ is just itself. These projections characterize precisely the projected gradient of the objective function $F(U, S, V)$ on the manifold $\mathcal{O}_n \times \mathbb{R}^k \times \mathcal{O}_m$.

The explicit formulation of the project gradient serves as the first order optimality condition that any local solution must satisfy. Many iterative methods making use of projected gradient for constrained optimization are available in the literature [16, 21]. On the other hand, we find it most natural and convenient to use the dynamical system,

$$(2.21) \quad \begin{aligned} \frac{dU}{dt} &= \mathcal{P}_{\mathcal{T}_U \mathcal{O}_n} \left(\frac{\partial F}{\partial U} \right), \\ \frac{dS}{dt} &= \frac{\partial F}{\partial S}, \\ \frac{dV}{dt} &= \mathcal{P}_{\mathcal{T}_V \mathcal{O}_m} \left(\frac{\partial F}{\partial V} \right), \end{aligned}$$

to navigate a flow on the manifold $\mathcal{O}_n \times \mathbb{R}^k \times \mathcal{O}_m$. The flow, called the *steepest ascent flow*, for later reference, moves along the steepest ascent direction to maximize the objective functional F . Since (2.21) defines an ascent flow by an analytic vector field, by the well-known Lojasiewicz–Simon theorem [5, 25, 30] the flow converges to a single point (U_*, S_*, V_*) of equilibrium at which

$$(2.22) \quad Z_* = (\text{diag}(U_* S_* S_*^\top U_*^\top))^{-1/2} U_* S_* V_*^\top$$

is locally the best rank k approximation to A subject to constraint of unit row vectors.

2.2. Nearness test. In the context of LSI, each document is represented by a single point in the term space $\mathbb{R}^m$. It is natural to evaluate the quality of an approximation Z to A by measuring how far Z is from A . The low-rank approximation problem therefore becomes the problem of minimizing the functional

$$(2.23) \quad E(U, S, V) := \frac{1}{2} \langle \alpha(U, S)^{1/2} U S V^\top - A, \alpha(U, S)^{1/2} U S V^\top - A \rangle,$$

subject to the constraints that $U \in \mathcal{O}_n$, $S \in \mathbb{R}^k$, and $V \in \mathcal{O}_m$. This notion of finding low-rank matrices nearest to the given A is called the *nearest low-rank approximation*. Nearest low-rank approximation problems under different (linear) structures have been considered in [8, 13, 27]. By a linear structure, we mean that all matrices of the same structure form an affine subspace. Our structure, defined by the requirement that each row is of unit length, is not a linear structure.

We quickly point out the fact that

$$(2.24) \quad \frac{1}{2} \langle Z - A, Z - A \rangle = \frac{1}{2} (\langle Z, Z \rangle + \langle A, A \rangle) - \langle Z, A \rangle.$$

Note that $\langle Z, Z \rangle = \langle A, A \rangle = n$. Therefore, minimizing the functional $E(U, S, V)$ in (2.23) is equivalent to maximizing the functional $F(U, S, V)$ in (2.3). The fidelity test is exactly the same as the nearest test under the Frobenius inner product.

2.3. Absolute fidelity test. The data matrix A in general might contain negative entries. In this case, cancellations among terms in the trace calculation might inadvertently reduce the fidelity quality. To prevent cancellation, we might want

to consider $\langle \cdot, \cdot \rangle_{\text{diag}}$ the sum of squares of the diagonal elements of $ZA^\top$. The objective functional to be maximized is

$$(2.25) \quad G(U, S, V) := \frac{1}{2} \langle \text{diag}(\alpha(U, S)^{1/2} U S V^\top A^\top), \text{diag}(\alpha(U, S)^{1/2} U S V^\top A^\top) \rangle,$$

subject to the constraints that $U \in \mathcal{O}_n$, $S \in \mathbb{R}^k$, and $V \in \mathcal{O}_m$. We remark in passing that in a certain sense the absolute fidelity test has the effect of rewarding both anticorrelation and positive correlation between matrices A and Z . In certain applications, such as identifying genes with similar functionalities, detecting anticorrelated gene clusters sometimes is as important as detecting positively correlated genes [14].

Once again, denote the Fréchet derivative of G at (U, S, V) acting on any $(H, D, K) \in \mathbb{R}^{n \times n} \times \mathbb{R}^k \times \mathbb{R}^{m \times m}$ as

$$(2.26) \quad G'(U, S, V) \cdot (H, D, K) = \frac{\partial G}{\partial U} \cdot H + \frac{\partial G}{\partial S} \cdot D + \frac{\partial G}{\partial V} \cdot K.$$

We can calculate each partial derivative in (2.26) to obtain the gradient ∇G .

For convenience, denote

$$(2.27) \quad \theta(U, S, V) := \text{diag}(\alpha(U, S)^{1/2} U S V^\top A^\top).$$

Observe that $\text{diag}(\cdot)$ is a linear operator, so its derivative is just itself. It follows that

$$(2.28) \quad \frac{\partial G}{\partial V} \cdot K = \langle \alpha(U, S)^{1/2} U S K^\top A^\top, \theta(U, S, V) \rangle.$$

Using (2.7) and the fact that diagonal matrices commute, we see that

$$(2.29) \quad \frac{\partial G}{\partial V} = A^\top \theta(U, S, V)^\top \alpha(U, S)^{1/2} U S = A^\top \alpha(U, S) \text{diag}(U S V^\top A^\top) U S.$$

Likewise,

$$\begin{aligned} \frac{\partial G}{\partial S} \cdot D &= \langle \text{diag}(\alpha(U, S)^{1/2} U D V^\top A^\top), \theta(U, S, V) \rangle \\ &\quad + \left\langle \text{diag} \left(\left(\frac{\partial \alpha(U, S)^{1/2}}{\partial S} \cdot D \right) U S V^\top A^\top \right), \theta(U, S, V) \right\rangle, \end{aligned}$$

whereas the action $\frac{\partial \alpha(U, S)^{1/2}}{\partial S} \cdot D$ is already calculated in (2.10). Upon substitution, we obtain

$$\begin{aligned} \frac{\partial G}{\partial S} \cdot D &= \langle D, \text{diag}(U^\top \alpha(U, S)^{1/2} \theta(U, S, V) A V) \rangle \\ &\quad - \frac{1}{2} \langle \text{diag}(U D S^\top U^\top + U S D^\top U^\top), \alpha(U, S)^{3/2} \theta(U, S, V) \text{diag}(A V S^\top U^\top) \rangle \\ (2.30) \quad &= \langle D, \text{diag}(U^\top \alpha(U, S)^{1/2} \theta(U, S, V) A V) \rangle \\ &\quad - \langle D, \text{diag}(U^\top \alpha(U, S)^{3/2} \theta(U, S, V) \text{diag}(A V S^\top U^\top) U S) \rangle. \end{aligned}$$

We thus conclude that

$$(2.31) \quad \begin{aligned} \frac{\partial G}{\partial S} &= \text{diag}_k(U^\top \alpha(U, S)^{1/2} \theta(U, S, V) A V) \\ &\quad - \text{diag}_k(U^\top \alpha(U, S)^{3/2} \theta(U, S, V) \text{diag}(A V S^\top U^\top) U S). \end{aligned}$$

Finally,

$$\begin{aligned} \frac{\partial G}{\partial U} \cdot H &= \langle \text{diag}(\alpha(U, S)^{1/2} H S V^\top A^\top), \theta(U, S, V) \rangle \\ &\quad + \left\langle \text{diag} \left(\left(\frac{\partial \alpha(U, S)^{1/2}}{\partial U} \cdot H \right) U S V^\top A^\top \right), \theta(U, S, V) \right\rangle. \end{aligned}$$

Substituting the expression of $\frac{\partial \alpha(U, S)^{1/2}}{\partial U} \cdot H$ computed in (2.15) and simplifying, we see that

$$\begin{aligned} \frac{\partial G}{\partial U} \cdot H &= \langle H, \alpha(U, S)^{1/2} \theta(U, S, V) A V S^\top \rangle \\ &\quad - \frac{1}{2} \langle \text{diag}(H S S^\top U^\top + U S S^\top H^\top), \alpha(U, S)^{3/2} \theta(U, S, V) A V S^\top U^\top \rangle \\ &= \langle H, \alpha(U, S)^{1/2} \theta(U, S, V) A V S^\top \rangle \\ &\quad - \langle H, \alpha(U, S)^{3/2} \theta(U, S, V) \text{diag}(A V S^\top U^\top) U S S^\top \rangle \end{aligned}$$

and obtain

$$(2.32) \quad \begin{aligned} \frac{\partial G}{\partial U} &= \alpha(U, S)^{1/2} \theta(U, S, V) A V S^\top \\ &\quad - \alpha(U, S)^{3/2} \theta(U, S, V) \text{diag}(A V S^\top U^\top) U S S^\top. \end{aligned}$$

Formulas (2.29), (2.31), and (2.32) constitute the gradient,

$$\nabla G = \left\langle \frac{\partial G}{\partial U}, \frac{\partial G}{\partial S}, \frac{\partial G}{\partial V} \right\rangle,$$

of the objection function G in the general ambient space $\mathbb{R}^{n \times n} \times \mathbb{R}^k \times \mathbb{R}^{m \times m}$. Similar to the preceding section, we can express by using (2.20) the projected gradient of G in explicit form to establish the first order optimality condition. We can also formulate the problem (2.1) that evolves on the manifold $\mathcal{O}_n \times \mathbb{R}^k \times \mathcal{O}_m$ to maximize $G(U, S, V)$.

We conclude this section by pointing out that the formulas of the vector fields discussed above are not as computationally extensive as they appear to be. Many blocks of expressions repeat themselves in the formulas, whereas all radicals of matrices acquired involve only diagonal matrices.

3. Compact form and Stiefel manifold. The above discussion provides a basis for computing the low-rank approximation. We must notice, however, that the square matrices are employed just for the sake of deriving the formulas. A lot of information computed is not needed in defining Z in (2.2). It suffices to know only a portion of columns of $U \in \mathcal{O}_n$ and $V \in \mathcal{O}_m$, particularly when the desirable rank k of Z is assumed to be much smaller than m and n . Indeed, we may write Z precisely in the same way as (2.2) but assume that

$$(3.1) \quad U \in \mathcal{O}(n, k), \quad S \in \mathbb{R}^k, \quad \text{and} \quad V \in \mathcal{O}(m, k),$$

where

$$(3.2) \quad \mathcal{O}(p, q) := \{Q \in \mathbb{R}^{p \times q} \mid Q^\top Q = I_q\}$$

denotes the set of all $p \times q$ real matrices with orthonormal columns. This set, known as the Stiefel manifold, forms a smooth manifold [32].

The set $\mathcal{O}(p, q)$ enjoys many properties similar to those of the orthogonal group. Upon close examination, it can be checked that all expressions derived above for the gradients of $F(U, S, V)$ and $G(U, S, V)$ remain valid, even if we restrict U to $\mathcal{O}(n, k)$ and V to $\mathcal{O}(m, k)$. It only remains to compute the projected gradients onto the Stiefel manifolds. We outline in this section some main points for this purpose.

Embedding $\mathcal{O}(p, q)$ in the Euclidean space $\mathbb{R}^{p \times q}$ which is equipped with the Frobenius inner product, it is easy to see that any vector H in the tangent space $\mathcal{T}_Q \mathcal{O}(p, q)$ is necessarily of the form

$$(3.3) \quad H = QK + (I_p - QQ^\top)W,$$

where $K \in \mathbb{R}^{q \times q}$ and $W \in \mathbb{R}^{p \times q}$ are arbitrary, and K is skew-symmetric. Furthermore, the space $\mathbb{R}^{p \times q}$ can be written as the direct sum of three mutually perpendicular subspaces

$$(3.4) \quad \mathbb{R}^{p \times q} = Q\mathcal{S}(q) \oplus \mathcal{N}(Q^\top) \oplus Q\mathcal{S}(q)^\perp,$$

where $\mathcal{S}(q)$ is subspace of $q \times q$ symmetric matrices, $\mathcal{S}(q)^\perp$ is the subspace of $q \times q$ skew-symmetric matrices, and $\mathcal{N}(Q^\top) := \{X \in \mathbb{R}^{p \times q} \mid Q^\top X = 0\}$. Any $M \in \mathbb{R}^{p \times q}$ can be uniquely split as

$$(3.5) \quad M = Q \frac{Q^\top M - M^\top Q}{2} + (I - QQ^\top)M + Q \frac{Q^\top M + M^\top Q}{2}.$$

Similarly to (2.20), it follows that the projection $\mathcal{P}_{\mathcal{O}(p, q)}(M)$ of $M \in \mathbb{R}^{p \times q}$ onto the tangent space $\mathcal{T}_Q \mathcal{O}(p, q)$ is given by

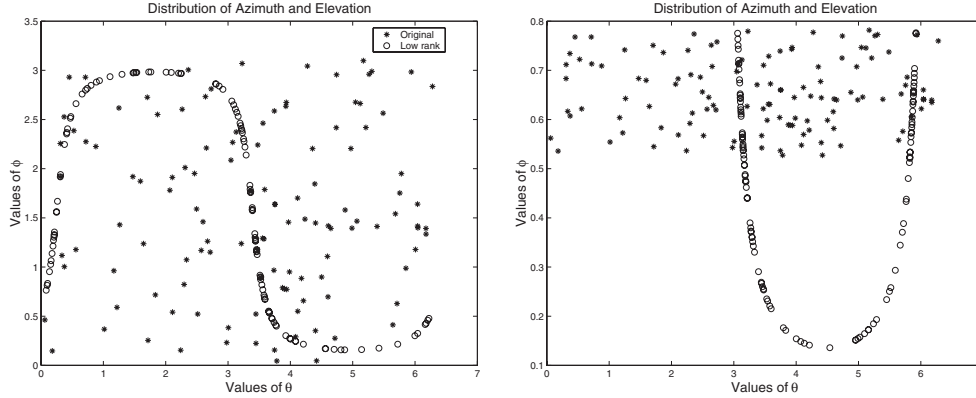
$$(3.6) \quad \mathcal{P}_{\mathcal{O}(p, q)}(M) = Q \frac{Q^\top M - M^\top Q}{2} + (I - QQ^\top)M.$$

Note that in case $p = q$ so that Q is orthogonal, the second term in (3.6) is identically zero and the above notion is identical to that of (2.20).

We may now project each of the partial gradients calculated in the preceding sections onto the appropriate Stiefel subspace according to (3.6) and obtain projected gradient flows. The gain is that the dimensionality is much reduced and the computation is much more cost efficient. For a given $A \in \mathbb{R}^{n \times m}$ and a desirable rank k , the gradient flow on the orthogonal group would involve $n^2 + k + m^2$ dependent variables, whereas the flow on the Stiefel manifold involves only $(n + m + 1)k$ variables.

Be cautious that because $\mathcal{O}(m, k)$ is a subspace of $\mathcal{O}_m$, the projections onto $\mathcal{O}(m, k)$ and $\mathcal{O}_m$ are different in general. The resulting dynamics of the gradient flows on the manifolds will behave differently. For computation, we prefer the compacted flows on the Stiefel manifolds.

4. Numerical experiments. The information of projected gradients can be utilized in various ways to tackle the low-rank approximation problem. See, for example, the review article [20] where a variety of Lie group methods, continuous or discrete, are discussed. For demonstration purposes, we report in this section some preliminary numerical results from using the gradient flow approach only. To make a distinction between the two gradient flows related to (2.3) and (2.25), we further denote the corresponding solutions by Z_F and Z_G , respectively. For convenience, we

FIG. 4.1. *Big circle approximation of points on the unit sphere S^2 .*

shall employ existing routines in MATLAB as the ODE integrators, even though more elaborate coding, especially the recently developed geometric integrators techniques, might be more effective. At the moment, our primary concern is to focus more on the dynamical behavior of the resulting flows than on the efficiency. Much more can be done to improve the efficiency.

The ODE Suite [29] in MATLAB contains several variable step-size solvers at our disposal. Considering the fact that the original data matrix A is not precise in its own right in practice, high accuracy approximation of A is not needed. We set both local tolerances $AbsTol = RelTol = 10^{-6}$ while maintaining all other parameters at the default values of the MATLAB codes.

1. To visualize the effect of data fitting by great circles, consider the case where $A \in \mathbb{R}^{n \times 3}$. These points on S^2 can be conveniently represented in $\mathbb{R}^2$ by their azimuths θ and elevations ϕ . A rank 2 approximation of A corresponds to points on the intersection of S^2 with a plane passing through the origin. The fidelity test is equivalent to finding the nearest great circle to the original points on S^2 in the sense of least squares.

In the left panel of Figure 4.1, we randomly generate $n = 120$ uniformly distributed angles $0 \leq \theta \leq 2\pi$ and $0 \leq \phi \leq \pi$. These angles are used as azimuths and elevations to produce the 120×3 matrix A . A total of 248 independent variables are involved in the dynamical system (2.21) to calculate the rank 2 approximation matrix Z_F , which then is converted into azimuths and elevations. The resulting more regular pattern (sinuous curve) of the angles for Z_F in the left panel of Figure 4.1 demonstrates the great circle approximation.

Suppose we limit the elevations to be between the cones $\frac{\pi}{6} \leq \phi \leq \frac{\pi}{4}$. These rows of A reside within a circular band in the northern hemisphere. It would be difficult to predict how a great circle would fit these data. In the right panel of Figure 4.1, it is interesting to observe that the azimuths θ on the best great circle approximation are distributed over only half of a circle.

2. Given a matrix A and a desirable low-rank k , it is known that the (globally) nearest approximation of rank k to A is given by the truncated SVD $A^{(k)}$ of A . However, $A^{(k)}$ generally does not satisfy the normalization condition. Let $Z_0^{(k)}$ denote the rowwise normalized $A^{(k)}$. This example demonstrates that the normalized truncated SVD $Z_0^{(k)}$ generally does not maintain the highest fidelity.

TABLE 4.1
Comparison of nearness of truncated SVD and fidelity flow.

Rank k	$\ A^{(k)} - A\ _2$	$\ A^{(k)} - A\ _F$	$\ Z_0^{(k)} - A\ _F$	$\ Z_*^{(k)} - A\ _F$	$\ Z_0^{(k)} - Z_*^{(k)}\ _F$
1	1.5521	2.7426	3.4472	3.3945	0.9227
2	1.4603	2.2612	2.6551	2.5330	2.1866
3	1.3056	1.7264	1.9086	1.8588	0.8085
4	1.1291	1.1296	1.1888	1.1744	0.3341
5	0.0237	0.0343	0.0343	0.0343	5.5969×10^{-7}
6	0.0183	0.0248	0.0248	0.0248	2.1831×10^{-7}
7	0.0141	0.0167	0.0167	0.0167	1.1083×10^{-7}
8	0.0083	0.0089	0.0089	0.0089	6.4180×10^{-9}
9	0.0034	0.0034	0.0034	0.0034	2.7252×10^{-9}

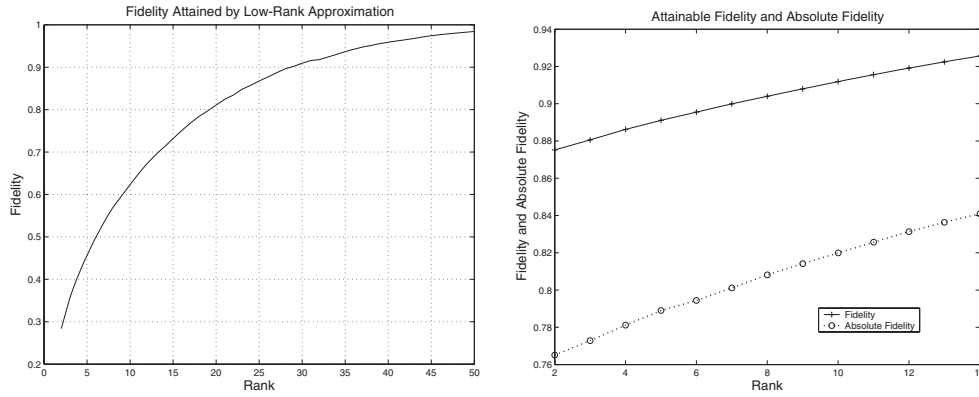


FIG. 4.2. Attainable fidelity with various rank values and initial values. Left: $A \in \mathbb{R}^{120 \times 50}$ and random initial values. Right: $A \in \mathbb{R}^{200 \times 50}$ and high fidelity initial values.

Let $Z_*^{(k)}$ denote the (nearly) limit point of the fidelity flow using $Z_0^{(k)}$ as the starting value. Table 4.1 compares the distances $\|A^{(k)} - A\|_F$, $\|Z_0^{(k)} - A\|_F$, $\|Z_*^{(k)} - A\|_F$, and $\|Z_0^{(k)} - Z_*^{(k)}\|_F$ of a 10×10 random matrix A with controlled singular values. For reference, we have also listed $\|A^{(k)} - A\|_2$, which is known to be the $(k+1)$ th singular value of A .

Because $\sum_{i=1}^{10} \sigma_i^2 = 10$, we purposefully control the singular values of A so that five singular values are relatively larger than the other five. We then notice that the gap $\|Z_0^{(k)} - Z_*^{(k)}\|_F$ is relatively significant up to $k = 4$. Since the local tolerance in the ODE integrator is set only at 10^{-6} , the difference between $Z_0^{(k)}$ and $Z_*^{(k)}$ is probably negligible when $k \geq 5$.

3. We measure the fidelity and the absolute fidelity by the ratios

$$(4.1) \quad r(Z_F) = \frac{\text{trace}(Z_F A^\top)}{\text{trace}(A A^\top)} = \frac{\text{trace}(Z_F A^\top)}{n},$$

$$(4.2) \quad q(Z_G) = \frac{\|\text{diag}(Z_G A^\top)\|_2^2}{\|\text{diag}(A A^\top)\|_2^2} = \frac{\|\text{diag}(Z_G A^\top)\|_2^2}{n},$$

respectively. The highest attainable value for either $r(Z_F)$ or $q(Z_G)$ should be 1.

The left panel in Figure 4.2 represents a typical attainable $r(Z_F)$ of a rank k

approximation Z_F to a matrix A of size 120×50 . For each $k = 2, \dots, 50$, we are solving a dynamical system of $171k$ many independent variables. These are fairly large-scale dynamical systems when k is large. Our proposed projected gradient flow approach thus connects to and can benefit from another interesting research subject—model reduction. See, for example, [1, 4] for an overview and many other studies in the literature. The right panel in Figure 4.2 compares the attainable $r(Z_F)$ and $q(Z_G)$ to a matrix A of size 200×50 with rank $k = 2, \dots, 14$. The plots in Figure 4.2 seem to suggest that the value of fidelity is an increasing function of k . However, such an observation is true only because we systematically modify the initial values one column a time. The different scales along the y -axis in the left and right panels clearly illustrate the effect of different initial values for the flows. The initial values for experiments involved in the left panel are randomly generated and are very far away from A , while those for experiments in the right panel are built to have at least 80% fidelity to A to begin with.

5. Conclusion and future work. Motivated by the fact that in many applications data in high-dimensional space are normalized to unit length, this paper touches upon the issue of best approximation by low-dimensional data while maintaining unit length. The problem is cast as a low-rank matrix approximation subject to row normalization. After proper parametrization of the feasible set, we offer explicit formulas for the project gradients which can be turned into a numerical means for computing the approximation.

We experiment with a projected gradient flow approach. The capacity of the differential equation approach is limited only by current development of numerical ODE techniques. Thus far, a direct ODE approach can handle matrices of sizes up to thousands within a reasonable frame of time, say, a few CPU minutes in a desktop computing environment. However, it may not be efficient enough to handle very large-scale problems, which often is the case in real applications. Other techniques such as model reduction might prove beneficial but are not discussed in the presentation. The availability of projected gradient in explicit form also makes it possible to apply other optimization techniques, including iterative methods. It might be worth further investigation in this direction. It is our hope that this study will have shed some light on the interesting low-rank approximation problem.

REFERENCES

- [1] A. C. ANTOUNAS AND D. C. SORESENSEN, *Approximation of large-scale dynamical systems: An overview*, Int. J. Appl. Math. Comput. Sci., 11 (2001), pp. 1093–1121.
- [2] M. W. BERRY, ED., *Computational Information Retrieval*, SIAM, Philadelphia, 2001.
- [3] J. P. BURG, D. G. LUENBERGER, AND D. L. WENGER, *Estimation of structured covariance matrices*, Proc. IEEE, 70 (1982), pp. 963–974.
- [4] Y. CHAHLAOUI, *Low-Rank Approximation and Model Reduction*, Ph.D. Thesis, Département d'ingénierie mathématique, Université Catholique De Louvain, 2003.
- [5] R. CHILL, *On the Łojasiewicz-Simon gradient inequality*, J. Funct. Anal., 201 (2003), pp. 572–601.
- [6] M. T. CHU, *On the Statistical Meaning of the Truncated Singular Decomposition*, preprint, North Carolina State University, Raleigh, NC, 2000.
- [7] M. T. CHU AND K. R. DRIESSEL, *The projected gradient method for least squares matrix approximations with spectral constraints*, SIAM J. Numer. Anal., 27 (1990), pp. 1050–1060.
- [8] M. T. CHU, R. E. FUNDERLIC, AND R. J. PLEMMONS, *Structured low-rank approximation*, Linear Algebra Appl., 366 (2003), pp. 157–172.
- [9] R. DE BEER, *Quantitative in vivo NMR (Nuclear Magnetic Resonance on Living Objects)*, available online from http://dutnsic.tn.tudelft.nl:8080/c59_to_html/c59.html, University of Technology–Delft, The Netherlands, 1995.

- [10] S. DEERWESTER, S. DUMAIS, G. FURNAS, T. LANDAUER, AND R. HARSHMAN, *Indexing by latent semantic analysis*, J. Amer. Soc. Inform. Sci., 41 (1990), pp. 391–407.
- [11] B. DE MOOR, *Total least squares for affinely structured matrices and the noisy realization problem*, IEEE Trans. Signal Process., 42 (1994), pp. 3104–3113.
- [12] M. DENDRINOS, S. BAKAMIDIS, AND G. CARAYANNIS, *Speech enhancement from noise: A regenerative approach*, Speech Commun., 10 (1991), pp. 45–57.
- [13] I. S. DHILLON AND D. M. MODHA, *Concept decompositions for large sparse text data using clustering*, Machine Learning, 42 (2001), pp. 143–175.
- [14] I. S. DHILLON, E. M. MARCOTTE, AND U. ROSHAN, *Diametrical clustering for identifying anti-correlated gene clusters*, Bioinformatics, 19 (2003), pp. 1612–1619.
- [15] R. O. DUDA, P. E. HART, AND D. G. STORK, *Pattern Classification*, Wiley, New York, 2001.
- [16] P. E. GILL, W. MURRAY, AND M. H. WRIGHT, *Practical Optimization*, Academic Press, London, 1981.
- [17] T. HASTIE, R. TIBSHIRANI, AND J. FRIEDMAN, *The Elements of Statistical Learning: Data Mining, Inference, and Prediction*, Springer-Verlag, New York, 2001.
- [18] P. HORST, *Factor Analysis of Data Matrices*, Holt, Rinehart and Winston, New York, 1965.
- [19] L. HUBERT, J. MEULMAN, AND W. HEISER, *Two purposes for matrix factorization: A historical appraisal*, SIAM Rev., 42 (2000), pp. 68–82.
- [20] A. ISERLES, H. Z. MUNTHE-KASS, S. P. NØRSETT, AND A. ZANNA, *Lie-group methods*, in Acta Numer., 2000, Cambridge University Press, Cambridge, UK, 2000, pp. 215–365.
- [21] A. N. IUSEM, *On the convergence properties of the projected gradient method for convex optimization*, Comput. Appl. Math., 22 (2003), pp. 37–52.
- [22] I. T. JOLLIFFE, *Principal Component Analysis*, 2nd ed., Springer-Verlag, New York, 2002.
- [23] J. KLEINBERG, C. PAPADIMITRIOU, AND P. RAGHAVAN, *A microeconomic view of data mining*, Data Min. Knowl. Discov., 2 (1998), pp. 311–324.
- [24] T. G. KOLDA AND D. P. O’LEARY, *A semi-discrete matrix decomposition for latent semantic indexing in information retrieval*, ACM Trans. Inform. Systems, 16 (1998), pp. 322–346.
- [25] S. LOJASIEWICZ, *Une propriété topologique des sous-ensembles analytiques réels*, in Les Équations aux Dérivées Partielles (Paris, 1962), Vol. 117, Éditions du Centre National de la Recherche Scientifique, Paris, 1963, pp. 87–89.
- [26] P. PAATERO, *Least squares formulation of robust nonnegative factor analysis*, Chemomet. Intell. Lab. Systems, 37 (1997), pp. 23–35.
- [27] H. PARK, M. JEON, AND J. B. ROSEN, *Lower dimensional representation of text data in vector space based information retrieval*, in Computational Information Retrieval, M. W. Berry, ed., SIAM, Philadelphia, 2001, pp. 3–23.
- [28] G. SALTON AND M. MCGILL, *Introduction to Modern Information Retrieval*, McGraw-Hill, New York, 1983.
- [29] L. F. SHAMPINE AND M. W. REICHEL, *The MATLAB ODE suite*, SIAM J. Sci. Comput., 18 (1997), pp. 1–22.
- [30] L. SIMON, *Asymptotics for a class of nonlinear evolution equations with applications to geometric problems*, Ann. of Math. (2), 118 (1983), pp. 525–571.
- [31] A. SINGHAL, C. BUCKLEY, AND M. MITRA, *Pivoted document length normalization*, in Proceedings of the 19th ACM SIGIR Conference on Research and Development in Information Retrieval, ACM, New York, 1996, pp. 21–29.
- [32] E. STIEFEL, *Richtungsfelder und fernparallelismus in n-dimensionalen mannigfaltigkeiten*, Comment. Math. Helv., 8 (1935-1936), pp. 305–353.
- [33] P. STOICA AND M. VIBERG, *Maximum likelihood parameter and rank estimation in reduced rank multivariate linear regressions*, IEEE Trans. Signal Process., 44 (1996), pp. 3069–3078.
- [34] J. TROPP, I. DHILLON, R. HEATH, AND T. STROHMER, *Designing structured tight frames via an alternating projection method*, IEEE Trans. Inform. Theory, 51 (2005), pp. 188–209.

GENERALIZED FINITE ALGORITHMS FOR CONSTRUCTING HERMITIAN MATRICES WITH PRESCRIBED DIAGONAL AND SPECTRUM*

INDERJIT S. DHILLON^{†¶}, ROBERT W. HEATH JR.[‡], MÁTYÁS A. SUSTIK[†], AND
JOEL A. TROPP[§]

Abstract. In this paper, we present new algorithms that can replace the diagonal entries of a Hermitian matrix by any set of diagonal entries that majorize the original set without altering the eigenvalues of the matrix. They perform this feat by applying a sequence of $(N - 1)$ or fewer plane rotations, where N is the dimension of the matrix. Both the Bendel–Mickey and the Chan–Li algorithms are special cases of the proposed procedures. Using the fact that a positive semidefinite matrix can always be factored as X^*X , we also provide more efficient versions of the algorithms that can directly construct factors with specified singular values and column norms. We conclude with some open problems related to the construction of Hermitian matrices with joint diagonal and spectral properties.

Key words. Hermitian matrices, majorization, Schur–Horn theorem, inverse eigenvalue problem, plane rotations, finite-step algorithms

AMS subject classifications. 15A18, 15A57, 65F18

DOI. 10.1137/S0895479803438183

1. Introduction. It is sometimes of interest to construct a collection of Hermitian matrices that have specified diagonal elements and eigenvalues. When all the eigenvalues are nonnegative, the problem is essentially equivalent to constructing a collection of rectangular matrices with specified column norms and singular values. In particular, if a rectangular matrix X has requirements on its singular values and squared column norms, the Hermitian matrix X^*X has a corresponding requirement on its eigenvalues and diagonal entries.

A specific example of this problem is to construct Hermitian matrices with unit diagonal and prescribed nonnegative eigenvalues. Such matrices are called correlation matrices—Davies and Higham [6] discuss several applications that require such matrices, ranging from the generation of test matrices for eigenvalue solvers to the design of statistical experiments. A related matrix construction problem has also arisen in connection with wireless communications. It turns out that $d \times N$ matrices, $d < N$, with d identical nonzero singular values and with prescribed column norms satisfy a certain “sum capacity” bound and “minimum squared correlation” property that is important in wireless applications. These matrices only exist if a majorization condition holds, as discussed in section 2.1. For a quick introduction to squared correlation; see the appendix in [7]. Refer to [20, 22] for details on how the problem arises in wireless applications.

*Received by the editors November 21, 2003; accepted for publication (in revised form) by U. Helmke September 24, 2004; published electronically June 22, 2005.

<http://www.siam.org/journals/simax/27-1/43818.html>

[†]Department of Computer Sciences, The University of Texas, Austin, TX 78712 (inderjit@cs.utexas.edu, sustik@cs.utexas.edu).

[‡]Department of Electrical and Computer Engineering, The University of Texas, Austin, TX 78712 (rheath@ece.utexas.edu).

[§]Institute for Computational Engineering and Sciences (ICES), The University of Texas, Austin, TX 78712 (jtropp@ices.utexas.edu).

[¶]The research of this author was supported by NSF CAREER Award ACI-0093404.

Two finite step techniques, the Bendel–Mickey [2] and Chan–Li [4] algorithms, are available for special cases. Both algorithms apply a sequence of plane rotations to an initial matrix that change its diagonal entries while preserving its spectrum. The Chan–Li algorithm starts with the diagonal matrix of eigenvalues and can reach a real, symmetric matrix with a specified majorizing diagonal. On the other hand, the Bendel–Mickey algorithm can start with an arbitrary Hermitian matrix and transform it to a Hermitian matrix with $\lambda_1, \dots, \lambda_N$ diagonal entries.

In this paper, we present new algorithms that generalize the Chan–Li and Bendel–Mickey procedures so that we can start with an arbitrary Hermitian matrix and change its diagonal entries to specified values while retaining its original spectrum. The only requirement is that the new diagonal elements majorize (in essence, average) the original ones. Thus our generalized algorithms permit us to construct a class of Hermitian matrices satisfying spectral and diagonal constraints that is richer than the collection yielded by earlier algorithms.

We now give a brief outline of the paper. In section 2, we provide the necessary background and summarize previous results. We present our generalized algorithms in section 3, and section 4 contains some numerical examples. We conclude in section 5 with some open problems.

2. Background and related work.

2.1. Majorization. Majorization is a relation between vectors that appears in a striking number of apparently unrelated contexts. Lorenz originally developed the ordering for econometrics, where he used it to compare the equitability of income distributions [1]. An intuitive definition is that one vector majorizes another if the former has “more average” entries than the latter. Let us make this notion precise.

DEFINITION 2.1. Let $\mathbf{a} = (a_1, \dots, a_N) \in \mathbb{R}^N$ and $\mathbf{z} = (z_1, \dots, z_N) \in \mathbb{R}^N$. Let $a_{(k)}$ and $z_{(k)}$ denote the k th order statistic of $\mathbf{a}$ and $\mathbf{z}$, respectively.

DEFINITION 2.2. Let $\mathbf{a} = (a_1, \dots, a_N) \in \mathbb{R}^N$ and $\mathbf{z} = (z_1, \dots, z_N) \in \mathbb{R}^N$. We say that $\mathbf{z}$ majorizes $\mathbf{a}$, denoted $\mathbf{z} \succcurlyeq \mathbf{a}$, if

$$\begin{aligned} a_{(1)} &\leq z_{(1)}, \\ a_{(1)} + a_{(2)} &\leq z_{(1)} + z_{(2)}, \end{aligned}$$

$$\begin{aligned} a_{(1)} + a_{(2)} + \dots + a_{(N-1)} &\leq z_{(1)} + z_{(2)} + \dots + z_{(N-1)}, \\ a_{(1)} + a_{(2)} + \dots + a_{(N)} &= z_{(1)} + z_{(2)} + \dots + z_{(N)}. \end{aligned}$$

We say that $\mathbf{z}$ strictly majorizes $\mathbf{a}$, denoted $\mathbf{z} \succ \mathbf{a}$, if $\mathbf{z} \succcurlyeq \mathbf{a}$ and $\mathbf{z} \neq \mathbf{a}$. We say that $\mathbf{z}$ majorizes $\mathbf{a}$ in the weak sense, denoted $\mathbf{z} \succcurlyeq_w \mathbf{a}$, if $\mathbf{z} \succcurlyeq \mathbf{a}$ and $\mathbf{z} \neq \mathbf{a}$.

The inequality in the above definition is reversed in some treatments, but we follow the convention in Horn and Johnson’s book [13, p. 192]. An equivalent definition is that $\mathbf{z} \succcurlyeq \mathbf{a}$ if and only if $\mathbf{z} = M\mathbf{a}$ for some doubly-stochastic matrix M . Birkhoff’s theorem states that the collection of doubly-stochastic matrices of size N is identical with the convex hull of the permutation matrices having size N . It follows that those vectors which majorize a fixed vector form a compact, convex set; see [13, p. 197 and p. 527] and [18] for more details.

It is easy to verify that majorization is antisymmetric and transitive on $\mathbb{R}^N$. But the majorization relation is reflexive, and hence a partial ordering, only when we restrict our attention to nondecreasing vectors in $\mathbb{R}^N$ [3].

Majorization plays a role on our stage because it defines the precise relationship between the diagonal entries and eigenvalues of a Hermitian matrix.

THEOREM 2.3 (Schur–Horn [13, pp. 193–196]).

$$a \succcurlyeq \lambda \iff \text{there exists a Hermitian matrix } A \text{ with diagonal } a \text{ and eigenvalues } \lambda$$

I. Schur demonstrated the necessity of the majorization condition in 1923, and A. Horn proved the converse some thirty years later [12]. Horn’s original proof is quite complicated and a small cottage industry has grown up to produce simpler, more constructive arguments; see, for example, [4, 5, 16]. A comprehensive reference on majorization is Marshall and Olkin’s monograph [18].

2.2. Some posets. First, we define some concepts related to partial orderings, and then we develop some new partial orderings on Hermitian matrices that are closely related to the matrix construction problem.

DEFINITION 2.4. Let S be a set. $(S, \succcurlyeq)$ is a poset if

$$(S, \succcurlyeq) \text{ is a poset if } a, b \in S \text{ are comparable, } a \succcurlyeq b \implies b \succcurlyeq a \text{ and } a \succcurlyeq a \text{ (chain).}$$

We may equip any poset with the ordering topology, where each basic open set is given by $\{a \neq b : b \succcurlyeq a\}$ for some point b . This is the minimal topology in which the ordering is continuous [15, p. 57].

Observe that vectors are comparable by majorization only when their entries have the same sum. Let $\mathcal{C}_\alpha$ denote the set of N -dimensional nondecreasing vectors whose entries sum to α . Every $\mathcal{C}_\alpha$ has a unique majorization-maximal element: the constant vector with entries α/N . On the other hand, there are no minimal vectors under the majorization relation; every z succeeds an infinite number of other vectors.

DEFINITION 2.5. Let A, B be Hermitian matrices.

$$A \equiv B \iff [A] = [B] \text{ (where } [A] \text{ is the diagonal of } A \text{)}$$

We indicate the collection of Schur–Horn equivalence classes by $\mathcal{H}$. Notice that the members of $\mathcal{H}$ vary significantly. For example, the Schur–Horn equivalence class of a diagonal matrix is the set of diagonal matrices with the same entries in permuted order. Meanwhile, the equivalence class of a matrix with unit diagonal and nonnegative eigenvalues λ is the set of “correlation matrices” that have spectrum λ [6, 17]. Even though similarity transformations preserve the eigenvalues of a Hermitian matrix, very few simultaneously preserve the diagonal. Therefore, Schur–Horn equivalence classes are not stable under most transformations. Exceptions include symmetric permutations and diagonal similarity transforms.

DEFINITION 2.6. Let A, Z be Hermitian matrices.

$$[Z] \succcurlyeq [A] \iff \text{diag } Z \succcurlyeq \text{diag } A$$

It is not hard to check that this construction yields a well-defined partial ordering on $\mathcal{H}$. Clearly, two Schur–Horn equivalence classes are comparable only if their members have the same spectrum. Suppose that the entries of $\lambda \in \mathbb{R}^N$ already occur in nondecreasing order, viz. $\lambda_k = \lambda_{(k)}$ for each k . Then we may write $\mathcal{H}_\lambda$ to denote the elements of $\mathcal{H}$ with spectrum λ . Each $\mathcal{H}_\lambda$ forms an isolated component of the poset $(\mathcal{H}, \succcurlyeq)$, and it has a unique maximal element: the equivalence class of matrices with eigenvalues λ and with a constant diagonal. A significant difference between majorization and the matrix ordering is that every chain under the matrix ordering has a minimal element: $[\text{diag } \lambda]$, where λ lists the (common) eigenvalues of the members

of the chain.

2.3. Algorithms. Now we discuss two algorithms which have been proposed for constructing Hermitian matrices with diagonal and spectral properties. In what follows, we use $\mathbb{M}_N$ to denote the set of complex $N \times N$ matrices and $\mathbb{M}_{d,N}$ to denote the set of complex $d \times N$ matrices.

The Bendel–Mickey algorithm produces random (Hermitian) correlation matrices with given spectrum [2]. Suppose that $A \in \mathbb{M}_N$ is a Hermitian matrix with $\text{Tr } A = N$. If A does not have a unit diagonal, we can locate two diagonal elements so that $A_{jj} < 1 < A_{kk}$; otherwise, the trace condition would be violated. It is then possible to construct a real rotation Q in the jk -plane for which $(Q^*AQ)_{jj} = 1$. The transformation $A \mapsto Q^*AQ$ preserves the conjugate symmetry and the spectrum of A , but it reduces the number of nonunit diagonal entries by at least one. Therefore, at most $(N - 1)$ rotations are required before the resulting matrix has a unit diagonal. If the output matrix is Z , it follows that $[Z] \succcurlyeq [A]$. Indeed, $[Z]$ is the unique $\succcurlyeq$ -maximal element in every chain that contains $[A]$.

The Chan–Li algorithm, on the other hand, was developed as a constructive proof of the Schur–Horn theorem [4]. Suppose that $\mathbf{a} \succcurlyeq \boldsymbol{\lambda}$. The Chan–Li algorithm begins with the diagonal matrix $\Lambda \stackrel{\text{def}}{=} \text{diag } \boldsymbol{\lambda}$. Then it applies a sequence of $(N - 1)$ cleverly chosen (real) plane rotations to generate a real, symmetric matrix A with the same eigenvalues as Λ but with diagonal entries listed by $\mathbf{a}$. Once again, the output and input satisfy the relationship $[A] \succcurlyeq [\Lambda]$. While the Bendel–Mickey algorithm starts from any element of a chain and moves to the top; the Chan–Li algorithm starts at the bottom of a chain and moves upward.

The Bendel–Mickey algorithm is a surjective map from the set of Hermitian matrices with spectrum $\boldsymbol{\lambda}$ onto the set of correlation matrices with spectrum $\boldsymbol{\lambda}$. If the initial matrix is chosen uniformly at random (which may be accomplished with standard techniques [19]), the result may be construed as a random correlation matrix. The distribution of the output, however, is unknown [11]. On the other hand, due to the special form of the initial matrix and the rigid choice of rotations, the Chan–Li algorithm cannot construct very many distinct matrices with a specified diagonal. It would be interesting to develop a procedure which can calculate every member of a given equivalence class; see section 5 for a brief discussion.

A brief discussion of how to use plane rotations to equalize the diagonal entries of a Hermitian matrix appears on page 77 of Horn and Johnson [13]. Problems 8.4.1 and 8.4.2 of Golub and Van Loan outline the Bendel–Mickey algorithm [10].

Davies and Higham present a numerically stable version of the Bendel–Mickey algorithm in their article [6]. Other references on this topic include [14, 23].

3. Generalized algorithms. We propose methods that generalize the Bendel–Mickey and Chan–Li algorithms. Like them, our techniques use a sequence of $(N - 1)$ or fewer plane rotations to move upward between two points in a chain. The crux of the matter is the strategy for selecting the planes of rotation. The two methods we present can be viewed, respectively, as direct generalizations of the Chan–Li strategy and the Bendel–Mickey strategy. Unlike the earlier algorithms, these new techniques do not require ending at the top of a chain like Bendel–Mickey nor starting at the bottom like Chan–Li. Therefore, our techniques allow the construction of a much larger set of matrices than the Chan–Li algorithm, while retaining its ability to select the final diagonal entries.

3.1. Plane rotations. First, we discuss how to use a plane rotation to modify the diagonal entries while preserving the spectrum. Let us suppose that A is a 2×2 matrix with diagonal $\mathbf{a}$ that we wish to transform to $\mathbf{z}$, where $\mathbf{z} \succcurlyeq \mathbf{a}$. Without loss of generality, we can assume that $a_1 \leq z_1 \leq z_2 \leq a_2$. We will explicitly construct a real plane rotation Q so that the diagonal of Q^*AQ equals $\mathbf{z}$. Recall that a two-dimensional plane rotation is an orthogonal matrix of the form

$$Q = \begin{bmatrix} c & s \\ -s & c \end{bmatrix},$$

where $c^2 + s^2 = 1$ [10]. The desired plane rotation yields the matrix equation

$$\begin{bmatrix} c & s \\ -s & c \end{bmatrix}^* \begin{bmatrix} a_1 & a_{21}^* \\ a_{21} & a_2 \end{bmatrix} \begin{bmatrix} c & s \\ -s & c \end{bmatrix} = \begin{bmatrix} z_1 & z_{21}^* \\ z_{21} & \tilde{z}_2 \end{bmatrix}.$$

The equality of the upper-left entries can be stated as

$$c^2 a_1 - 2sc \operatorname{Re} a_{21} + s^2 a_2 = z_1.$$

This equation is quadratic in $t \stackrel{\text{def}}{=} s/c$,

$$(3.1) \quad (a_2 - z_1)t^2 - 2t \operatorname{Re} a_{21} + (a_1 - z_1) = 0,$$

whence

$$(3.2) \quad t = \frac{\operatorname{Re} a_{21} \pm \sqrt{(\operatorname{Re} a_{21})^2 - (a_1 - z_1)(a_2 - z_1)}}{a_2 - z_1}.$$

Notice that the discriminant is nonnegative due to the majorization condition. The $\pm$ sign in (3.2) is taken to avoid sign cancellations with $\operatorname{Re} a_{21}$. If necessary, we can extract the other root of (3.1) using the fact that the product of its roots equals $(a_1 - z_1)/(a_2 - z_1)$. Finally, determine the parameters of the rotation using

$$(3.3) \quad c = \frac{1}{\sqrt{1+t^2}} \quad \text{and} \quad s = ct.$$

Floating-point arithmetic is inexact so the rotation may not yield $(Q^*AQ)_{11} = z_1$. A better implementation sets this entry to z_1 explicitly. Davies and Higham have shown that this method of computing rotations is numerically stable [6]. Since Q is orthogonal, Q^*AQ preserves the spectrum of A but replaces its diagonal with $\mathbf{z}$.

3.2. Generalized Bendel–Mickey. Let $\mathbf{z}$ and $\mathbf{a}$ be N -dimensional vectors for which $\mathbf{z} \succcurlyeq \mathbf{a}$. We will show how to transform a Hermitian matrix A with diagonal $\mathbf{a}$ and spectrum $\boldsymbol{\lambda}$ into one with diagonal $\mathbf{z}$ and spectrum $\boldsymbol{\lambda}$ using a sequence of plane rotations. It is enough to prove the result when the components of $\mathbf{a}$ and $\mathbf{z}$ are sorted in ascending order, so we place that restriction in the sequel.

Suppose that $\mathbf{a} \neq \mathbf{z}$. On account of the majorization relationship, it is possible to select indices $i < j$ that satisfy two properties: $a_i < z_i \leq z_j < a_j$ and $a_k = z_k$ for all k strictly between i and j . If $z_i - a_i \leq a_j - z_j$, then we construct a plane rotation Q in the (i, j) -plane such that $(Q^*AQ)_{ii} = z_i$. Otherwise, we find Q such that $(Q^*AQ)_{jj} = z_j$. Either rotation can be calculated using appropriate versions of (3.2) and (3.3). To see that this strategy can be repeated, we just need to check that $\mathbf{z}$ majorizes the diagonal of Q^*AQ . In the first case, the plane rotation transforms a_i

to z_i and a_j to $a_i + a_j - z_i$, while the remaining diagonal entries do not change. Since $a_i < z_i \leq z_j \leq a_i + a_j - z_i < a_j$ the diagonal entries of Q^*AQ remain in ascending order. The first $(i-1)$ majorization conditions are clearly unaffected. Notice that

$$\sum_{\ell=1}^{i-1} a_\ell + z_i \leq \sum_{\ell=1}^{i-1} z_\ell + z_i,$$

which proves the i th majorization condition. The next $(j-i-1)$ majorization inequalities follow in consequence of a_k being equal to z_k whenever $i < k < j$. The rest of the majorization conditions hold since

$$\sum_{\ell=1}^{i-1} a_\ell + z_i + \sum_{k=i+1}^{j-1} a_k + (a_i + a_j - z_i) = \sum_{\ell=1}^j a_\ell \leq \sum_{\ell=1}^j z_\ell.$$

The argument in the case when $z_i - a_i > a_j - z_j$ is similar. It follows that our rotation strategy may be applied until $\text{diag } A = \mathbf{z}$. This proof leads to the following algorithm.

ALGORITHM 1 (Generalized Bendel–Mickey). *Given* A , $N \times N$ Hermitian, $\mathbf{a}$, N -dimensional, $\mathbf{z} \succcurlyeq \mathbf{a}$.

Output A with $\text{diag } A = \mathbf{z}$.
 1. Find $i < j$ for which $a_i < z_i$, $z_j < a_j$ and $a_k = z_k$ for $i < k < j$ (in our implementation we pick the smallest such i). If no such pair exists, we are either done ($\mathbf{z} = \mathbf{a}$) or the majorization condition is violated.

2. Construct a plane rotation Q in the (i, j) -plane using (3.2) and (3.3) with appropriate modifications to transform a_i to z_i in the case $z_i - a_i \leq a_j - z_j$ or transform a_j to z_j otherwise.

3. Replace A by Q^*AQ .

4. Repeat Steps 1–3 until the diagonal is transformed to $\mathbf{z}$.

This algorithm requires about $12N^2$ real floating-point operations if conjugate symmetry is exploited. The storage requirement is about $N(N+1)/2$ complex floating-point numbers.

3.3. Generalized Chan–Li. Distinct algorithms arise by changing the strategy for selecting the planes of rotation. Let $\mathbf{z}$ and $\mathbf{a}$ be N -dimensional vectors for which $\mathbf{z} \succcurlyeq \mathbf{a}$. As before, we assume that they are sorted in ascending order, and suppose that A is a Hermitian matrix with diagonal $\mathbf{a}$. We now exhibit a different method for transforming the diagonal of A to $\mathbf{z}$ while preserving its eigenvalues. It can be viewed as a generalization of the Chan–Li algorithm [4].

We will use induction on the dimension, so grant us for a moment that we can perform the advertised feat on Hermitian matrices of size $(N-1)$. Now we consider N -dimensional vectors for which $\mathbf{z} \succcurlyeq \mathbf{a}$, and suppose that $\text{diag } A = \mathbf{a}$. The majorization condition implies that $a_1 \leq z_1 \leq z_N \leq a_N$, so it is always possible to select a least integer $j > 1$ so that $a_{j-1} \leq z_1 \leq a_j$. Let P_1 be a permutation matrix for which

$$\text{diag}(P_1^*AP_1) = (a_1, a_j, a_2, \dots, a_{j-1}, a_{j+1}, \dots, a_N).$$

Observe that $a_1 \leq z_1 \leq a_j$ and $a_1 \leq a_1 + a_j - z_1 \leq a_j$. Thus we modify (3.2) and (3.3) to construct a two-dimensional plane rotation Q_2 that sets the upper-left entry of

$$Q_2^* \begin{bmatrix} a_1 & a_{j1}^* \\ a_{j1} & a_j \end{bmatrix} Q_2$$

to z_1 . If we define the rotation

$$P_2 \stackrel{\text{def}}{=} \begin{bmatrix} Q_2 & \mathbf{0}^* \\ \mathbf{0} & I_{N-2} \end{bmatrix},$$

then

$$P_2^* P_1^* A P_1 P_2 = \begin{bmatrix} z_1 & \mathbf{v}^* \\ \mathbf{v} & A_{N-1} \end{bmatrix},$$

where $\mathbf{v}$ is an appropriate vector and A_{N-1} is an appropriate submatrix with

$$\text{diag}(A_{N-1}) = (a_1 + a_j - z_1, a_2, \dots, a_{j-1}, a_{j+1}, \dots, a_N).$$

In order to apply the induction hypothesis, it remains to check that the vector $(z_2, z_3, \dots, z_N)$ majorizes the diagonal of A_{N-1} . We accomplish this in three steps. First, recall that $a_k \leq z_1$ for $k = 2, \dots, j-1$. Therefore,

$$\sum_{k=2}^m z_k \geq (m-1) z_1 \geq \sum_{k=2}^m a_k$$

for each $m = 2, \dots, j-1$. The sum on the right-hand side obviously exceeds the sum of the smallest $(m-1)$ entries of the vector $\text{diag } A_{N-1}$, so the first $(j-2)$ majorization inequalities are in force. Second, we use the fact that $\mathbf{z} \succcurlyeq \mathbf{a}$ to calculate that for $m = j, \dots, N$,

$$\sum_{k=2}^m z_k = \sum_{k=1}^m z_k - z_1 \geq \sum_{k=1}^m a_k - z_1 = (a_1 + a_j - z_1) + \sum_{k=2}^{j-1} a_k + \sum_{k=j+1}^m a_k.$$

Once again, observe that the sum on the right-hand side exceeds the sum of the smallest $(m-1)$ entries of the vector $\text{diag } A_{N-1}$, so the remaining majorization inequalities are in force. Finally, rearranging the relation $\sum_{k=1}^N z_k = \sum_{k=1}^N a_k$ yields $\sum_{k=2}^N z_k = \text{Tr } A_{N-1}$. In consequence, the induction furnishes a rotation Q_{N-1} that sets the diagonal of A_{N-1} equal to the vector $(z_2, \dots, z_N)$. Defining

$$P_3 \stackrel{\text{def}}{=} \begin{bmatrix} 1 & \mathbf{0}^* \\ \mathbf{0} & Q_{N-1} \end{bmatrix},$$

we see that conjugating A by the orthogonal matrix $P = P_1 P_2 P_3$ transforms the diagonal entries of A to $\mathbf{z}$ while retaining the spectrum $\boldsymbol{\lambda}$. This proof leads to the following algorithm.

ALGORITHM 2 (Generalized Chan–Li). *Let A be an $N \times N$ Hermitian matrix with diagonal entries $\mathbf{a}$ and spectrum $\boldsymbol{\lambda}$. Let $\mathbf{z} \succcurlyeq \mathbf{a}$ be a vector with the same spectrum as $\mathbf{a}$. The algorithm constructs a unitary matrix U such that $U^* A U$ has diagonal entries $\mathbf{z}$.*

1. Set $i = 1$.
2. Find the least $j > i$ so that $a_{j-1} \leq z_i \leq a_j$.
3. Use a symmetric permutation to set a_{i+1} equal to a_j while shifting diagonal entries $i+1, \dots, j-1$ one place down the diagonal.
4. Construct a plane rotation Q in the $(i, i+1)$ -plane using (3.2) and (3.3) with appropriate modifications.

5. Replace A by Q^*AQ .
6. Use a symmetric permutation to re-sort the diagonal entries of A in ascending order.
7. Increment i and repeat Steps 2–7 while $i < N$.

This algorithm has the same functionality and complexity as Algorithm 1 but it is different in the plane rotations used. It is conceptually simpler to perform the permutations described in the algorithm, but an implementation can avoid them. The MATLAB code provided in the appendix of [7] demonstrates one such implementation.

3.4. One-sided algorithms. It is well known that any positive semidefinite matrix $A \in \mathbb{M}_N$ can be expressed as the product X^*X where $X \in \mathbb{M}_{d,N}$ and $d \geq \text{rank } A$. With this factorization, the two-sided transformation $A \mapsto Q^*AQ$ is equivalent to a one-sided transformation $X \mapsto XQ$. In consequence, the machinery of Algorithm 1 requires little adjustment to produce these factors. The following algorithm generalizes the one-sided version proposed by Davies and Higham in [6].

ALGORITHM 3 (One-sided generalized Bendel–Mickey). $\mathbf{z} \leftarrow \mathbf{a}$

$\mathbf{z} \leftarrow \mathbf{a}$ N $d \times N$ X $\mathbf{a}$

1. Find $i < j$ for which $\|\mathbf{x}_{i-1}\|_2^2 < z_i$, $z_j < \|\mathbf{x}_j\|_2^2$ and $\|\mathbf{x}_k\|_2^2 = z_k$ for $i < k < j$. If no such pair exists, we are either done or the majorization condition is violated.
2. Form the quantities

$$a_i = \|\mathbf{x}_i\|_2^2, \quad a_{ji} = \langle \mathbf{x}_j, \mathbf{x}_i \rangle, \quad \text{and} \quad a_j = \|\mathbf{x}_j\|_2^2.$$

3. Construct a plane rotation Q in the (i, j) -plane using (3.2) and (3.3) with appropriate modifications to transform a_i to z_i in the case $z_i - a_i \leq a_j - z_j$, or transform a_j to z_j otherwise.
4. Replace X by XQ .
5. Repeat Steps 1–4 until all column norms are as desired.

The algorithm requires about $12dN$ real floating-point operations and storage of $N(d+2)$ complex floating-point numbers including the desired column norms and the current column norms. A similar modification of our generalized Chan–Li algorithm also leads to a one-sided version.

4. Numerical examples. The generalized algorithms can produce a richer set of matrices with prescribed diagonal entries and eigenvalues, making it possible to find solutions that satisfy additional properties or better suit the application.

We illustrate the generalized Chan–Li algorithm by comparing it with the classical algorithm. Suppose we want to produce a Hermitian matrix with eigenvalues $(1, 4, 5, 7, 9)$ and diagonal entries $(2, 5, 6, 6, 7)$. This example was presented in [4]; our generalized algorithm (essentially) yields the same result

$$A_{\text{ChanLi}}^{(1)} = \begin{bmatrix} 2.0000 & 0 & 0.7071 & -0.9487 & 0.7746 \\ 0 & 5.0000 & 0 & 0 & 0 \\ 0.7071 & 0 & 6.0000 & 1.3416 & -1.0954 \\ -0.9487 & 0 & 1.3416 & 6.0000 & 2.4495 \\ 0.7746 & 0 & -1.0954 & 2.4495 & 7.0000 \end{bmatrix}.$$

Notice the sparsity structure in the above matrix. In applications such as designing matrices for testing eigenvalue solvers [9], it would be better to produce a more

random Hermitian matrix that satisfies the diagonal and eigenvalue constraints. The generalized algorithms can be used for this purpose. We first generate a sequence of (say) six vectors satisfying

$$(1, 4, 5, 7, 9) = \mathbf{z}_0 \preceq \mathbf{z}_1 \preceq \mathbf{z}_2 \preceq \mathbf{z}_3 \preceq \mathbf{z}_4 \preceq \mathbf{z}_5 = (2, 5, 6, 6, 7).$$

The vectors $\mathbf{z}_1, \mathbf{z}_2, \mathbf{z}_3, \mathbf{z}_4$ may be generated as random convex combinations of $\mathbf{z}_0$ and $\mathbf{z}_5$. This construction ensures that the majorization conditions hold. Five steps of the generalized Chan–Li algorithm are then successively used to transform the diagonal matrix $\text{diag}(\mathbf{z}_0)$ to have diagonals $\mathbf{z}_1, \mathbf{z}_2, \dots, \mathbf{z}_5$. We arrive at the final matrix

$$A_{\text{GenChanLi}}^{(1)} = \begin{bmatrix} 2.0000 & 1.0400 & 1.4517 & -0.6294 & -0.3720 \\ 1.0400 & 5.0000 & 0.3620 & -0.2157 & 1.4731 \\ 1.4517 & 0.3620 & 6.0000 & 1.6901 & -0.6544 \\ -0.6294 & -0.2157 & 1.6901 & 6.0000 & -1.2822 \\ -0.3720 & 1.4731 & -0.6544 & -1.2822 & 7.0000 \end{bmatrix}.$$

For the wireless application mentioned in the introduction, the matrices in question must have all nonzero eigenvalues equal to one; see [20] for more details. The following example calls for the generation of matrices with eigenvalues $(0, 0, 1, 1, 1)$ and diagonal $(0.4, 0.6, 0.6, 0.6, 0.8)$. The Chan–Li algorithm produces

$$A_{\text{ChanLi}}^{(2)} = \begin{bmatrix} 0.4000 & -0.4899 & 0 & 0 & 0 \\ -0.4899 & 0.6000 & 0 & 0 & 0 \\ 0 & 0 & 0.6000 & 0.4000 & -0.2828 \\ 0 & 0 & 0.4000 & 0.6000 & 0.2828 \\ 0 & 0 & -0.2828 & 0.2828 & 0.8000 \end{bmatrix}.$$

In the wireless application, it is often desirable to have lower variance in the magnitudes of the off-diagonal entries, which are also known as “cross-correlations.”

We can use the generalized algorithms to obtain “random” solutions as follows. Generate random matrices with the given spectrum and select those which have a diagonal majorized by the target diagonal. Then apply either one of the generalized algorithms. Using such a strategy, the generalized Bendel–Mickey algorithm provides the following matrix:

$$A_{\text{GenBendelMickey}}^{(2)} = \begin{bmatrix} 0.4000 & 0.1803 & 0.2101 & -0.3558 & 0.1916 \\ 0.1803 & 0.6000 & -0.3991 & -0.1487 & 0.1615 \\ 0.2101 & -0.3991 & 0.6000 & -0.1237 & 0.1457 \\ -0.3558 & -0.1487 & -0.1237 & 0.6000 & 0.2756 \\ 0.1916 & 0.1615 & 0.1457 & 0.2756 & 0.8000 \end{bmatrix}.$$

5. Conclusions. We have shown that a sequence of $(N-1)$ rotations is sufficient to replace the original diagonal of $N \times N$ Hermitian matrix with any set of diagonal entries that majorizes the original set, all the while preserving the spectrum of the matrix. The algorithms we have presented can move up a chain in the poset of Schur–Horn equivalence classes as given in Definition 2.5.

An obvious question is whether it is possible to obtain an algorithm that moves down a chain instead. In other words, is it possible to construct a finite sequence of rotations to replace the diagonal with a set of entries that majorizes the eigenvalues but not necessarily the original diagonal? Since the diagonal matrix of eigenvalues lies at the bottom of every chain, it might seem at first glance that, as a special case, we

are attempting to calculate the eigenvalues in finite time. We avoid this paradox since we assume that the target diagonal is already known. In fact, to get to the bottom of the chain, $O(N^2)$ Givens rotations can be used to reduce the initial matrix to tridiagonal form and then to the desired diagonal matrix of eigenvalues (these include the application of perfect shifts to the tridiagonal matrix; see [8]). By combining these $O(N^2)$ rotations with the Chan–Li algorithm, we answer our question in the affirmative. However this strategy goes through the bottom of the chain, so a natural question is whether there is a more direct construction. A direct algorithm may also be computationally more efficient. However, it seems harder to transform a vector into one of its predecessors than into one of its successors. Entropy may provide a reasonable explanation: it is easier to make things more uniform than less uniform.

Other interesting questions arise. What is the structure of a general Schur–Horn equivalence class of Hermitian matrices? Is there a fast procedure that is capable of constructing every member of a given equivalence class? Is it possible to define a uniform probability measure on each class and to construct members from a class uniformly at random? In this paper, we have restricted our attention to finite step algorithms. Iterative algorithms are an alternative, especially for the case of producing Hermitian matrices that satisfy additional constraints; see [21] for such an approach. It would be useful to understand these problems better, and we hope that other researchers will take interest.

REFERENCES

- [1] B. C. ARNOLD, *Majorization and the Lorenz Order: A Brief Introduction*, Lecture Notes in Statistics, 43, Springer-Verlag, Berlin, 1987.
- [2] R. B. BENDEL AND M. R. MICKEY, *Population correlation matrices for sampling experiments*, Commun. Statist. Simul. Comp., B7 (1978), pp. 163–182.
- [3] R. BHATIA, *Matrix Analysis*, Springer-Verlag, New York, 1996.
- [4] N. N. CHAN AND K.-H. LI, *Diagonal elements and eigenvalues of a real symmetric matrix*, J. Math. Anal. Appl., 91 (1983), pp. 562–566.
- [5] M. T. CHU, *Constructing a Hermitian matrix from its diagonal entries and eigenvalues*, SIAM J. Matrix Anal. Appl., 16 (1995), pp. 207–217.
- [6] P. I. DAVIES AND N. J. HIGHAM, *Numerically stable generation of correlation matrices and their factors*, BIT, 40 (2000), pp. 640–651.
- [7] I. S. DHILLON, R. W. HEATH JR., M. A. SUSTIK, AND J. A. TROPP, *Generalized finite algorithms for constructing Hermitian matrices with prescribed diagonal and spectrum*, Technical Report TR-03-49, Department of Computer Sciences, The University of Texas at Austin, Austin, TX, 2003.
- [8] I. S. DHILLON AND A. N. MALYSHEV, *Inner deflation for symmetric tridiagonal matrices*, Linear Algebra Appl., 358 (2003), pp. 139–144.
- [9] Z. DRMAČ, *Accurate computation of the product-induced singular value decomposition with applications*, SIAM J. Numer. Anal., 35 (1998), pp. 1969–1994.
- [10] G. H. GOLUB AND C. F. VAN LOAN, *Matrix Computations*, 3rd ed., Johns Hopkins University Press, Baltimore, MD, 1996.
- [11] R. B. HOLMES, *On random correlation matrices*, SIAM J. Matrix Anal. Ap., 12 (1991), pp. 239–272.
- [12] A. HORN, *Doubly stochastic matrices and the diagonal of a rotation matrix*, Amer. J. Math., 76 (1954), pp. 620–630.
- [13] R. A. HORN AND C. R. JOHNSON, *Matrix Analysis*, Cambridge University Press, Cambridge, UK, 1985.
- [14] K. D. IKRAMOV, *A remark on “A note on constructing a symmetric matrix with specified diagonal entries and eigenvalues,”* BIT, 38 (1998), p. 807.
- [15] J. L. KELLEY, *General Topology*, Van Nostrand, New York, 1955.
- [16] R. S. LEITE, R. W. RICHA, AND C. TOMEI, *Geometric proofs of some theorems of Schur–Horn type*, Linear Algebra Appl., 286 (1999), pp. 149–173.
- [17] G. MARSAGLIA AND I. OLKIN, *Generating correlation matrices*, SIAM J. Sci. Statist. Comput.,

- 5 (1984), pp. 470–475.
- [18] A. W. MARSHALL AND I. OLKIN, *Inequalities: Theory of Majorization and its Applications*, Academic Press, New York, 1979.
 - [19] G. W. STEWART, *The efficient generation of random orthogonal matrices with an application to condition estimation*, SIAM J. Numer. Anal., 17 (1980), pp. 403–409.
 - [20] J. A. TROPP, I. S. DHILLON, AND R. W. HEATH JR., *Finite-step algorithms for constructing optimal CMDA signature sequences*, IEEE Trans. Inform. Theory, 50 (2004), pp. 2916–2921.
 - [21] J. A. TROPP, I. S. DHILLON, R. W. HEATH JR., AND T. STROHMER, *Designing structured tight frames via an alternating projection method*, IEEE Trans. Inform. Theory, 51 (2004), pp. 188–209.
 - [22] P. VISWANATH AND V. ANANTHARAM, *Optimal sequences and sum capacity of synchronous CDMA systems*, IEEE Trans. Inform. Theory, 45 (1999), pp. 1984–1991.
 - [23] H. ZHA AND Z. ZHANG, *A note on constructing a symmetric matrix with specified diagonal entries and eigenvalues*, BIT, 35 (1995), pp. 448–451.

A PERTURBATION BOUND OF THE DRAZIN INVERSE OF A MATRIX BY SEPARATION OF SIMPLE INVARIANT SUBSPACES*

YIMIN WEI[†], XIEZHANG LI[‡], AND FANBIN BU[§]

Abstract. A constructive perturbation bound of the Drazin inverse of a square matrix is derived using a technique proposed by G. Stewart and based on perturbation theory for invariant subspaces. This is an improvement of the result published by the authors Wei and Li [*Numer. Linear Algebra Appl.*, 10 (2003), pp. 563–575]. It is a totally new approach to developing perturbation bounds for the Drazin inverse of a matrix. A numerical example which indicates the sharpness of the perturbation bound is presented.

Key words. Drazin inverse, perturbations, separation function, simple invariant subspaces

AMS subject classifications. 15A09, 65F20

DOI. 10.1137/S0895479804439948

1. Introduction. The Drazin inverse occurs in a number of applications, such as singular differential and difference equations [2], Markov chains [10], and iterative methods and numerical analysis [6, 11, 21, 22].

For each matrix $A \in \mathbb{C}^{n \times n}$, the index of A , written as $\text{Ind}(A)$, is the smallest nonnegative integer k for which $\text{rank}(A^k) = \text{rank}(A^{k+1})$. If A is singular, $\text{Ind}(A)$ is the largest order of Jordan blocks of A corresponding to the zero eigenvalues of A . The k -th power of a square matrix A , denoted by A^D , is the matrix satisfying

$$(1.1) \quad A^{k+1}X = A^k, \quad XAX = X, \quad AX = XA,$$

where $k = \text{Ind}(A)$. For any square matrix A , there exists a unique Drazin inverse of A (cf. [2]). In particular, if $\text{Ind}(A) \leq 1$, the Drazin inverse is called the group inverse of A , denoted by $A^\#$. From now on, let $k = \text{Ind}(A)$ for convenience.

A sequence of matrices $\{A_j\}$ in $\mathbb{C}^{n \times n}$ converges to a matrix A , denoted by $A_j \rightarrow A$ as $j \rightarrow \infty$, if $\|A_j - A\| \rightarrow 0$ as $j \rightarrow \infty$, where $\|\cdot\|$ is a matrix norm. Two matrix norms will be used in this paper: the Frobenius norm of A defined by

$$\|A\|_F = \sqrt{\sum_{i,j=1}^n |a_{ij}|^2},$$

where $A = [a_{ij}]$, and the spectral norm of A defined by

$$\|A\|_2 = \max_{\|x\|_2=1} \|Ax\|_2,$$

*Received by the editors January 15, 2004; accepted for publication (in revised form) by B. T. Kågström November 3, 2004; published electronically June 22, 2005.

<http://www.siam.org/journals/simax/27-1/43994.html>

[†]Department of Mathematics, Fudan University, Shanghai, 200433, People's Republic of China (ymwei@fudan.edu.cn). The research of this author was supported by the National Natural Science Foundation of China under grant 10471027 and by the Shanghai Education Committee.

[‡]Department of Mathematical Sciences, Georgia Southern University, Statesboro, GA 30460 (xli@georgiasouthern.edu). Part of this work was completed while the author visited Fudan University, China.

[§]Institute of Mathematics, Fudan University, Shanghai, 200433, People's Republic of China (fanbin@gmail.com).

where $\|x\|_2 = \sqrt{\sum_{i=1}^n |x_i|^2}$. A necessary and sufficient condition for the continuity of the Drazin inverse was established by Campbell and Meyer [1] in 1975 as follows. If A_j converges to A , then A_j^D converges to A^D if and only if

$$(1.2) \quad \text{rank}(A_j^{k_j}) = \text{rank}(A^k),$$

where $k_j = \text{Ind}(A_j)$ for all sufficiently large j . It is mentioned in [1] that this continuity of the Drazin inverse A^D of the matrix A is not the standard continuity but a continuity relative to special perturbations E in A . The same is valid for the convergence of the Drazin inverse of a sequence of matrices. In the same paper, the authors indicated two difficulties in establishing the perturbation bounds for the Drazin inverse.

Rong [13] in 1982 gave an upper bound for the Drazin inverse which is based on the Moore–Penrose inverse. But the assumptions are not easy to verify. In 1997, Wei and Wang [17] gave a simple expression for the Drazin inverse with the two-sided condition. Later, Wei [19] relaxed it to a one-sided condition. Several authors [3, 4, 5, 6, 8, 9, 12, 18] studied the special perturbation of the Drazin inverse under some restrictions.

Wei and Wu [20] derived a general upper bound for the Drazin inverse in 2000. Koliha [7] also obtained upper bounds for the Drazin inverse in terms of distance between the matrices and their eigenprojections. Wei and Li [23] recently improved the perturbation bounds for the Drazin inverse up to the first order of the norm of a perturbation matrix E . However, the upper bound may be enlarged by using the matrix $G = (A + E)^l - A^l$, where $l = \max\{\text{Ind}(A + E), \text{Ind}(A)\}$.

In this paper, we derive a constructive perturbation bound for the Drazin inverse A^D of a square matrix A using a technique proposed by G. Stewart and based on perturbation theory for invariant subspaces of a matrix. The matrix G is not used, which is an improvement of the result of [23]. Instead, a totally new approach is provided to develop perturbation bounds for the Drazin inverse of a matrix. This paper is organized as follows. Preliminaries are studied in section 2. An explicit formula for $(A + E)^D$ is presented in section 3 through the new approach. This is a key step of the paper. A new perturbation bound for A^D is derived in section 4. Finally in section 5, a numerical example is given to indicate the sharpness of the perturbation bound.

2. Preliminaries. Let $A \in \mathbb{C}^{n \times n}$. It follows from the Schur decomposition theorem [15] that there is a unitary matrix $\begin{bmatrix} X_1 & Y_2 \end{bmatrix}$ such that

$$\begin{bmatrix} X_1 & Y_2 \end{bmatrix}^H A \begin{bmatrix} X_1 & Y_2 \end{bmatrix} = \begin{bmatrix} C & H \\ 0 & N \end{bmatrix},$$

where $C \in \mathbb{C}^{r \times r}$ is nonsingular and $N \in \mathbb{C}^{(n-r) \times (n-r)}$ is a nilpotent matrix of index k . It follows [15, Theorem 5.1.5] that there are matrices X_2 and Y_1 such that

$$\begin{bmatrix} X_1 & X_2 \end{bmatrix}^{-1} = \begin{bmatrix} Y_1 & Y_2 \end{bmatrix}^H$$

and

$$(2.1) \quad \begin{bmatrix} Y_1^H \\ Y_2^H \end{bmatrix} A \begin{bmatrix} X_1 & X_2 \end{bmatrix} = \begin{bmatrix} C & 0 \\ 0 & N \end{bmatrix}.$$

It is easy to verify by (1.1) that the Drazin inverse A^D of A is given by [2]

$$(2.2) \quad A^D = \begin{bmatrix} X_1 & X_2 \end{bmatrix} \begin{bmatrix} C^{-1} & 0 \\ 0 & 0 \end{bmatrix} \begin{bmatrix} Y_1^H \\ Y_2^H \end{bmatrix}.$$

Let E be a perturbation matrix of A such that $\text{rank}(A^k) = \text{rank}((A + E)^j)$, where $j = \text{Ind}(A + E)$. This assumption is essential for the continuity of the Drazin inverse. Suppose

$$(2.3) \quad \begin{bmatrix} Y_1^H \\ Y_2^H \end{bmatrix} E \begin{bmatrix} X_1 & X_2 \end{bmatrix} = \begin{bmatrix} F_{11} & F_{12} \\ F_{21} & F_{22} \end{bmatrix};$$

then

$$\begin{bmatrix} Y_1^H \\ Y_2^H \end{bmatrix} (A + E) \begin{bmatrix} X_1 & X_2 \end{bmatrix} = \begin{bmatrix} C + F_{11} & F_{12} \\ F_{21} & N + F_{22} \end{bmatrix}.$$

It follows that

$$(2.4) \quad \begin{aligned} X_1 Y_1^H &= A A^D, & X_2 Y_2^H &= I - A A^D, \\ C^{-1} &= Y_1^H A^D X_1, \\ F_{i,j} &= Y_i^H E X_j, \quad i, j = 1, 2. \end{aligned}$$

Recall that $\begin{bmatrix} X_1 & X_2 \end{bmatrix}$ is a unitary matrix. We have

$$\|X_1 W_1\|_F = \|W_1\|_F \quad \text{and} \quad \|W_2 Y_2^H\|_F = \|W_2\|_F,$$

where $X_1 W_1$ and $W_2 Y_2^H$ are well defined. It is known that if AB is well defined, then

$$\|AB\|_F \leq \min\{\|A\|_F \|B\|_2, \|A\|_2 \|B\|_F\}.$$

The following relations are obvious:

$$(2.5) \quad \begin{aligned} \|A A^D\|_2 &= \|I - A A^D\|_2 \text{ if } A \text{ is a nonzero singular matrix,} \\ \|X_1\|_2 &= \|Y_2\|_2 = 1, \\ \|C^{-1}\|_F &= \|X_1 Y_1^H A^D X_1\|_F \leq \|A^D\|_F, \\ \|F_{11}\|_F &\leq \|A A^D E\|_F \leq \|E\|_F \|A A^D\|_2, \\ \|F_{12}\|_F &\leq \|A A^D E(I - A A^D)\|_F \leq \|E\|_F \|A A^D\|_2^2, \\ \|F_{21}\|_F &\leq \|E\|_F, \\ \|F_{22}\|_F &\leq \|E(I - A A^D)\|_F \leq \|E\|_F \|A A^D\|_2. \end{aligned}$$

Now we introduce simple invariant subspaces of a matrix. It follows from (2.1) that

$$A X_1 = X_1 C \quad \text{and} \quad A X_2 = X_2 N.$$

Let $\sigma(M)$ denote the set of all eigenvalues of a square matrix M . Since $\sigma(C) \cap \sigma(N) = \emptyset$, the ranges of X_1 and X_2 , denoted, respectively, by $\mathcal{R}(X_1)$ and $\mathcal{R}(X_2)$, are called *simple invariant subspaces* of A [15]. With an analogous argument, $A + E$ can be expressed as

$$\begin{bmatrix} \tilde{Y}_1^H \\ \tilde{Y}_2^H \end{bmatrix} (A + E) \begin{bmatrix} \tilde{X}_1 & \tilde{X}_2 \end{bmatrix} = \begin{bmatrix} \tilde{C} & 0 \\ 0 & \tilde{N} \end{bmatrix},$$

where $\tilde{C}$ is nonsingular, $\tilde{N}$ is a nilpotent matrix, and

$$\begin{bmatrix} \tilde{X}_1 & \tilde{X}_2 \end{bmatrix}^{-1} = \begin{bmatrix} \tilde{Y}_1^H \\ \tilde{Y}_2^H \end{bmatrix}.$$

The condition for $\lim_{\|E\|_F \rightarrow 0} (A + E)^D = A^D$ is equivalent to $\text{rank}(C) = \text{rank}(\tilde{C})$. Notice that $\mathcal{R}(X_1)$ and $\mathcal{R}(\tilde{X}_1)$ are the simple invariant subspaces of A and $A + E$ corresponding to nonzero eigenvalues of A and $A + E$, respectively. This implies that a necessary and sufficient condition for the continuity of the Drazin inverse of a matrix is that the dimension of the simple invariant subspace corresponding to all nonzero eigenvalues of $A + E$ equals that of A as $\|E\|_F \rightarrow 0$, i.e., $\dim(\mathcal{R}(X_1)) = \dim(\mathcal{R}(\tilde{X}_1))$.

The separation function of two square matrices A and B is defined as [14]

$$\text{sep}_F(A, B) = \begin{cases} \min_{\|W\|_F=1} \|AW - WB\|_F & \text{if } \sigma(A) \cap \sigma(B) = \emptyset; \\ 0 & \text{otherwise.} \end{cases}$$

In fact, $\text{sep}_F(A, B)$ is the minimum singular value of the matrix $I \otimes A - B^T \otimes I$, where $\otimes$ is the Kronecker product, and we know that $\text{sep}_F(A, B) \leq \min\{|\lambda - \mu| : \lambda \in \sigma(A) \text{ and } \mu \in \sigma(B)\}$. We, nevertheless, should also keep in mind that $\text{sep}_F(A, B)$ can be very small even if the eigenvalues of A and B are well separated.

The stability of the function sep

$$\begin{aligned} \text{sep}_F(C, N) - \|C_1\|_F - \|N_1\|_F &\leq \text{sep}_F(C + C_1, N + N_1) \\ &\leq \text{sep}_F(C, N) + \|C_1\|_F + \|N_1\|_F \end{aligned}$$

is important in establishing the following result of perturbation bounds of simple invariant subspaces.

PROPOSITION 2.1 (see [15]). Let $\begin{bmatrix} X_1 & Y_2 \end{bmatrix} \in \mathbb{C}^{n \times n}$ and $\begin{bmatrix} X_1 & X_2 \end{bmatrix}^{-1} = \begin{bmatrix} Y_1 & Y_2 \end{bmatrix}^H$

where $A = \begin{bmatrix} L_1 & 0 \\ 0 & L_2 \end{bmatrix}$ and $E = \begin{bmatrix} F_{11} & F_{12} \\ F_{21} & F_{22} \end{bmatrix}$ are

$$\begin{bmatrix} Y_1 & Y_2 \end{bmatrix}^H A \begin{bmatrix} X_1 & X_2 \end{bmatrix} = \begin{bmatrix} L_1 & 0 \\ 0 & L_2 \end{bmatrix},$$

and $\sigma(L_1) \cap \sigma(L_2) = \emptyset$.

$$\begin{bmatrix} Y_1 & Y_2 \end{bmatrix}^H E \begin{bmatrix} X_1 & X_2 \end{bmatrix} = \begin{bmatrix} F_{11} & F_{12} \\ F_{21} & F_{22} \end{bmatrix}.$$

Then

$$\text{sep}_F(L_1, L_2) - \|F_{11}\|_F - \|F_{22}\|_F > 2\sqrt{\|F_{12}\|_F \|F_{21}\|_F},$$

then the perturbation P is bounded by

$$\|P\|_F < \frac{2\|F_{21}\|_F}{\text{sep}_F(L_1, L_2) - \|F_{11}\|_F - \|F_{22}\|_F},$$

where $\tilde{X}_1 = X_1 + X_2 P$

$$\tilde{X}_1 = X_1 + X_2 P$$

and

$$\tilde{Y}_2 = Y_2 - Y_1 P^H$$

such that $\tilde{A} = A + E$.

3. An expression for $(A + E)^D$. In this section, we derive an expression for $(A + E)^D$ by applying perturbation bounds of simple invariant subspaces stated in Proposition 2.1 to the matrices A and E in (2.1) and (2.3).

THEOREM 3.1. *Let A and E be $n \times n$ matrices satisfying (2.1) and (2.3). Then, $\text{rank}(A^k) = \text{rank}((A + E)^j)$, where j is the smallest integer such that $\|E\|_F \|A\|_2^j < \text{sep}_F(A, 0)$. Moreover, $(A + E)^D$ is given by*

$$(3.1) \quad (A + E)^D = \begin{bmatrix} X_1 & X_2 \end{bmatrix} \begin{bmatrix} (C + F_{11} + F_{12}P)^{-1}(I + QP) & -(C + F_{11} + F_{12}P)^{-1}Q \\ P(C + F_{11} + F_{12}P)^{-1}(I + QP) & -P(C + F_{11} + F_{12}P)^{-1}Q \end{bmatrix} \begin{bmatrix} Y_1^H \\ Y_2^H \end{bmatrix},$$

$$P = (C + F_{11} + F_{12}P)^{-1}(I + QP), \quad Q = (C + F_{11} + F_{12}P)^{-1}Q.$$

$$(3.2) \quad \|P\|_F < \frac{2\|E\|_F}{\text{sep}_F(C, N) - 2\|E\|_F \|AA^D\|_2}$$

$$(3.3) \quad \|Q\|_F < \frac{2\|E\|_F \|AA^D\|_2^2}{\text{sep}_F(C, N) - 2\|E\|_F \|AA^D\|_2 - 2\|E\|_F \|AA^D\|_2^2 \|P\|_F}.$$

Assume that $\|E\|_F$ is sufficiently small such that

$$(3.4) \quad \text{sep}_F(C, N) - \|F_{11}\|_F - \|F_{22}\|_F > 2\sqrt{\|F_{12}\|_F \|F_{21}\|_F}.$$

Then, according to Proposition 2.1, there exists a unique matrix P satisfying

$$\|P\|_F < \frac{2\|F_{21}\|_F}{\text{sep}_F(C, N) - \|F_{11}\|_F - \|F_{22}\|_F} \leq \frac{2\|E\|_F}{\text{sep}_F(C, N) - 2\|E\|_F \|AA^D\|_2}$$

such that

$$\begin{bmatrix} X_1 + X_2P & X_2 \end{bmatrix}^{-1} = \begin{bmatrix} Y_1 & Y_2 - Y_1P^H \end{bmatrix}^H$$

and

$$\begin{bmatrix} Y_1^H \\ Y_2^H - PY_1^H \end{bmatrix} (A + E) \begin{bmatrix} X_1 + X_2P & X_2 \end{bmatrix} = \begin{bmatrix} C + F_{11} + F_{12}P & F_{12} \\ 0 & N + F_{22} - PF_{12} \end{bmatrix}.$$

Thus,

$$(3.5) \quad \begin{bmatrix} I & 0 \\ -P & I \end{bmatrix} \begin{bmatrix} Y_1^H \\ Y_2^H \end{bmatrix} (A + E) \begin{bmatrix} X_1 & X_2 \end{bmatrix} \begin{bmatrix} I & 0 \\ P & I \end{bmatrix} = \begin{bmatrix} C + F_{11} + F_{12}P & F_{12} \\ 0 & N + F_{22} - PF_{12} \end{bmatrix}.$$

It is obvious that for a sufficiently small $\|E\|_F$,

$$(3.6) \quad \|P\|_F < 1$$

and

$$(3.7) \quad C + F_{11} + F_{12}P \text{ is nonsingular,}$$

which will also be shown in the later discussion. Moreover, by the assumption of $\text{rank}(A^k) = \text{rank}((A + E)^j)$, $N + F_{22} - PF_{12}$ is nilpotent, which implies that

$$\sigma(C + F_{11} + F_{12}P) \cap \sigma(N + F_{22} - PF_{12}) = \emptyset.$$

Furthermore, if $\|E\|_F$ is sufficiently small, then

$$(3.8) \quad \text{sep}_F(C, N) - \|F_{11} + F_{12}P\|_F - \|F_{22} - PF_{12}\|_F > 0.$$

It is noted that [16] $\text{sep}_F(C, N) = \text{sep}_F(N, C)$. Thus, in an analogous way, by Proposition 2.1 and (2.5), there exists a unique matrix Q satisfying

$$\begin{aligned} \|Q\|_F &< \frac{2\|F_{12}\|_F}{\text{sep}_F(C, N) - \|F_{11} + F_{12}P\|_F - \|F_{22} - PF_{12}\|_F} \\ &\leq \frac{2\|E\|_F \|AA^D\|_2^2}{\text{sep}_F(C, N) - 2\|E\|_F \|AA^D\|_2 - 2\|E\|_F \|AA^D\|_2^2 \|P\|_F} \end{aligned}$$

such that

$$(3.9) \quad \begin{aligned} &\begin{bmatrix} I & -Q \\ 0 & I \end{bmatrix} \begin{bmatrix} I & 0 \\ -P & I \end{bmatrix} \begin{bmatrix} Y_1^H \\ Y_2^H \end{bmatrix} (A + E) \begin{bmatrix} X_1 & X_2 \end{bmatrix} \begin{bmatrix} I & 0 \\ P & I \end{bmatrix} \begin{bmatrix} I & Q \\ 0 & I \end{bmatrix} \\ &= \begin{bmatrix} C + F_{11} + F_{12}P & 0 \\ 0 & N + F_{22} - PF_{12} \end{bmatrix}. \end{aligned}$$

Relation (3.1) follows from (3.9). This completes the proof. $\square$

We next show how small $\|E\|_F$ needs to be such that (3.4), (3.6)–(3.8) are fulfilled.

THEOREM 3.2. *Let $A \in \mathbb{C}^{n \times n}$ be a matrix with $\text{rank}(A) = r$.*

(i) *If*

$$(3.10) \quad \|E\|_F < \frac{\text{sep}_F(C, N)}{4\|AA^D\|_2},$$

then

$$\text{sep}_F(C, N) - \|F_{11}\|_F - \|F_{22}\|_F > 2\sqrt{\|F_{12}\|_F \|F_{21}\|_F}$$

$$(ii) \quad \|P\|_F < \frac{1}{\|AA^D\|_2} (\leq 1).$$

then

$$(3.11) \quad \|E\|_F < \frac{1}{2\|AA^D\|_2},$$

then (3.4)–(3.8) hold.

(iii) *If* (3.10) *and*

$$\text{sep}_F(C, N) - \|F_{11} + F_{12}P\|_F - \|F_{22} - PF_{12}\|_F > 0.$$

then the relations (2.5) will be used in the following proofs.

(i) It follows from (2.5) that

$$\|F_{11}\|_F + \|F_{22}\|_F + 2\sqrt{\|F_{12}\|_F \|F_{21}\|_F} \leq 4\|E\|_F \|AA^D\|_2 < \text{sep}_F(C, N).$$

The last inequality follows from (3.10).

Moreover, by (3.2) and (3.10) we have

$$\|P\|_F < \frac{2\|E\|_F}{\text{sep}_F(C, N) - 2\|E\|_F\|AA^D\|_2} < \frac{1}{\|AA^D\|_2}.$$

(ii) Assume that

$$\|E\|_F < \frac{1}{2\|A^D\|_2}.$$

Then

$$\begin{aligned} \|C^{-1}(F_{11} + F_{12}P)\|_F &\leq \|Y_1^H A^D X_1 Y_1^H E X_1\|_F + \|Y_1^H A^D X_1 Y_1^H E X_2 P\|_F \\ &\leq \|A^D E\|_F + \|A^D E\|_F \|AA^D\|_2 \|P\|_F \\ (3.12) \quad &\leq 2\|E\|_F \|A^D\|_2 < 1. \end{aligned}$$

Thus $I + C^{-1}(F_{11} + F_{12}P)$ is nonsingular and so is $C + F_{11} + F_{12}P$.

(iii) Suppose that (3.10) holds. It follows from (i) that $\|P\|_F < 1/\|AA^D\|_2$. Then we have

$$\begin{aligned} &\|F_{11} + F_{12}P\|_F + \|F_{22} - PF_{12}\|_F \\ &\leq \|E\|_F \|AA^D\|_2 + 2\|E\|_F \|AA^D\|_2^2 \|P\|_F + \|E\|_F \|AA^D\|_2 \\ &\leq 4\|E\|_F \|AA^D\|_2 < \text{sep}_F(C, N). \end{aligned}$$

This completes the proof. $\square$

4. A perturbation bound. Next, we derive a perturbation bound of A^D with the explicit expression (3.1) for $(A + E)^D$. It is the main result of this paper.

THEOREM 4.1. *Let $A \in \mathbb{C}^{n \times n}$ be a matrix with $\text{rank}(A) = r$ and $E \in \mathbb{C}^{n \times n}$ be a perturbation matrix. If (3.1) holds, then*

$$\begin{aligned} (3.10) \quad &\frac{\|(A + E)^D - A^D\|_F}{\|A^D\|_F} \leq g\{\|E\|_F \|A^D\|_2 + p\|E\|_F \|A^D\|_2 \|AA^D\|_2 \\ (4.1) \quad &+ q + 2pq\|AA^D\|_2 + p\|AA^D\|_2(1 + pq\|AA^D\|_2)\}, \end{aligned}$$

$$\begin{aligned} p &= \frac{2\|E\|_F}{\text{sep}_F(C, N) - 2\|E\|_F\|AA^D\|_2}, \\ q &= \frac{2\|E\|_F\|AA^D\|_2^2}{\text{sep}_F(C, N) - 2\|E\|_F\|AA^D\|_2 - 2p\|E\|_F\|AA^D\|_2^2}, \\ (4.2) \quad g &= \frac{1}{1 - \|E\|_F\|A^D\|_2 - p\|E\|_F\|A^D\|_2\|AA^D\|_2}. \end{aligned}$$

With (2.2) and (3.1) we denote

$$(A + E)^D - A^D \equiv \begin{bmatrix} X_1 & X_2 \end{bmatrix} \begin{bmatrix} T_{11} & T_{12} \\ T_{21} & T_{22} \end{bmatrix} \begin{bmatrix} Y_1^H \\ Y_2^H \end{bmatrix},$$

where

$$\begin{aligned} T_{11} &= (C + F_{11} + F_{12}P)^{-1}(I + QP) - C^{-1}, \\ T_{12} &= -(C + F_{11} + F_{12}P)^{-1}Q, \\ T_{21} &= P(C + F_{11} + F_{12}P)^{-1}(I + QP), \\ T_{22} &= -P(C + F_{11} + F_{12}P)^{-1}Q. \end{aligned}$$

Then

$$(4.3) \quad (A + E)^D - A^D = X_1 T_{11} Y_1^H + X_1 T_{12} Y_2^H + X_2 T_{21} Y_1^H + X_2 T_{22} Y_2^H.$$

The first term $X_1 T_{11} Y_1^H$ on the right-hand side of (4.3) can be spilt into two terms,

$$(4.4) \quad X_1 T_{11} Y_1^H = X_1 (C + F_{11} + F_{12} P)^{-1} Q P Y_1^H + X_1 ((C + F_{11} + F_{12} P)^{-1} - C^{-1}) Y_1^H.$$

First we estimate $\|(C + F_{11} + F_{12} P)^{-1}\|_F$. By perturbation theory of a nonsingular matrix, we have

$$\begin{aligned} \|(C + F_{11} + F_{12} P)^{-1}\|_F &\leq \|(I + C^{-1}(F_{11} + F_{12} P))^{-1}\|_2 \|C^{-1}\|_F \\ &\leq \frac{\|C^{-1}\|_F}{1 - \|C^{-1}(F_{11} + F_{12} P)\|_2} \\ &\leq \frac{\|A^D\|_F}{1 - \|E\|_F(\|A^D\|_2 + \|P\|_F \|A^D\|_2 \|AA^D\|_2)}, \end{aligned}$$

since (3.12) holds in the proof of (ii) in Theorem 3.1. With the notation p and g defined in (4.2), and the inequality in (3.2), we know that $\|P\|_F < p$ and

$$(4.5) \quad \|(C + F_{11} + F_{12} P)^{-1}\|_F \leq \frac{\|A^D\|_F}{1 - \|E\|_F(\|A^D\|_2 + p\|A^D\|_2 \|AA^D\|_2)} = g\|A^D\|_F.$$

Moreover, we have

$$\|(I + C^{-1}(F_{11} + F_{12} P))^{-1} - I\|_2 \leq \frac{\|C^{-1}(F_{11} + F_{12} P)\|_2}{1 - \|C^{-1}(F_{11} + F_{12} P)\|_2}.$$

Now $\|X_1((C + F_{11} + F_{12} P)^{-1} - C^{-1})Y_1^H\|_F$ can be estimated as follows:

$$\begin{aligned} &\|X_1((C + F_{11} + F_{12} P)^{-1} - C^{-1})Y_1^H\|_F \\ &= \|[I + C^{-1}(F_{11} + F_{12} P))^{-1} - I]C^{-1}Y_1^H\|_F \\ &\leq \frac{\|A^D\|_F \|C^{-1}(F_{11} + F_{12} P)\|_2}{1 - \|C^{-1}(F_{11} + F_{12} P)\|_2} \\ (4.6) \quad &\leq \frac{\|E\|_F \|A^D\|_F (\|A^D\|_2 + p\|A^D\|_2 \|AA^D\|_2)}{1 - \|E\|_F \|A^D\|_2 - p\|E\|_F \|A^D\|_2 \|AA^D\|_2}. \end{aligned}$$

Similarly, we know that $\|Q\|_F < q$ from (3.3) and (4.2). Then it follows from (4.4)–(4.6) that

$$(4.7) \quad \begin{aligned} &\|X_1 T_{11} Y_1^H\|_F \\ &\leq g\|A^D\|_F (\|E\|_F \|A^D\|_2 + p\|E\|_F \|A^D\|_2 \|AA^D\|_2 + pq\|AA^D\|_2). \end{aligned}$$

In an analogous way, we have

$$(4.8) \quad \begin{aligned} &\|X_1 T_{12} Y_2^H\|_F \leq gq\|A^D\|_F, \\ &\|X_2 T_{21} Y_1^H\|_F \leq gp\|A^D\|_F \|AA^D\|_2 (1 + pq\|AA^D\|_2), \\ &\|X_2 T_{22} Y_2^H\|_F \leq gpq\|A^D\|_F \|AA^D\|_2. \end{aligned}$$

Finally, (4.1) follows by combining (4.3), (4.7), and (4.8). $\square$

We remark that if $\|E\|_F$ is much smaller than $\frac{\text{sep}_F(C, N)}{2\|AA^D\|_2}$, then up to the first order of $\|E\|_F$, the upper bound (4.1) is given by

$$\frac{\|(A + E)^D - A^D\|_F}{\|A^D\|_F} \leq \|E\|_F \|A^D\|_2 + \frac{4\|E\|_F \|AA^D\|_2^2}{\text{sep}_F(C, N)} + O(\|E\|_F^2).$$

Since the quantity $1/\text{sep}_F(C, N)$ is the condition number of the invariant subspace $\mathcal{R}(X_1)$ of A , the function sep is sure to appear in the perturbation theory for invariant subspaces as well as for Drazin inverses.

5. Example. An example is given to illustrate that the upper bound of (4.1) is better than previous bounds. Let

$$A = \begin{bmatrix} 150 & 0 & 0 & 0 & 0 \\ 140 & a & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 & 0 \end{bmatrix}, \quad E = \epsilon \begin{bmatrix} 1 & 0 & 1 & 0 & 0 \\ 0 & 1 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 & 0 \\ 1 & 0 & 0 & 0 & 0 \end{bmatrix},$$

where $a > 0$ and $\epsilon = 10^{-9}$. It is known that $\text{Ind}(A) = 3$, $\text{Ind}(A + E) = 3$, $\text{rank}(A^3) = \text{rank}((A + E)^3) = 2$. The exact error $\frac{\|B^D - A^D\|_F}{\|A^D\|_F}$ and its various upper bounds are listed in Table 5.1. We can see from this table that our upper bound in this paper is sharper than previous upper bounds, but it is still not sharp when $\text{sep}_F(C, N)$ is small. This indicates that $1/\text{sep}_F(C, N)$ is a condition number of the Drazin inverse of a matrix.

TABLE 5.1
Exact error and upper bounds.

	$a = 40$	$a = 15$	$a = 1.3$	$a = 0.05$
$\text{sep}_F(C, N)$	28	10	0.5	9×10^{-5}
by Thm. 2 in [20]	5.11×10^{-7}	1.26×10^{-4}	2.48×10^2	3.19×10^8
by (3.1) in [13]	4.78×10^{-8}	1.58×10^{-6}	2.23×10^{-2}	1.00×10^4
by (6) in [23]	1.39×10^{-8}	2.19×10^{-7}	3.07×10^{-4}	invalid
by (17) in [23]	1.53×10^{-9}	2.36×10^{-8}	3.34×10^{-5}	5.80×10^{-1}
by (4.1)	2.76×10^{-10}	7.49×10^{-10}	1.30×10^{-8}	1.16×10^{-4}
Exact error	3.91×10^{-11}	9.71×10^{-11}	9.11×10^{-10}	2.00×10^{-8}

Acknowledgments. The authors would like to thank Prof. B. Kågström and two anonymous referees for their valuable suggestions and detailed comments.

REFERENCES

- [1] S. L. CAMPBELL AND C. D. MEYER, *Continuity properties of the Drazin inverse*, Linear Algebra Appl., 10 (1975), pp. 77–83.
- [2] S. L. CAMPBELL AND C. D. MEYER, *Generalized Inverse of Linear Transformations*, Pitman, London, 1979.
- [3] N. C. GONZÁLEZ, J. J. KOLIHA, AND Y. WEI, *Perturbation of the Drazin inverse for matrices with equal eigen-projections at zero*, Linear Algebra Appl., 312 (2000), pp. 181–189.
- [4] N. C. GONZÁLEZ, J. J. KOLIHA, AND Y. WEI, *Error bounds for perturbation of the Drazin inverse of closed operator with equal spectral projections*, Appl. Anal., 81 (2002), pp. 915–928.

- [5] D. S. DJORDJEVIĆ AND Y. WEI, *Additive results for the generalized Drazin inverse*, J. Austral. Math. Soc., 73 (2002), pp. 115–125.
- [6] R. E. HARTWIG, G. WANG, AND Y. WEI, *Some additive results on Drazin inverse*, Linear Algebra Appl., 322 (2001), pp. 207–217.
- [7] J. J. KOLIHA, *Error bounds for a general perturbation of the Drazin inverse*, Appl. Math. Comput., 126 (2002), pp. 181–185.
- [8] X. LI AND Y. WEI, *An improvement on the perturbation of the group inverse and oblique projection*, Linear Algebra Appl., 338 (2001), pp. 53–66.
- [9] X. LI AND Y. WEI, *A note on the perturbation bound of the Drazin inverse*, Appl. Math. Comput., 140 (2003), pp. 329–340.
- [10] C. D. MEYER, *The condition of a finite Markov chain and perturbation bounds for the limiting probabilities*, SIAM J. Algebraic Discrete Math., 1 (1980), pp. 273–283.
- [11] M. Z. NASHED AND Y. ZHAO, *The Drazin inverse for singular evolution equations and partial differential operators*, World Sci. Ser. Appl. Anal., 1 (1992), pp. 441–456.
- [12] V. RAKOČEVIĆ AND Y. WEI, *The perturbation theory for the Drazin inverse and its applications. II*, J. Austral. Math. Soc., 70 (2001), pp. 189–197.
- [13] G. RONG, *The error bounds for the perturbation of the Drazin inverse*, Linear Algebra Appl., 47 (1982), pp. 159–168.
- [14] G. W. STEWART, *Error bounds for approximate invariant subspaces of closed linear operators*, SIAM J. Numer. Anal., 8 (1971), pp. 796–808.
- [15] G. W. STEWART AND J. G. SUN, *Matrix Perturbation Theory*, Academic Press, Boston, 1990.
- [16] J. G. SUN, *Estimation of the separation of two matrices*, J. Comput. Math., 2 (1984), pp. 189–200.
- [17] Y. WEI AND G. WANG, *The perturbation theory for the Drazin inverse and its applications*, Linear Algebra Appl., 258 (1997), pp. 179–186.
- [18] Y. WEI, *On the perturbation of the group inverse and oblique projection*, Appl. Math. Comput., 98 (1999), pp. 29–42.
- [19] Y. WEI, *The Drazin inverse of updating of a square matrix with application to perturbation formula*, Appl. Math. Comput., 108 (2000), pp. 77–83.
- [20] Y. WEI AND H. WU, *The perturbation of the Drazin inverse and oblique projection*, Appl. Math. Lett., 13 (2000), pp. 77–83.
- [21] Y. WEI AND H. WU, *Challenging problems on the perturbation of Drazin inverse*, Ann. Oper. Res., 103 (2001), pp. 371–378.
- [22] Y. WEI, *Perturbation bound of the Drazin inverse*, Appl. Math. Comput., 125 (2002), pp. 231–244.
- [23] Y. WEI AND X. LI, *An improvement on perturbation bounds for the Drazin inverse*, Numer. Linear Algebra Appl., 10 (2003), pp. 563–575.

HOW TO DEDUCE A PROPER EIGENVALUE CLUSTER FROM A PROPER SINGULAR VALUE CLUSTER IN THE NONNORMAL CASE*

STEFANO SERRA-CAPIZZANO[†], DANIELE BERTACCINI[‡], AND GENE H. GOLUB[§]

Abstract. We consider a generic sequence of matrices (the nonnormal case is of interest) showing a proper cluster at zero in the sense of the singular values. By a direct use of the notion of majorizations, we show that the uniform spectral boundedness is sufficient for the proper clustering at zero of the eigenvalues: if the assumption of boundedness is removed, then we can construct sequences of matrices with a proper singular value clustering and having all the eigenvalues of an arbitrarily big modulus. Applications to the preconditioning theory are discussed.

Key words. proper and weak clustering, majorization, preconditioning

AMS subject classifications. 65F10, 15A18, 15A12

DOI. 10.1137/040608027

1. The result. Let $\mathbf{x}$ be a generic real vector with entries x_j , $j = 1, \dots, s$. We define $\mathbf{x}^\downarrow$ with entries $x_j^\downarrow$, $j = 1, \dots, s$, to be the nonincreasing ordering of $\mathbf{x}$, i.e.,

$$x_1^\downarrow \geq x_2^\downarrow \geq \dots \geq x_s^\downarrow$$

with $\mathbf{x}^\downarrow = P\mathbf{x}$ and P a suitable s -by- s permutation matrix. A vector $\mathbf{a}$ with real entries a_j , $j = 1, \dots, s$, is weakly majorized (see, e.g., [3]) by a vector $\mathbf{b}$ with real entries b_j , $j = 1, \dots, s$, if

$$\sum_{j=1}^k \mathbf{a}^\downarrow \leq \sum_{j=1}^k \mathbf{b}^\downarrow \quad \forall k = 1, \dots, s;$$

in that case, we write $\mathbf{a} \prec_w \mathbf{b}$. Weyl's majorant theorem (see, e.g., [3, Theorem II.3.6]) establishes a fundamental majorization relation between the eigenvalues and the singular values of a given matrix $A \in M_s(\mathbf{C})$ ($M_s(\mathbf{C})$ denotes the space of the complex s -by- s matrices).

THEOREM 1.1. Let $A \in M_s(\mathbf{C})$ and let $\sigma_1 \geq \sigma_2 \geq \dots \geq \sigma_s$ and $\lambda_1, \lambda_2, \dots, \lambda_s$ be the singular values and the eigenvalues of A , respectively. Let $\phi: \mathbf{R}_+ \rightarrow \mathbf{R}_+$ be a continuous function such that $\phi(\exp(t))$ is increasing for all $t \in \mathbf{R}$. Then

$$\phi(|\lambda|) \prec_w \phi(\sigma)$$

where $\phi(|\lambda|)$ and $\phi(\sigma)$ are the vectors with entries $\phi(|\lambda_j|)$ and $\phi(\sigma_j)$, respectively, for $j = 1, \dots, s$.

*Received by the editors May 11, 2004; accepted for publication (in revised form) November 1, 2004; published electronically June 22, 2005.

<http://www.siam.org/journals/simax/27-1/60802.html>

[†]Dipartimento di Fisica e Matematica, Università dell'Insubria, Via Valleggio 11, 22100 Como, Italy (stefano.serrac@uninsubria.it, serra@mail.dm.unipi.it). The work of this author was partially supported by MIUR grant 2002014121.

[‡]Dipartimento di Matematica, Università di Roma "La Sapienza," P.le A. Moro 2, 00185 Roma, Italy (bertaccini@mat.uniroma1.it). The work of this author was partially supported by MIUR grant 2002014121.

[§]Department of Computer Science, Stanford University, Gates 2B, CA 94305 (golub@stanford.edu).

The former result is now used in a context of asymptotic linear algebra for deducing eigenvalue clustering from the singular value clustering: we recall that a sequence A_n is ϵ -clustered at $\alpha \in \mathbf{R}$ if for every $\epsilon > 0$ there exists a pure constant c_ϵ such that

$$\begin{aligned} \#\{\text{singular values of } A_n \notin (\alpha - \epsilon, \alpha + \epsilon)\} &\leq c_\epsilon \quad (\text{singular values proper clustering}), \\ \#\{\text{eigenvalues of } A_n \notin (\alpha - \epsilon, \alpha + \epsilon)\} &\leq c_\epsilon \quad (\text{eigenvalues proper clustering}). \end{aligned}$$

THEOREM 1.2. *Let $A_n \in M_n(\mathbf{C})$ be a sequence of matrices such that A_n is ϵ -clustered at zero. Let $\lambda_n = (\lambda_1^{(n)}, \dots, \lambda_n^{(n)})^T$ and $\sigma_n = (\sigma_1^{(n)}, \dots, \sigma_n^{(n)})^T$ be the (ordered) vectors of the eigenvalues and singular values of A_n , respectively, where $\sigma_1^{(n)} \geq \sigma_2^{(n)} \geq \dots \geq \sigma_n^{(n)}$ and $|\lambda_1^{(n)}| \geq |\lambda_2^{(n)}| \geq \dots \geq |\lambda_n^{(n)}|$. From the assumptions we know that there exists M a positive constant independent of n , and, for every $\epsilon > 0$, there exists $\bar{n} = \bar{n}_\epsilon$ such that*

$$(1.1) \quad \|A_n\| = \sigma_1^{(n)} \leq M \quad \forall n \quad (\text{uniform boundedness}),$$

$$(1.2) \quad \sigma_{\bar{n}}^{(n)} < \epsilon \quad \forall n \geq \bar{n} \quad (\text{singular value clustering at zero}).$$

Now we would like to prove the spectral clustering at zero of the eigenvalues. In the normal case this is a trivial result since $\sigma_j^{(n)} = |\lambda_j^{(n)}|$, $j = 1, \dots, n$, while, in the general case, the essential tool is Weyl's majorant theorem: Theorem 1.1 with $\phi(z) = z$. In such a way, for every $k = 1, \dots, n$, we have

$$(1.3) \quad \sum_{j=1}^k |\lambda_j^{(n)}| \leq \sum_{j=1}^k \sigma_j^{(n)}.$$

By contradiction, we suppose that the eigenvalues of A_n are not properly clustered at zero. Therefore there exists a sequence of positive integers α_n monotonically going to infinity, and there exists $c > 0$ (independent of n) such that $|\lambda_{\alpha_n}^{(n)}| > c$ at least for a subsequence $n = n_q$ (where n_q is a strictly increasing sequence of integers). As a consequence, for every q large enough, we deduce

$$\sum_{j=1}^{\alpha_{n_q}} |\lambda_j^{(n_q)}| > c\alpha_{n_q}$$

and, due to (1.1) and (1.2), simultaneously we have

$$\sum_{j=1}^{\alpha_{n_q}} \sigma_j^{(n_q)} < M\bar{n} + (\alpha_{n_q} - \bar{n})\epsilon, \quad \alpha_{n_q} > \bar{n}.$$

Thus, by putting together (1.3) with $k = \alpha_{n_q}$ and the latter two inequalities, we find $c\alpha_{n_q} < M\bar{n} + (\alpha_{n_q} - \bar{n})\epsilon$ for every q large enough. Finally, by dividing by α_{n_q} and by making the limit as q tends to infinity (α_{n_q} will go to infinity as well), we conclude that for every $\epsilon > 0$, there exists $\bar{n} = \bar{n}_\epsilon$ (independent of $n = n_q$ and therefore of q) such that $c \leq \epsilon$. Since c is a pure positive constant and $\epsilon > 0$ is arbitrary, the choice of $\epsilon = c/2$ leads to $c \leq c/2$ with $c > 0$, which is a contradiction, and the proof is complete. $\square$

2. Discussion and applications. The first observation concerns the essentiality of the assumptions of Theorem 1.2. If the singular values of A_n are not properly clustered at zero and the matrices A_n are definitely normal, then the eigenvalues cannot be properly clustered at zero since the singular values coincide with the absolute value of the eigenvalues. More interestingly, we will see that the hypothesis of the spectral boundedness is also essential. Let ϵ_n be a sequence of positive numbers and let B_n be the sequence of matrices given by

$$B_n = \begin{pmatrix} 0 & \epsilon_1 & 0 & \dots & 0 \\ \vdots & \ddots & \ddots & \ddots & \vdots \\ & & \ddots & \ddots & 0 \\ \vdots & & & \ddots & \epsilon_{n-1} \\ 0 & \dots & & \dots & 0 \end{pmatrix}, \quad 1 \geq \epsilon_1 \geq \epsilon_2 \geq \dots \geq \epsilon_k > 0 \quad \forall k \geq 2, \quad \lim_{k \rightarrow \infty} \epsilon_k = 0.$$

A simple check shows that this sequence is spectrally clustered to zero in the sense of the eigenvalues ($\lambda_j^{(n)} = 0$ for every $j = 1, \dots, n$) and in the sense of the singular (since $\sigma_1^{(n)} = \epsilon_1 \geq \sigma_2^{(n)} = \epsilon_2 \geq \dots \geq \sigma_{n-1}^{(n)} = \epsilon_{n-1} > \sigma_n^{(n)} = 0$). Now we consider a rank one perturbation of B_n having an exploding norm. More precisely we define A_n as

$$A_n = B_n + \psi_n \mathbf{e}_n \mathbf{e}_1^T,$$

where $\mathbf{e}_k$ is the k th vector of the canonical basis of $\mathbf{C}^n$ and $\psi_n > 0$ such that

$$\lim_{n \rightarrow \infty} \left[\psi_n \left[\prod_{k=1}^{n-1} \epsilon_k \right] \right]^{1/n} = \infty.$$

By the Cauchy interlace theorem (see, e.g., [4]) applied to the singular values, we deduce immediately that the singular values of A_n are also properly clustered to zero (because the singular values of B_n are properly clustered to zero and A_n is a one-rank modification of B_n ; see [8]). However, a direct computation proves that the characteristic polynomial of A_n coincides with

$$p_{A_n}(\lambda) = (-\lambda)^n - \psi_n \left[\prod_{k=1}^{n-1} \epsilon_k \right]$$

and therefore the eigenvalues of A_n have all the same modulus, which is given by

$$\left[\psi_n \left[\prod_{k=1}^{n-1} \epsilon_k \right] \right]^{1/n}.$$

Since the latter tends to infinity as n tends to infinity, we have that the eigenvalues of B_n not only are not clustered at zero in the proper sense but are also clustered at infinity in the modulus. The responsibility for this very pathological behavior is the huge norm of the one-rank correction and the high nonnormality of B_n : the combination of these two ingredient gives (as is well known to numerical analysts) a great sensitivity to the eigenvalues, whose global distribution shows a dramatic change, i.e., from a proper clustering to zero for B_n to a proper clustering to infinity for

A_n (see [5, section 7] for an illustration of the numerical difficulties related to similar examples).

The next step is to show some consequences of Theorem 1.2, and a very natural and important application is the study of the convergence behavior of Krylov methods when nonnormal iteration matrices are involved. In [2] we considered the preconditioning of non-Hermitian (but positive definite) matrices coming from convection-diffusion equations. If B_n denotes the sequence of the discretized problems and P_n is the sequence of preconditioners (Hermitian and positive definite), setting $B_n = \text{Re}(B_n) + \mathbf{i}\text{Im}(B_n)$, $\mathbf{i}^2 = -1$, $\text{Re}(B_n)$, $\text{Im}(B_n)$ Hermitian matrices (the real and the imaginary part of B_n , i.e., $\text{Re}(B_n) = (B_n + B_n^*)/2$, $\text{Im}(B_n) = (B_n - B_n^*)/(2\mathbf{i})$), we have proved the following:

F1 $P_n^{-1}\text{Re}(B_n)$ has eigenvalues properly clustered to 1 and lying in a strictly positive uniformly bounded interval.

F2 $P_n^{-1}\text{Im}(B_n)$ has eigenvalues properly clustered to 0 and lying in a uniformly bounded interval.

From these two items we prove the proper clustering at 1 of the eigenvalues of $P_n^{-1}B_n$ by using Theorem 1.2.

PROPOSITION 2.1.

F1 **F2** $P_n^{-1}B_n$ has eigenvalues properly clustered to 1 and lying in a strictly positive uniformly bounded interval.

We first observe that the matrices $P_n^{-1/2}\text{Re}(B_n)P_n^{-1/2}$, $P_n^{-1/2}\text{Im}(B_n)P_n^{-1/2}$ are both Hermitian since P_n is Hermitian positive definite (so that by the Schur canonical decomposition, its square root exists and is Hermitian positive definite) and $\text{Re}(B_n)$, $\text{Im}(B_n)$ are Hermitian by construction. Moreover, $P_n^{-1}\text{Re}(B_n)$ is similar to $P_n^{-1/2}\text{Re}(B_n)P_n^{-1/2}$ and $P_n^{-1}\text{Im}(B_n)$ is similar to $P_n^{-1/2}\text{Im}(B_n)P_n^{-1/2}$: as a consequence both the statements contained in F1 and F2 are true for $P_n^{-1/2}\text{Re}(B_n)P_n^{-1/2}$ and $P_n^{-1/2}\text{Im}(B_n)P_n^{-1/2}$, respectively. Therefore we can deduce properties on the singular values since the involved matrices are Hermitian (and a fortiori normal); more precisely, both $P_n^{-1/2}\text{Re}(B_n)P_n^{-1/2} - I_n$ and $P_n^{-1/2}\text{Im}(B_n)P_n^{-1/2}$ are spectrally uniformly bounded and properly clustered to zero in the singular value sense. From this, by using the field of values notion (see [4]), and since (by F1) the eigenvalues of $P_n^{-1/2}\text{Re}(B_n)P_n^{-1/2}$ belong to a strictly positive uniformly bounded interval, all the eigenvalues of $P_n^{-1}B_n$ lie in a uniformly bounded rectangle with positive real part well separated from zero. Moreover, by a direct SVD inspection, we see that the sequence

$$P_n^{-1/2}\text{Re}(B_n)P_n^{-1/2} - I_n + \mathbf{i}P_n^{-1/2}\text{Im}(B_n)P_n^{-1/2}$$

is spectrally uniformly bounded and properly clustered to zero in the singular value sense. As a consequence, all the assumptions of Theorem 1.2 are fulfilled with $A_n = P_n^{-1/2}\text{Re}(B_n)P_n^{-1/2} - I_n + \mathbf{i}P_n^{-1/2}\text{Im}(B_n)P_n^{-1/2}$, and thus the sequence A_n is also properly clustered to zero in the sense of the eigenvalues. Finally, we complete the proof by noticing that $A_n + I_n$ is properly clustered to 1 in the eigenvalue sense and

$$P_n^{-1}B_n = P_n^{-1/2}(A_n + I_n)P_n^{1/2} \quad (\text{a similarity transformation}). \quad \square$$

The conclusions are in some sense contained in the former proposition: the discussed result could be the key to proving eigenvalue clustering when the preconditioner is Hermitian positive definite but the original problem is nonnormal. Some of these

situations occur when dealing with partial differential equations and structured matrices (see, e.g., [2, 7]); in particular, concerning the aforementioned applications, we stress that there exist many tools for proving the singular value clustering in the non-normal case [8, 6, 7], but not so many for dealing with the eigenvalues: in this direction we must mention [9], where deep and beautiful results are proven with regard to the notion of general clustering, i.e., up to $o(n)$ outliers; unfortunately, from the viewpoint of a fine convergence analysis of Krylov methods, we remark that often only the proper clustering studied in this note is of interest. Indeed, the weak clustering can be useful only when there is an estimate of the number of outliers which is described by a mildly growing function (for instance, a poly-logarithm of n). Furthermore, in both the cases (proper clustering or poly-logarithmic number of outlying eigenvalues) and for practical purposes, attention has to be paid to the multiplicative constants: as a matter of fact, the practical dependency on $\epsilon > 0$ is such that it usually appears in a denominator, and this represents a delicate point in a convergence analysis especially in the partial differential equations context where ϵ can be a function of the finesse parameter and therefore of n . Finally, we conclude by observing that future work should investigate the direction of providing specific tools in the case where the preconditioning sequence is constituted by nonnormal matrices (an attempt is contained in Theorem 4.3 of [1]).

REFERENCES

- [1] D. BERTACCINI, G. GOLUB, AND S. SERRA CAPIZZANO, *Spectral Analysis and Superlinear Convergence of a Preconditioned Iterative Method for the Convection-Diffusion Equation*, reprint SCCM-04-11, Stanford University, Stanford, CA.
- [2] D. BERTACCINI, G. H. GOLUB, S. SERRA CAPIZZANO, AND C. TABLINO POSSIO, *Preconditioned HSS methods for the solution of non-Hermitian positive definite linear systems and applications to the discrete convection-diffusion equation*, Numer. Math., 99 (2005), pp. 441–484.
- [3] R. BHATIA, *Matrix Analysis*, Springer-Verlag, New York, 1997.
- [4] G. GOLUB AND C. VAN LOAN, *Matrix Computations*, The Johns Hopkins University Press, Baltimore, MD, 1983.
- [5] G. H. GOLUB AND J. H. WILKINSON, *Ill-conditioned eigensystems and the computation of the Jordan canonical form*, SIAM Rev., 18 (1976), pp. 578–619.
- [6] S. SERRA CAPIZZANO, *Spectral behavior of matrix sequences and discretized boundary value problems*, Linear Algebra Appl., 337 (2001), pp. 37–78.
- [7] S. SERRA CAPIZZANO, *Generalized locally Toeplitz sequences: Spectral analysis and applications to discretized partial differential equations*, Linear Algebra Appl., 366 (2003), pp. 371–402.
- [8] E. E. TYRTYSHNIKOV, *A unifying approach to some old and new theorems on distribution and clustering*, Linear Algebra Appl., 232 (1996), pp. 1–43.
- [9] E. E. TYRTYSHNIKOV AND N. L. ZAMARASHKIN, *On the eigen and singular value clusters*, Calcolo, 33 (1996), pp. 71–78.

NONLINEAR DISCRIMINANT ANALYSIS USING KERNEL FUNCTIONS AND THE GENERALIZED SINGULAR VALUE DECOMPOSITION*

CHEONG HEE PARK[†] AND HAESUN PARK[‡]

Abstract. Linear discriminant analysis (LDA) has been widely used for linear dimension reduction. However, LDA has limitations in that one of the scatter matrices is required to be nonsingular and the nonlinearly clustered structure is not easily captured. In order to overcome the problems caused by the singularity of the scatter matrices, a generalization of LDA based on the generalized singular value decomposition (GSVD) was recently developed. In this paper, we propose a nonlinear discriminant analysis based on the kernel method and the GSVD. The GSVD is applied to solve the generalized eigenvalue problem which is formulated in the feature space defined by a nonlinear mapping through kernel functions. Our GSVD-based kernel discriminant analysis is theoretically compared with other kernel-based nonlinear discriminant analysis algorithms. The experimental results show that our method is an effective nonlinear dimension reduction method.

Key words. dimension reduction, generalized singular value decomposition, kernel functions, linear discriminant analysis, nonlinear discriminant analysis

AMS subject classifications. 15A09, 68T10, 62H30, 65F15, 15A18

DOI. 10.1137/S0895479804442334

1. Introduction. In linear discriminant analysis (LDA), a linear transformation is found which maximizes the between-class scatter and minimizes the within-class scatter [7, 9]. Although LDA is conceptually simple and has been used in many application areas, it has some limitations: it requires at least one of the scatter matrices to be nonsingular and it cannot easily capture a nonlinearly clustered structure.

One common situation where all of the scatter matrices are singular is when the number of data points is smaller than the dimension of the data space, and this situation is often referred to as an undersampled problem. Numerous methods have been proposed to overcome this difficulty [4, 8, 19]. Recently, a method called LDA/GSVD was proposed, which is a generalization of LDA based on the generalized singular value decomposition (GSVD) [10]. It overcomes the problem caused by the singularity of the scatter matrices in undersampled problems by applying the GSVD to solve a generalized eigenvalue problem.

In order to make LDA applicable to nonlinearly structured data, kernel-based methods have been applied. The main idea of kernel-based methods is to map the input data to a feature space by a nonlinear mapping where inner products in the feature space can be computed by a kernel function without knowing the nonlinear mapping explicitly. Kernel principal component analysis (kernel PCA) [18], kernel

*Received by the editors March 19, 2004; accepted for publication (in revised form) by L. Eldén November 23, 2004; published electronically July 13, 2005. This material is based upon work supported by National Science Foundation grants CCR-0204109 and ACI-0305543. Any opinions, findings, and conclusions or recommendations expressed in this material are those of the authors and do not necessarily reflect the views of the National Science Foundation.

<http://www.siam.org/journals/simax/27-1/44233.html>

[†]Department of Computer Science and Engineering, Chungnam National University, 220 Gung-dong, Yuseong-gu, Daejeon, 305-764, South Korea (cheonghee@cnu.ac.kr).

[‡]Department of Computer Science and Engineering, University of Minnesota, Minneapolis, MN 55455 (hpark@cs.umn.edu), and The National Science Foundation, 4201 Wilson Boulevard, Arlington, VA 22230. The work of this author was performed while at the National Science Foundation (NSF) and was partly supported by IR/D from the NSF.

Fisher discriminant analysis (KFD) [12], and nonlinear discriminant analysis [1, 2, 16] are kernel-based nonlinear extensions of the well-known PCA, Fisher discriminant analysis, and linear discriminant analysis, respectively. However, PCA or kernel PCA may not be appropriate as a dimension reduction method for clustered data, since the purpose of these methods is an optimal lower-dimensional representation rather than discrimination. KFD [12] and the method proposed in [2] were developed to handle data consisting of two classes only. In generalized discriminant analysis (GDA) [1], centering in the feature space is performed by shifting each vector by the global average, and then the kernel matrix is computed. Although this kernel matrix is assumed to be nonsingular, centering in the feature space makes the kernel matrix singular, even when the kernel function is symmetric positive definite. This makes the theoretical development of GDA [1] break down. In addition, when the input space is mapped to a feature space through a kernel function, the dimension of the feature space often becomes much larger than that of the original data space, and as a result the scatter matrices become singular.

Towards a general nonlinear discriminant analysis, we propose a kernel-based nonlinear extension of LDA using the GSVD. We also show the relationships of our GSVD-based kernel discriminant analysis with other kernel-based nonlinear discriminant analysis algorithms. After reviewing the linear dimension reduction method LDA/GSVD in section 2, we present the new kernel discriminant analysis, KDA/GSVD, in section 3. The relationships of KDA/GSVD with other kernel-based methods are discussed in section 4. Experimental results are given in section 5, and we conclude with discussions.

2. LDA. Throughout the paper, a data set of n data vectors in an m -dimensional space is denoted as

$$(2.1) \quad A = [a_1, \dots, a_n] = [A_1, \dots, A_r] \in \mathbf{R}^{m \times n},$$

where the data is clustered to r classes and each block $A_i \in \mathbf{R}^{m \times n_i}$ has n_i data vectors. Let N_i ($1 \leq i \leq r$) be the set of column indices that belong to the class i . The between-class scatter matrix S_b and the within-class scatter matrix S_w are defined as

$$(2.2) \quad S_b = \sum_{i=1}^r n_i (c_i - c)(c_i - c)^T \quad \text{and} \quad S_w = \sum_{i=1}^r \sum_{j \in N_i} (a_j - c_i)(a_j - c_i)^T,$$

where

$$(2.3) \quad c_i = \frac{1}{n_i} \sum_{j \in N_i} a_j \quad \text{and} \quad c = \frac{1}{n} \sum_{j=1}^n a_j$$

are the centroid of the class i and the global centroid, respectively. The separability of classes in a data set can be measured by using the traces of these scatter matrices.

The goal of LDA is to find a transformation matrix $G \in \mathbf{R}^{m \times l}$ for some integer l with $l \leq m$ that defines a linear transformation

$$G^T : a \in \mathbf{R}^{m \times 1} \rightarrow y = G^T a \in \mathbf{R}^{l \times 1}$$

and preserves the cluster structure by maximizing the between-class scatter and minimizing the within-class scatter. In the space transformed by G^T , the between-class scatter matrix $\tilde{S}_b$ and the within-class scatter matrix $\tilde{S}_w$ become

$$\tilde{S}_b = G^T S_b G \quad \text{and} \quad \tilde{S}_w = G^T S_w G,$$

respectively. A commonly used criterion in LDA for finding an optimal clustered structure preserving transformation G^T is

$$(2.4) \quad \max_G \text{trace}((G^T S_w G)^{-1} (G^T S_b G)).$$

It is well known [9] that this criterion is satisfied when $l = r - 1$, where r is the number of the classes in the data, and the columns of $G \in \mathbf{R}^{m \times (r-1)}$ are the eigenvectors corresponding to the $r - 1$ largest eigenvalues for the eigenvalue problem

$$(2.5) \quad S_w^{-1} S_b x = \lambda x.$$

However, as in many applications such as information retrieval [11] and face recognition [4], when the number of data items is smaller than the dimension of data space, S_w becomes singular. Recently, a method which applies the GSVD to solve the generalized eigenvalue problem

$$(2.6) \quad S_b x = \lambda S_w x$$

was developed [10].

The method in [10] utilizes representations of the scatter matrices as

$$(2.7) \quad S_b = H_b H_b^T \quad \text{and} \quad S_w = H_w H_w^T,$$

where

$$H_b = [\sqrt{n_1}(c_1 - c), \dots, \sqrt{n_r}(c_r - c)] \in \mathbf{R}^{m \times r},$$

$$H_w = [A_1 - c_1 e_1^T, \dots, A_r - c_r e_r^T] \in \mathbf{R}^{m \times n} \quad \text{and} \quad e_i = [1, \dots, 1]^T \in \mathbf{R}^{n_i \times 1}.$$

By applying the GSVD to the pair (H_b^T, H_w^T) , we have

$$(2.8) \quad U^T H_b^T X = \begin{bmatrix} \underbrace{\Sigma_b}_t & \underbrace{0}_{m-t} \end{bmatrix} \quad \text{and} \quad V^T H_w^T X = \begin{bmatrix} \underbrace{\Sigma_w}_t & \underbrace{0}_{m-t} \end{bmatrix}, \quad t = \text{rank} \left(\begin{bmatrix} H_b^T \\ H_w^T \end{bmatrix} \right),$$

where U and V are orthogonal and X is nonsingular, $\Sigma_b^T \Sigma_b + \Sigma_w^T \Sigma_w = I_t$, and $\Sigma_b^T \Sigma_b$ and $\Sigma_w^T \Sigma_w$ are diagonal matrices with nonincreasing and nondecreasing diagonal components, respectively. Then the simultaneous diagonalizations of S_b and S_w can be obtained as

$$(2.9) \quad X^T S_b X = \begin{bmatrix} \Sigma_b^T \Sigma_b & 0 \\ 0 & 0 \end{bmatrix}, \quad X^T S_w X = \begin{bmatrix} \Sigma_w^T \Sigma_w & 0 \\ 0 & 0 \end{bmatrix}.$$

Let us denote the diagonal elements of $\Sigma_b^T \Sigma_b$ and $\Sigma_w^T \Sigma_w$ as η_i and ζ_i , i.e.,

$$\Sigma_b^T \Sigma_b = \text{diag}(\eta_1, \dots, \eta_t), \quad \eta_1 \geq \dots \geq \eta_t, \quad \text{and}$$

$$\Sigma_w^T \Sigma_w = \text{diag}(\zeta_1, \dots, \zeta_t), \quad \zeta_1 \leq \dots \leq \zeta_t,$$

and $X = [x_1, \dots, x_m]$. From (2.9), the column vectors x_i of X satisfy

$$(2.10) \quad \zeta_i S_b x_i = \eta_i S_w x_i, \quad 1 \leq i \leq t.$$

For $t + 1 \leq i \leq m$

$$x_i^T S_b x_i = 0 \quad \text{and} \quad x_i^T S_w x_i = 0,$$

and they do not convey discriminative information among classes. Hence an optimal dimension reducing transformation G^T can be obtained from the first $r - 1$ columns¹ of X as

$$G = [x_1, \dots, x_{r-1}].$$

The algorithm for LDA/GSVD can be found in [10, 14].

3. Nonlinear discriminant analysis based on kernel functions and the GSVD. In this section, we present a nonlinear extension of LDA based on kernel functions and the GSVD. The main idea of the kernel method is that without knowing the nonlinear feature mapping or the mapped feature space explicitly, we can work on the feature space through kernel functions, as long as the problem formulation depends only on the inner products between data points. This is based on the fact that for any kernel function κ satisfying Mercer's condition [5], there exists a mapping Φ such that

$$(3.1) \quad \langle \Phi(a), \Phi(b) \rangle = \kappa(a, b),$$

where $\langle \cdot, \cdot \rangle$ is an inner product in the feature space transformed by Φ [17, 3]. For a finite data set $\{a_1, \dots, a_n\}$, a kernel function κ satisfying Mercer's condition can be rephrased as the kernel matrix $K = [\kappa(a_i, a_j)]_{1 \leq i, j \leq n}$ being positive semidefinite [5]. The polynomial kernel

$$(3.2) \quad \kappa(x, y) = (\gamma_1(x \cdot y) + \gamma_2)^d, \quad d > 0 \text{ and } \gamma_1, \gamma_2 \in \mathbf{R},$$

and the Gaussian kernel

$$(3.3) \quad \kappa(x, y) = \exp(-\|x - y\|^2/\sigma), \quad \sigma \in \mathbf{R},$$

are two of the most widely used kernel functions. The feature map Φ can be either linear or nonlinear depending on kernel functions used. If the inner product kernel function $\kappa(x, y) = x \cdot y$ is used, the feature map is an identity map. In the kernel methods neither the feature map nor the feature space needs to be formed explicitly due to the relation (3.1) once the kernel function κ is known.

We apply the kernel method to perform LDA in the feature space instead of the original input space. Given a kernel function κ , let Φ be a mapping satisfying (3.1) and define $\mathcal{F} \subset \mathbf{R}^N$ to be the feature space from the mapping Φ . As in (2.7), scatter matrices $\mathcal{S}_b$ and $\mathcal{S}_w$ in the feature space $\mathcal{F}$ can be expressed as

$$(3.4) \quad \mathcal{S}_b = \mathcal{H}_b \mathcal{H}_b^T \quad \text{and} \quad \mathcal{S}_w = \mathcal{H}_w \mathcal{H}_w^T,$$

where

$$(3.5) \quad \begin{aligned} \mathcal{H}_b &= [\sqrt{n_1}(\tilde{c}_1 - \tilde{c}), \dots, \sqrt{n_r}(\tilde{c}_r - \tilde{c})] \in \mathbf{R}^{N \times r}, \\ \mathcal{H}_w &= [\Phi(A_1) - \tilde{c}_1 e_1^T, \dots, \Phi(A_r) - \tilde{c}_r e_r^T] \in \mathbf{R}^{N \times n}, \\ \tilde{c}_i &= \frac{1}{n_i} \sum_{j \in N_i} \Phi(a_j), \quad \tilde{c} = \frac{1}{n} \sum_{i=1}^n \Phi(a_i), \quad \text{and} \quad e_i = [1, \dots, 1]^T \in \mathbf{R}^{n_i \times 1}. \end{aligned}$$

¹When $l = \text{rank}(S_b)$, $x_i^T S_b x_i = 0$ and $x_i^T S_w x_i = 1$ for $l + 1 \leq i \leq t$. Hence one can consider taking only the first l columns from X . However, in practice the rank of S_b is one less than the number of classes, that is, $r - 1$ in most of the real data sets.

The notation $\Phi(A_i)$ is used to denote $\Phi([a_j, \dots, a_k]) = [\Phi(a_j), \dots, \Phi(a_k)]$. Then the LDA in $\mathcal{F}$ finds a transformation matrix

$$\mathcal{G} = [\varphi_1, \dots, \varphi_{r-1}] \in \mathbf{R}^{N \times (r-1)},$$

where the columns of $\mathcal{G}$ are the generalized eigenvectors corresponding to the $r - 1$ largest eigenvalues of

$$(3.6) \quad \mathcal{S}_b \varphi = \lambda \mathcal{S}_w \varphi.$$

Now we show how to solve the problem (3.6) without knowing the explicit representation of the mapping Φ and the feature space $\mathcal{F}$, therefore without forming $\mathcal{S}_b$ and $\mathcal{S}_w$ explicitly. Note that any vector $\varphi \in \mathbf{R}^{N \times 1}$ can be represented as

$$\varphi = \varphi_1 + \varphi_2,$$

where $\varphi_1 \in \text{span}\{\Phi(A)\}$ and $\varphi_2 \in \text{span}\{\Phi(A)\}^\perp$, and $\mathcal{S}_b \varphi_2 = 0$ and $\mathcal{S}_w \varphi_2 = 0$ for any $\varphi_2 \in \text{span}\{\Phi(A)\}^\perp$. Therefore, for any vector φ satisfying (3.6),

$$\mathcal{S}_b \varphi_1 = \mathcal{S}_b(\varphi_1 + \varphi_2) = \lambda \mathcal{S}_w(\varphi_1 + \varphi_2) = \lambda \mathcal{S}_w \varphi_1.$$

Hence we can restrict the solution space for (3.6) to $\text{span}\{\Phi(A)\}$. One may refer to [13] for an alternative explanation.

Let φ be represented as a linear combination of $\Phi(a_i)$, $i = 1, \dots, n$,

$$(3.7) \quad \varphi = \sum_{i=1}^n \alpha_i \Phi(a_i) \quad \text{and} \quad \alpha = [\alpha_1, \dots, \alpha_n]^T.$$

The following theorem gives a formula by which $\mathcal{S}_b$ can be expressed through the kernel function.

THEOREM 3.1.

$$(3.8) \quad \mathcal{K}_b = [b_{ij}]_{(1 \leq i \leq n, 1 \leq j \leq r)}, \quad b_{ij} = \sqrt{n_j} \left(\frac{1}{n_j} \sum_{s \in N_j} \kappa(a_i, a_s) - \frac{1}{n} \sum_{s=1}^n \kappa(a_i, a_s) \right).$$

$$(3.9) \quad \mathcal{H}_b^T \varphi = \mathcal{K}_b^T \alpha.$$

Proof. From (3.5) and (3.7),

$$(3.10) \quad \begin{aligned} \mathcal{H}_b^T \varphi &= \begin{bmatrix} \sqrt{n_1}(\tilde{c}_1 - \tilde{c})^T \\ \vdots \\ \sqrt{n_r}(\tilde{c}_r - \tilde{c})^T \end{bmatrix} \left(\sum_{i=1}^n \alpha_i \Phi(a_i) \right) \\ &= \begin{bmatrix} \sqrt{n_1}(\frac{1}{n_1} \sum_{s \in N_1} \Phi(a_s) - \frac{1}{n} \sum_{s=1}^n \Phi(a_s))^T \\ \vdots \\ \sqrt{n_r}(\frac{1}{n_r} \sum_{s \in N_r} \Phi(a_s) - \frac{1}{n} \sum_{s=1}^n \Phi(a_s))^T \end{bmatrix} \\ &\quad \times [\Phi(a_1), \dots, \Phi(a_n)] \begin{bmatrix} \alpha_1 \\ \vdots \\ \alpha_n \end{bmatrix} \\ &= \mathcal{K}_b^T \alpha. \quad \square \end{aligned}$$

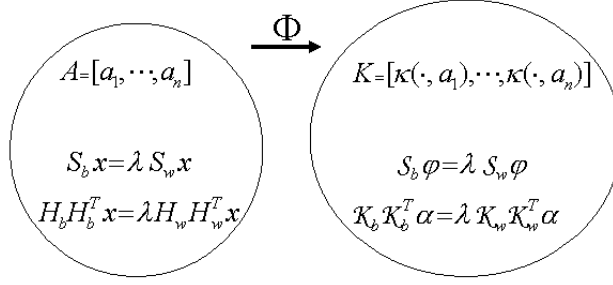


FIG. 3.1. The problem formulations of LDA in the original data space and the feature space defined by a nonlinear mapping Φ through a kernel function, $\langle \Phi(a_i), \Phi(a_j) \rangle = \kappa(a_i, a_j)$. In the kernel matrix K , $\kappa(\cdot, a_i)$ denotes a column vector $[\kappa(a_1, a_i), \dots, \kappa(a_n, a_i)]^T$.

Similarly to Theorem 3.1, we can obtain

$$(3.11) \quad \mathcal{H}_w^T \varphi = \mathcal{K}_w^T \alpha,$$

where

$$(3.12) \quad \mathcal{K}_w = [w_{ij}]_{(1 \leq i \leq n, 1 \leq j \leq n)},$$

$$w_{ij} = \kappa(a_i, a_j) - \frac{1}{n_l} \sum_{s \in N_l} \kappa(a_i, a_s) \quad \text{when } a_j \text{ belongs to the class } l.$$

THEOREM 3.2. Let $\mathcal{H}_b$ and $\mathcal{H}_w$ be the between-class and within-class scatter matrices in the feature space, respectively. Then, the LDA problem in the feature space can be formulated as

$$\mathcal{S}_b \varphi = \lambda \mathcal{S}_w \varphi$$

where

$$(3.13) \quad \mathcal{K}_b \mathcal{K}_b^T \alpha = \lambda \mathcal{K}_w \mathcal{K}_w^T \alpha,$$

$$\varphi = \sum_{i=1}^n \alpha_i \Phi(a_i) \quad \alpha = [\alpha_1, \dots, \alpha_n]^T$$

From (3.9) and (3.11),

$$(3.14) \quad \begin{aligned} \mathcal{S}_b \varphi = \lambda \mathcal{S}_w \varphi &\Leftrightarrow \psi^T \mathcal{H}_b \mathcal{H}_b^T \varphi = \lambda \psi^T \mathcal{H}_w \mathcal{H}_w^T \varphi \\ &\Leftrightarrow \beta^T \mathcal{K}_b \mathcal{K}_b^T \alpha = \lambda \beta^T \mathcal{K}_w \mathcal{K}_w^T \alpha \\ &\Leftrightarrow \mathcal{K}_b \mathcal{K}_b^T \alpha = \lambda \mathcal{K}_w \mathcal{K}_w^T \alpha \end{aligned}$$

for any $\psi = \sum_{i=1}^n \beta_i \Phi(a_i)$ and $\beta = [\beta_1, \dots, \beta_n]^T$. $\square$

Note that $\mathcal{K}_b \mathcal{K}_b^T$ and $\mathcal{K}_w \mathcal{K}_w^T$ can be viewed as the between-class scatter matrix and within-class scatter matrix of the kernel matrix

$$K = [\kappa(a_i, a_j)]_{(1 \leq i \leq n, 1 \leq j \leq n)}$$

when each column in K is considered as a data point in the n -dimensional space. It can be observed by comparing the structures of $\mathcal{K}_b$ and $\mathcal{K}_w$ with those of $\mathcal{H}_b$ and $\mathcal{H}_w$ in (3.5). Figure 3.1 illustrates the corresponding relations in the original data space and the feature space.

Note that $\mathcal{K}_b \mathcal{K}_b^T$ and $\mathcal{K}_w \mathcal{K}_w^T$ are both singular and the classical LDA cannot be applied. Now we apply the GSVD to the pair $(\mathcal{K}_b^T, \mathcal{K}_w^T)$ in order to solve (3.13), and as in (2.9) we have

$$(3.15) \quad \mathcal{X}^T \mathcal{K}_b \mathcal{K}_b^T \mathcal{X} = \begin{bmatrix} \Gamma_b^T \Gamma_b & 0 \\ 0 & 0 \end{bmatrix}, \quad \mathcal{X}^T \mathcal{K}_w \mathcal{K}_w^T \mathcal{X} = \begin{bmatrix} \Gamma_w^T \Gamma_w & 0 \\ 0 & 0 \end{bmatrix},$$

where the columns of $\mathcal{X}$ solve (3.13). Let $\mathcal{G}$ be the matrix obtained by the first $r-1$ columns of $\mathcal{X}$ as

$$\mathcal{G} = [\alpha^{(1)}, \dots, \alpha^{(r-1)}] = \begin{bmatrix} \alpha_{11} & \cdots & \alpha_{1r-1} \\ \vdots & & \vdots \\ \alpha_{n1} & \cdots & \alpha_{nr-1} \end{bmatrix}.$$

Defining

$$\varphi_i = \sum_{j=1}^n \alpha_{ji} \Phi(a_j), \quad 1 \leq i \leq r-1,$$

by Theorem 3.2, φ_i satisfies

$$\mathcal{S}_b \varphi_i = \lambda_i \mathcal{S}_w \varphi_i$$

and $[\varphi_1, \dots, \varphi_{r-1}]$ gives a linear transformation by LDA in the feature space. Hence, for any input vector $z \in \mathbf{R}^{m \times 1}$, the dimension reduced representation of z is given by

$$\begin{aligned} & [\varphi_1, \dots, \varphi_{r-1}]^T \Phi(z) \\ &= \begin{bmatrix} \sum_{j=1}^n \alpha_{j1} \Phi(a_j)^T \Phi(z), \dots, \sum_{j=1}^n \alpha_{jr-1} \Phi(a_j)^T \Phi(z) \end{bmatrix} = \mathcal{G}^T \begin{bmatrix} \kappa(a_1, z) \\ \vdots \\ \kappa(a_n, z) \end{bmatrix}. \end{aligned}$$

This method, called KDA/GSVD, is summarized in Algorithm 1.

4. Comparison of kernel-based nonlinear discriminant analysis algorithms. In section 3, we showed that KDA/GSVD finds the solution by solving

$$(4.1) \quad \mathcal{K}_b \mathcal{K}_b^T \alpha = \lambda \mathcal{K}_w \mathcal{K}_w^T \alpha$$

using the GSVD. In this section, we compare our GSVD-based approach with two other methods—the regularization-based method [8] and the one based on the minimum squared error function [6, 7]—and derive the relationships of KDA/GSVD with other kernel-based nonlinear discriminant analysis algorithms.

4.1. A relationship to KFD. Mika et al. [12] developed a nonlinear extension of Fisher discriminant analysis, kernel Fisher discriminant analysis (KFD), using the regularization and kernel methods. KFD is a method for finding a dimension reducing transformation specifically when the data set has only two classes. In a two-class problem, the between-class scatter matrix $\mathcal{S}_b$ is expressed as

$$\mathcal{S}_b = \frac{n_1 n_2}{n} (\tilde{c}_1 - \tilde{c}_2)(\tilde{c}_1 - \tilde{c}_2)^T,$$

ALGORITHM 1. KDA/GSVD.

Given a data matrix $A = [a_1, \dots, a_n] \in \mathbf{R}^{m \times n}$ with r classes and a kernel function κ , it computes the $(r-1)$ -dimensional representation of any input vector $z \in \mathbf{R}^m$ by applying the GSVD in the feature space defined by a feature mapping Φ satisfying $\kappa(a_i, a_j) = \langle \Phi(a_i), \Phi(a_j) \rangle$.

1. Compute $\mathcal{K}_b \in \mathbf{R}^{n \times r}$ and $\mathcal{K}_w \in \mathbf{R}^{n \times n}$ according to (3.8) and (3.12).
2. Apply the GSVD to the pair $(\mathcal{K}_b^T, \mathcal{K}_w^T)$:

$$\mathcal{U}^T \mathcal{K}_b^T \mathcal{X} = [\Gamma_b \quad 0] \quad \text{and} \quad \mathcal{V}^T \mathcal{K}_w^T \mathcal{X} = [\Gamma_w \quad 0].$$

3. Assign the first $r-1$ columns of $\mathcal{X}$ to $\mathcal{G}$:
4. For any input vector $z \in \mathbf{R}^{m \times 1}$, a dimension reduced representation is computed as

$$\mathcal{G}^T \begin{bmatrix} \kappa(a_1, z) \\ \vdots \\ \kappa(a_n, z) \end{bmatrix} \in \mathbf{R}^{(r-1) \times 1}.$$

where $\tilde{c}_i$, $i = 1, 2$, denote the class centroids. Hence the KFD criterion, maximization of

$$(4.2) \quad \frac{\varphi^T (\tilde{c}_1 - \tilde{c}_2) (\tilde{c}_1 - \tilde{c}_2)^T \varphi}{\varphi^T \mathcal{S}_w \varphi},$$

is equivalent to maximization of

$$(4.3) \quad \frac{\varphi^T \mathcal{S}_b \varphi}{\varphi^T \mathcal{S}_w \varphi} = \frac{\alpha^T \mathcal{K}_b \mathcal{K}_b^T \alpha}{\alpha^T \mathcal{K}_w \mathcal{K}_w^T \alpha} \equiv \lambda,$$

where $\varphi = \sum_{i=1}^n \alpha_i \Phi(a_i)$, $\alpha = [\alpha_1, \dots, \alpha_n]^T$. Setting the derivative of (4.3) with respect to α as 0 gives the eigenvalue problem

$$(4.4) \quad \mathcal{K}_b \mathcal{K}_b^T \alpha = \lambda \mathcal{K}_w \mathcal{K}_w^T \alpha.$$

The KFD in [12] solves the problem (4.2), that is, the eigenvalue problem (4.4) by the regularization method where a positive diagonal matrix dI is added to $\mathcal{K}_w \mathcal{K}_w^T$ to make it nonsingular. However, the regularization parameter should be determined experimentally and this procedure can be expensive. The performances by the regularization method and KDA/GSVD are compared in our experiments.

4.2. Using the minimum squared error function. The minimum squared error (MSE) formulation in a two-class problem (i.e., $r = 2$) seeks a linear discriminant function

$$g(z) = w_0 + w^T z,$$

where

$$(4.5) \quad g(a_i) = w_0 + w^T a_i = \begin{cases} b_1 & \text{if } i \in N_1, \\ b_2 & \text{if } i \in N_2 \end{cases}$$

and b_i is the prespecified number for each class. For the data set A given in (2.1), the problem (4.5) can be reformulated as a problem of minimizing the squared error

$$(4.6) \quad \left\| \begin{bmatrix} 1 & a_1^T \\ \vdots & \vdots \\ 1 & a_n^T \end{bmatrix} \begin{bmatrix} w_0 \\ w \end{bmatrix} - \begin{bmatrix} y_1 \\ \vdots \\ y_n \end{bmatrix} \right\|_2^2 \equiv \left\| P \begin{bmatrix} w_0 \\ w \end{bmatrix} - y \right\|_2^2,$$

where $y_i = b_1$ if $i \in N_1$ and $y_i = b_2$ if $i \in N_2$. Note that the matrix P is $n \times (m+1)$ and the linear system involved in (4.6) is underdetermined when $n < m+1$ and overdetermined when $n > m+1$. In either case, the solution which minimizes the squared error (4.6) can be computed using the pseudoinverse P^+ of P as

$$(4.7) \quad \begin{bmatrix} w_0 \\ w \end{bmatrix} = P^+ y.$$

When $b_1 = n/n_1$ and $b_2 = -n/n_2$, the MSE solution is related to Fisher discriminant analysis (FDA) [6, 7]. The vector w in (4.7) is the same as the solution of FDA except for a scaling factor and

$$w_0 = -w^T c,$$

where c is the global centroid defined in (2.3). For a new data item, it is assigned to class 1 if

$$(4.8) \quad w^T z + w_0 = w^T (z - c) > 0;$$

otherwise it is assigned to class 2.

Now (4.6) can be extended to the squared error function in the feature space by a mapping Φ through a kernel function κ . By substituting a_i with $\Phi(a_i)$ and w with $\sum_{j=1}^n \alpha_j \Phi(a_j)$ as in (3.7), we obtain

$$(4.9) \quad \left\| \begin{bmatrix} 1 & \kappa(a_1, a_1) & \cdots & \kappa(a_1, a_n) \\ \vdots & \vdots & \ddots & \vdots \\ 1 & \kappa(a_n, a_1) & \cdots & \kappa(a_n, a_n) \end{bmatrix} \begin{bmatrix} w_0 \\ \alpha \end{bmatrix} - \begin{bmatrix} y_1 \\ \vdots \\ y_n \end{bmatrix} \right\|_2^2 \equiv \|\mathcal{P}\theta - y\|_2^2,$$

where $\alpha = [\alpha_1, \dots, \alpha_n]^T$. The matrix $\mathcal{P}$ in (4.9) is $n \times (n+1)$ and therefore $\mathcal{P}\theta = y$ is an underdetermined linear system. Choosing the solution $\theta = \mathcal{P}^+ y$, a data item z is assigned to class 1 if

$$(4.10) \quad \begin{bmatrix} 1 & \kappa(z, a_1) & \cdots & \kappa(z, a_n) \end{bmatrix} \begin{bmatrix} w_0 \\ \alpha \end{bmatrix} > 0.$$

While the MSE solution in (4.6) is related to FDA when the within-class scatter matrix is nonsingular [6, 7], it can be shown that the MSE solution in (4.6) is related to LDA/GSVD in the case of the singular scatter matrix. The relation between the MSE solution of (4.6) and LDA/GSVD for two-class problems and the corresponding relation between the kernel MSE solution of (4.9) and KDA/GSVD are presented in the appendix.

4.3. GDA. In GDA [1], centering of the data in the feature space is performed by shifting each feature vector by the global centroid

$$\tilde{c} = \frac{1}{n} \sum_{i=1}^n \Phi(a_i),$$

and then the kernel matrix is computed. This kernel matrix

$$\tilde{K} = [k_{i,j}]_{1 \leq i, j \leq n}, \quad \text{where} \quad k_{i,j} = (\Phi(a_i) - \tilde{c})^T (\Phi(a_j) - \tilde{c}),$$

is assumed to be nonsingular. However, $\tilde{K}$ is always singular since

$$\tilde{K} = \begin{bmatrix} (\Phi(a_1) - \tilde{c})^T \\ \vdots \\ (\Phi(a_n) - \tilde{c})^T \end{bmatrix} \begin{bmatrix} \Phi(a_1) - \tilde{c}, & \dots, & \Phi(a_n) - \tilde{c} \end{bmatrix}$$

and

$$\text{rank}([\Phi(a_1) - \tilde{c}, \dots, \Phi(a_n) - \tilde{c}]) \leq n - 1.$$

This makes the theoretical development in [1] break down. In numerical experiments, $\tilde{K}$ may be detected only as being very ill-conditioned since $\tilde{K}$ is symmetric positive semidefinite and zero eigenvalues may appear to be extremely small positive numbers due to rounding errors.

In solving the generalized eigenvalue problem (4.1), the GSVD is applied to $\mathcal{K}_b^T$ and $\mathcal{K}_w^T$ instead of $\mathcal{K}_b \mathcal{K}_b^T$ and $\mathcal{K}_w \mathcal{K}_w^T$, and hence the products in $\mathcal{K}_b \mathcal{K}_b^T$ and $\mathcal{K}_w \mathcal{K}_w^T$ do not need to be computed explicitly. Moreover, the GSVD does not require any parameter optimization such as a regularization parameter. When the regularization method is used for the problem (4.1), the inverse of the matrix $\mathcal{K}_w \mathcal{K}_w^T + dI$ should be computed in addition to the eigenvalue decomposition or singular value decomposition, which is also required in the GSVD approach or MSE solution. In the next section, experimental comparisons of the performances of KDA/GSVD and other kernel-based methods are presented.

5. Experimental results. We demonstrate that our proposed method KDA/GSVD is an effective nonlinear extension of LDA by comparing the performances of KDA/GSVD and other kernel-based nonlinear discriminant analysis algorithms as well as kernel PCA [18].

TABLE 5.1
The description of data sets.

Data set	Musk	Isolet	Car	Mfeature
No. of classes	2	26	4	10
Dim	166	617	6	649
No. of data	6599	7797	1728	2000

For the first experiment, data sets were collected from the UCI machine learning repository.² The detailed description of data sets is shown in Table 5.1. After randomly splitting the data into the equally sized training and test sets, cross-validation

²<http://www.ics.uci.edu/~mllearn/MLRepository.html>.

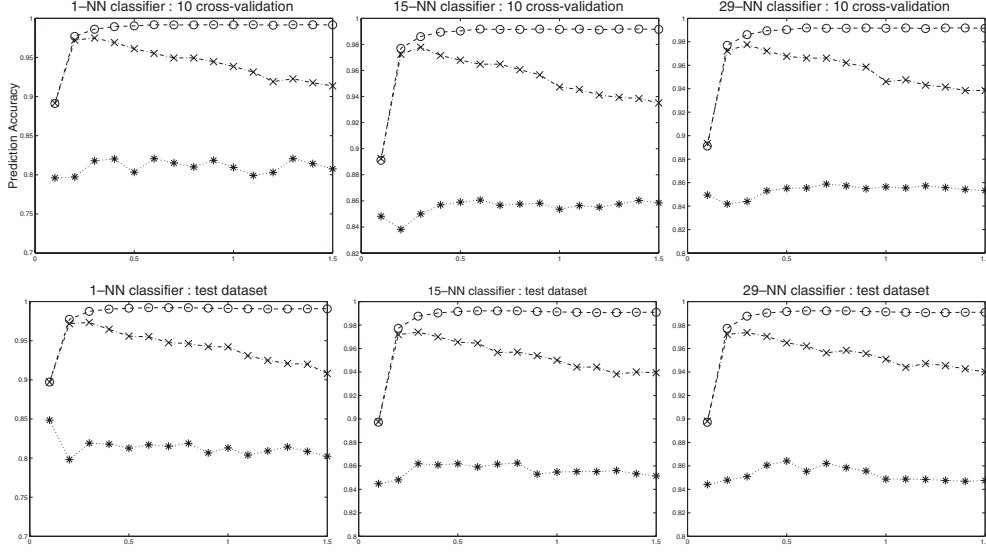


FIG. 5.1. The prediction accuracies in the Musk data (“o”: KDA/GSVD; “x”: KFD; “*”: kernel PCA). The top figures were obtained by the cross-validation with the training data and the bottom figures show the prediction accuracies in the test data. The x-axis corresponds to the values C in $\sigma = C \cdot \text{avg}_d$, where avg_d denotes the average of pairwise distances in the training data set.

is used with the training set in order to determine the optimal value for σ in the Gaussian kernel function:

$$(5.1) \quad \kappa(x, y) = \exp\left(-\frac{\|x - y\|^2}{2\sigma^2}\right).$$

In cross-validation, first, the average of pairwise distances in the training data is computed. Then, based on the average distance avg_d , an optimal value C in

$$(5.2) \quad \sigma = C \cdot \text{avg}_d,$$

which gives the highest prediction accuracy, is searched. In our experiments, we found that $[0.2, 1.0]$ is a reasonable range for C .

For the generalized eigenvalue problem (4.1), the regularization method is applied for the comparison with KDA/GSVD. A regularization parameter d ($d > 0$) is used to make the matrix $\mathcal{K}_w \mathcal{K}_w^T$ nonsingular, and then the eigenvalue problem

$$(\mathcal{K}_w \mathcal{K}_w^T + dI)^{-1} \mathcal{K}_b \mathcal{K}_b^T \alpha = \lambda \alpha$$

is solved. In our experiments, while the regularization parameter d was set as 1, the optimal value σ in the Gaussian kernel function was searched by cross-validation. In kernel PCA, the reduced dimension was one less than the number of classes as in other methods. After dimension reduction, k -nearest neighbor (k -NN) classifiers were used for the k -values of 1, 15, 29.

Figure 5.1 compares the performances of KDA/GSVD, KFD, and kernel PCA for the Musk data which has two classes. The top figures in Figure 5.1 show the prediction accuracies by 10 cross-validations in the training set of the Musk data, where the x-axis corresponds to the values C in (5.2), which ranged from 0.1 to 1.5

TABLE 5.2
The prediction accuracies (%) in the multiclass problems.

Data set	k -NN	LDA	KDA/GSVD	Regularization	Kernel PCA
Musk	$k = 1$	91.0	99.2	97.3	81.7
	15	93.5	99.2	97.4	85.9
	29	93.7	99.2	97.4	86.2
Isolet	1	94.1	97.0	95.8	84.0
	15	94.1	97.0	96.1	85.7
	29	93.9	97.0	96.2	85.0
Car	1	87.5	94.2	95.7	64.9
	15	88.8	94.2	95.1	71.3
	29	87.2	94.2	94.8	72.8
Mfeature ³	1	98.2	98.4	93.6	87.5
	15	97.9	98.4	94.4	85.6
	29	97.8	98.4	94.5	82.8

TABLE 5.3
Prediction accuracy (%) on MEDLINE data.

	Dimension	k -NN			Centroid-based	SVM
		30	45	60		
Original data	22095×1250	83.5	84.0	83.8	84.8	89.5
KDA/GSVD	4×1250	89.4	89.4	89.4	89.4	89.7

with an interval 0.1. In the second row in Figure 5.1, the prediction accuracies for the various C values in the test data are also shown for comparison with those obtained by cross-validation. Table 5.2 shows the results in the test sets using the parameters chosen by cross-validation. It also shows the prediction accuracies obtained by LDA in the original data space. The experimental results demonstrate that the GSVD-based nonlinear discriminant analysis, KDA/GSVD, obtained the competent prediction accuracies over the compared methods, while it does not require any additional parameter optimization, as in the regularization method, and it can naturally handle the multiclass problems.

In the next experiment, the purpose is to evaluate the performance of KDA/GSVD for an undersampled problem. The data set was constructed by randomly selecting 500 documents from each of five categories from the MEDLINE data set. The documents were preprocessed with stemming, stop-list, and rare term removal and encoded using the term frequency and inverse document frequency [11], resulting in a total of 22095 terms. Equally splitting documents in each category into training and test data sets, each of them has 1250 documents. Support vector machine (SVM) classifiers as well as k -NN and centroid-based classification methods were applied both in the original data space and in the reduced dimensional space. Since the SVM classifier is for binary class problems and our data set has 5 classes, we used a DAG scheme for multiclass classification [15]. Table 5.3 shows the prediction accuracies. After both linear and nonlinear SVMs were applied in the original data space, the best accuracy was obtained with a linear soft margin SVM. On the other hand, we obtained a competitive result by a linear SVM in the dimension reduced space by KDA/GSVD using the Gaussian kernel. Since the dimension was reduced dramatically from 22095 down to 4 and was trained with only a linear classifier in the reduced dimension, the training process was much faster than in the full dimension. Even with k -NN

³Due to ill-conditioned scatter matrices, LDA/GSVD instead of LDA was used for Mfeature data.

and centroid-based classification methods, we obtained prediction results that were as good as those obtained with SVM. The high prediction accuracies by k -NN or centroid-based classifiers in the reduced dimensional space by KDA/GSVD show that the difficulty of applying a binary classifier such as SVM to multiclass problems can be overcome effectively.

6. Discussion. We have introduced KDA/GSVD, which is a nonlinear extension of LDA based on kernel functions and the generalized singular value decomposition. One advantage of KDA/GSVD is that it can be applied regardless of singularity of the scatter matrices both in the original space and in the feature space by a nonlinear mapping. It is also shown that in two-class problems KDA/GSVD is related to the kernel version of the MSE solution. Comparison with other methods in solving the generalized eigenvalue problem demonstrates that KDA/GSVD is an effective dimension reduction method for multiclass problems.

Appendix. In this appendix, a relationship between the MSE solution of (4.6) and the LDA/GSVD solution for two-class problems is first presented. It is an extension of the relation between the MSE and FDA [7, 6] to the case of the singular scatter matrices. From the relation between the MSE solution of (4.6) and the LDA/GSVD, the relationship between the kernel MSE solution of (4.9) and the KDA/GSVD solution is also derived.

The problem (4.6) can be solved by the normal equations

$$(A.1) \quad \begin{bmatrix} 1 & \cdots & 1 \\ a_1 & \cdots & a_n \end{bmatrix} \begin{bmatrix} 1 & a_1^T \\ \vdots \\ 1 & a_n^T \end{bmatrix} \begin{bmatrix} w_0 \\ w \end{bmatrix} = \begin{bmatrix} 1 & \cdots & 1 \\ a_1 & \cdots & a_n \end{bmatrix} \begin{bmatrix} \frac{n}{n_1} e_{n_1} \\ -\frac{n}{n_2} e_{n_2} \end{bmatrix},$$

where e_{n_i} is the $n_i \times 1$ column vector with only 1's as its components. From (A.1), we obtain

$$(A.2) \quad \begin{cases} nw_0 + \left(\sum_{1 \leq j \leq n} a_j^T \right) w = 0, \\ \left(\sum_{1 \leq j \leq n} a_j \right) w_0 + \left(\sum_{1 \leq j \leq n} a_j a_j^T \right) w = \frac{n}{n_1} \sum_{j \in N_1} a_j - \frac{n}{n_2} \sum_{j \in N_2} a_j. \end{cases}$$

The first equation in (A.2) gives

$$w_0 = -c^T w,$$

and by substituting it in the second equation and using the alternative expressions⁴ of S_b and S_w

$$(A.3) \quad S_b = \sum_{1 \leq i \leq 2} n_i c_i c_i^T - n c c^T \quad \text{and} \quad S_w = \sum_{1 \leq j \leq n} a_j a_j^T - \sum_{1 \leq i \leq 2} n_i c_i c_i^T,$$

we obtain

$$(A.4) \quad (S_b + S_w)w = n(c_1 - c_2).$$

⁴In fact, S_b can also be represented as $S_b = \frac{n_1 n_2}{n} (c_1 - c_2)(c_1 - c_2)^T$, which will be used in (A.5) later.

On the other hand, since $\text{rank}(S_b) = 1$ in the two-class case, we have

$$\eta_1 > \eta_2 = \cdots = \eta_m = 0 \quad \text{in} \quad \zeta_i S_b x_i = \eta_i S_w x_i$$

by LDA/GSVD. Since $\eta_1 + \zeta_1 = 1$ and $\eta_1 S_w x_1 = \zeta_1 S_b x_1$,

$$(A.5) \quad \eta_1 (S_b + S_w) x_1 = (\eta_1 + \zeta_1) S_b x_1 = S_b x_1 = \frac{n_1 n_2}{n} (c_1 - c_2)(c_1 - c_2)^T x_1.$$

Denoting

$$(A.6) \quad \rho = \eta_1 \frac{n^2}{n_1 n_2 (c_1 - c_2)^T x_1},$$

equation (A.5) becomes

$$(A.7) \quad (S_b + S_w) \rho x_1 = n(c_1 - c_2).$$

Hence from (A.4) and (A.7), we have

$$(A.8) \quad S_t w = (S_b + S_w) w = (S_b + S_w) \rho x_1 = S_t \rho x_1,$$

where

$$S_t = \sum_{i=1}^n (a_i - c)(a_i - c)^T = S_b + S_w$$

is the total scatter matrix.

Let the symmetric eigenvalue decomposition of S_t be

$$S_t = Y \Sigma_t Y^T = \underbrace{[Y_1]}_s \underbrace{[Y_2]}_{m-s} \begin{bmatrix} \Sigma_1 & 0 \\ 0 & 0 \end{bmatrix} \begin{bmatrix} Y_1^T \\ Y_2^T \end{bmatrix},$$

where $s = \text{rank}(S_t)$, Y is orthogonal, and Σ_1 is a diagonal matrix with nonincreasing positive diagonal elements. Then from (A.8),

$$Y_1 \Sigma_1 Y_1^T w = Y_1 \Sigma_1 Y_1^T \rho x_1 \quad \text{and} \quad Y_1^T w = Y_1^T \rho x_1.$$

PROPOSITION A.1. *Let $x \in \text{null}(S_b) \cap \text{null}(S_w)$. Then $x^T S_b x = x^T S_w x = 0$. For any $x \in \text{null}(S_b) \cap \text{null}(S_w)$, $x^T S_b x = x^T S_w x = 0$. Hence*

$$(A.9) \quad 0 = x^T S_b x = (x^T H_b)(H_b^T x) = \|x^T H_b\|^2 = \sum_{i=1}^r n_i |x^T c_i - x^T c|^2 \quad \text{and}$$

$$(A.10) \quad 0 = x^T S_w x = \sum_{j=1}^n |x^T a_j - x^T c_i|^2, \quad \text{where } a_j \text{ belongs to class } i.$$

Equations (A.9) and (A.10) give

$$\begin{cases} x^T c_i = x^T c & \text{for } i = 1, \dots, r, \\ x^T a_j = x^T c_i & \text{for all } j \text{ in } N_i \text{ and } i = 1, \dots, r, \end{cases}$$

and these imply that all data items are transformed to one point by x^T . $\square$

Since

$$\text{span}\{Y_2\} = \text{null}(S_t) \subset \text{null}(S_b) \cap \text{null}(S_w),$$

by Proposition A.1, $Y_2^T(z - c) = 0$, and we obtain

$$\begin{aligned} w^T z + w_0 &= w^T(z - c) = w^T(Y_1 Y_1^T + Y_2 Y_2^T)(z - c) \\ (A.11) \quad &= w^T Y_1 Y_1^T(z - c) = \rho x_1^T Y_1 Y_1^T(z - c) \\ &= \rho x_1^T(Y_1 Y_1^T + Y_2 Y_2^T)(z - c) = \rho x_1^T(z - c). \end{aligned}$$

Equation (A.11) gives the relation between the MSE solution and the solution of LDA/GSVD for two-class problems, which holds regardless of the singularity of the scatter matrices.

Now it should be straightforward to derive the corresponding relation between the kernel MSE solution and KDA/GSVD in the two-class case. The formulation (4.9) for the kernel MSE solution is obtained by substituting the original data $[a_1, \dots, a_n]$ in (4.6) with the kernel matrix $K = [\kappa(a_i, a_j)]_{(1 \leq i \leq n, 1 \leq j \leq n)}$, where each column of the kernel matrix K can be considered as a data item. On the other hand, as illustrated in Figure 3.1, KDA/GSVD solves the generalized eigenvalue problem

$$\mathcal{K}_b \mathcal{K}_b^T \alpha = \lambda \mathcal{K}_w \mathcal{K}_w^T \alpha.$$

$\mathcal{K}_b \mathcal{K}_b^T$ and $\mathcal{K}_w \mathcal{K}_w^T$ are the scatter matrices of the kernel matrix K when each column of the kernel matrix K is considered as a data item. Hence the relation between the kernel MSE and KDA/GSVD corresponding to (A.11) can be derived by substituting the original data a_i with $[\kappa(a_i, a_1), \dots, \kappa(a_i, a_n)]^T$ in the proof of the relation between the MSE and LDA/GSVD.

REFERENCES

- [1] G. BAUDAT AND F. ANOUAR, *Generalized discriminant analysis using a kernel approach*, Neural Comput., 12 (2000), pp. 2385–2404.
- [2] S. BILLINGS AND K. LEE, *Nonlinear Fisher discriminant analysis using a minimum squared error cost function and the orthogonal least squares algorithm*, Neural Networks, 15 (2002), pp. 263–270.
- [3] C. BURGESS, *A tutorial on support vector machines for pattern recognition*, Data Min. Knowl. Discov., 2 (1998), pp. 121–167.
- [4] L. CHEN, H. LIAO, M. KO, J. LIN, AND G. YU, *A new LDA-based face recognition system which can solve the small sample size problem*, Pattern Recognition, 33 (2000), pp. 1713–1726.
- [5] N. CRISTIANINI AND J. SHAWE-TAYLOR, *An Introduction to Support Vector Machines and Other Kernel-Based Learning Methods*, Cambridge University Press, Cambridge, UK, 2000.
- [6] R. DUDA AND P. HART, *Pattern Classification and Scene Analysis*, Wiley-Interscience, New York, 1973.
- [7] R. DUDA, P. HART, AND D. STORK, *Pattern Classification*, Wiley-Interscience, New York, 2001.
- [8] J. FRIEDMAN, *Regularized discriminant analysis*, J. Amer. Statist. Assoc., 84 (1989), pp. 165–175.
- [9] K. FUKUNAGA, *Introduction to Statistical Pattern Recognition*, 2nd ed., Academic Press, New York, 1990.
- [10] P. HOWLAND, M. JEON, AND H. PARK, *Structure preserving dimension reduction for clustered text data based on the generalized singular value decomposition*, SIAM J. Matrix Anal. Appl., 25 (2003), pp. 165–179.
- [11] G. KOWALSKI, *Information Retrieval System: Theory and Implementation*, Kluwer Academic, Dordrecht, The Netherlands, 1997.

- [12] S. MIKA, G. RÄTSCH, J. WESTON, B. SCHÖLKOPF, AND K.-R. MÜLLER, *Fisher discriminant analysis with kernels*, in Neural Networks for Signal Processing IX, Y. H. Hu, J. Larsen, E. Wilson, and S. Douglas, eds., IEEE, Piscataway, NJ, 1999, pp. 41–48.
- [13] S. MIKA, G. RÄTSCH, J. WESTON, B. SCHÖLKOPF, A. SMOLA, AND K.-R. MÜLLER, *Invariant feature extraction and classification in kernel spaces*, Adv. Neural Inf. Process. Syst., 12 (2000), pp. 526–532.
- [14] C. H. PARK AND H. PARK, *A Comparison of Generalized LDA Algorithms for Undersampled Problems*, Tech. rep. 03-048, Department of Computer Science and Engineering, University of Minnesota, Minneapolis, MN, 2003.
- [15] J. PLATT, N. CRISTIANINI, AND J. SHAWE-TAYLOR, *Large margin dags for multiclass classification*, Adv. Neural Inf. Process. Syst., 12 (2000), pp. 547–553.
- [16] V. ROTH AND V. STEINHAGE, *Nonlinear discriminant analysis using kernel functions*, Adv. Neural Inf. Process. Syst., 12 (2000), pp. 568–574.
- [17] B. SCHÖLKOPF, S. MIKA, C. BURGESS, P. KNIRSCH, K.-R. MÜLLER, G. RÄTSCH, AND A. SMOLA, *Input space versus feature space in kernel-based methods*, IEEE Trans. Neural Networks, 10 (1999), pp. 1000–1017.
- [18] B. SCHÖLKOPF, A. SMOLA, AND K.-R. MÜLLER, *Nonlinear component analysis as a kernel eigenvalue problem*, Neural Comput., 10 (1998), pp. 1299–1319.
- [19] H. YU AND J. YANG, *A direct LDA algorithm for high-dimensional data with application to face recognition*, Pattern Recognition, 34 (2001), pp. 2067–2070.

GENERALIZED LYAPUNOV EQUATIONS AND POSITIVE DEFINITE FUNCTIONS*

RAJENDRA BHATIA[†] AND DRISS DRISSI[‡]

Abstract. We establish the positive definiteness of some functions and of some matrices that arise as solutions of generalized Lyapunov equations.

Key words. positive definite matrix, positive definite function, Fourier transform, Bochner's theorem, Lyapunov matrix equation, operator means

AMS subject classifications. 42A82, 47A62, 15A24, 15A48

DOI. 10.1137/040608970

1. Introduction. Let A be a positive definite matrix and consider the following matrix equations:

$$(L1) \quad AX + XA = B.$$

$$(L2) \quad A^2X + XA^2 + tAXA = B.$$

$$(L3) \quad A^3X + XA^3 + t(A^2XA + AXA^2) = B.$$

$$(L4) \quad A^4X + XA^4 + t(A^3XA + AXA^3) + 6A^2XA^2 = B.$$

$$(L5) \quad A^4X + XA^4 + 4(A^3XA + AXA^3) + tA^2XA^2 = B.$$

Equation (L1) is known as the *Lyapunov equation* and has been studied extensively.

We may choose an orthonormal basis for the underlying space in which A is diagonal, $A = \text{diag}(\lambda_1, \dots, \lambda_n)$. Then, in component form, (L1) may be written as

$$(\lambda_i + \lambda_j)x_{ij} = b_{ij}$$

and solved as

$$(1) \quad x_{ij} = \frac{b_{ij}}{\lambda_i + \lambda_j}.$$

The matrix C with entries

$$(2) \quad c_{ij} = \frac{1}{\lambda_i + \lambda_j}$$

is called the *Cauchy matrix*. It is positive semidefinite. One way of seeing this is by writing

$$(3) \quad \frac{1}{\lambda_i + \lambda_j} = \int_0^\infty e^{-t(\lambda_i + \lambda_j)} dt$$

*Received by the editors May 25, 2004; accepted for publication (in revised form) by V. Mehrmann October 11, 2004; published electronically July 13, 2005.

<http://www.siam.org/journals/simax/27-1/60897.html>

[†]Indian Statistical Institute, New Delhi 110 016, India (rbh@isid.ac.in).

[‡]Department of Mathematics, Kuwait University, Safat, Kuwait (drissi@mcs.sci.kuniv.edu.kw). The work of this author was supported by Kuwait University research grant SM 02/04.

and thus realizing C as the Gram matrix associated with the vectors $\{e^{-t\lambda_i}\}$ in the Hilbert space $L_2(0, \infty)$. The solution (1) can be written in matrix form as

$$X = C \circ B,$$

where $\circ$ denotes the entrywise product $(\dots, \dots, \dots)$ of two matrices. So, if B is positive semidefinite, then X is also positive semidefinite. This is an important fact in Lyapunov's theory of stability of dynamical systems [13]. Some generalizations of Lyapunov's equation where the solution is positive semidefinite, and their importance, are discussed in [5, Chap. 10].

The analogous question about (L2) was considered, and partly answered, by Kwong [12]. We want to know for what real numbers t does equation (L2) have a positive semidefinite solution X whenever B is positive semidefinite. Our arguments above show that this is the same as asking when the matrix Z with entries

$$(4) \quad z_{ij} = \frac{1}{\lambda_i^2 + \lambda_j^2 + t\lambda_i\lambda_j}$$

is positive semidefinite (for any given positive numbers $\lambda_1, \dots, \lambda_n$). Clearly the restriction $t > -2$ is necessary. By a somewhat intricate topological argument Kwong [12] showed that the condition $-2 < t \leq 2$ is sufficient to guarantee that the matrix (4) is positive semidefinite for all orders n . (A simple proof of this statement is given in section 2 below.) He showed also that when $n = 2$, the matrix (4) is positive semidefinite for all $t > -2$, but further restrictions on t are needed for higher dimensions. The problem was solved completely in a paper by Bhatia and Parthasarathy [4, Thm. 5.1]. We have the following.

THEOREM 1.1 (Kwong [12], Bhatia and Parthasarathy [4]). *Let Z be the $n \times n$ matrix with entries z_{ij} as in (4) for $\lambda_1, \dots, \lambda_n$ positive numbers. Then Z is positive semidefinite if and only if $-2 < t \leq 2$.*

The method of Bhatia and Parthasarathy [4] consists of showing that matrices such as (4) are congruent to others related to difference kernels, and then checking whether these kernels are positive definite. In the process, this leads to interesting examples of positive definite functions which may be useful in other contexts. See, in particular, the monograph of Hiai and Kosaki [9], which shows numerous applications to operator theory discovered by these authors in a series of papers [7, 8, 11]. Our aim in this paper is to apply these ideas to (L3)–(L5). We prove the following three theorems.

THEOREM 1.2. *Let $t > -1$, and let Y be the $n \times n$ matrix with entries y_{ij} as in (5) for $\lambda_1, \dots, \lambda_n$ positive numbers. Then Y is positive semidefinite if and only if $t \geq 4$.*

$$(5) \quad y_{ij} = \frac{1}{\lambda_i^3 + \lambda_j^3 + t(\lambda_i^2\lambda_j + \lambda_i\lambda_j^2)}$$

THEOREM 1.3. *Let $t \geq 4$, and let W be the $n \times n$ matrix with entries w_{ij} as in (6) for $\lambda_1, \dots, \lambda_n$ positive numbers. Then W is positive semidefinite if and only if $t > -4$.*

$$(6) \quad w_{ij} = \frac{1}{\lambda_i^4 + \lambda_j^4 + t(\lambda_i^3\lambda_j + \lambda_i\lambda_j^3) + 6\lambda_i^2\lambda_j^2}$$

THEOREM 1.4. Let $-10 < t \leq 6$, $\lambda_1, \dots, \lambda_n$ be real numbers, $n \geq 1$, and V be an $n \times n$ matrix with entries v_{ij} defined by

$$(7) \quad v_{ij} = \frac{1}{\lambda_i^4 + \lambda_j^4 + 4(\lambda_i^3 \lambda_j + \lambda_i \lambda_j^3) + t \lambda_i^2 \lambda_j^2}$$

then V is positive semidefinite for $t > -10$, $t > -4$, and $t \leq 6$, respectively.

Note the differences in the nature of restrictions on t in the three theorems. The restrictions $t > -1$, $t > -4$, and $t > -10$, respectively, are necessary even in the case $n = 1$. For $n \geq 2$, in the absence of these restrictions the matrices in question have nonpositive diagonal entries and cannot be positive semidefinite.

Some parts of these statements can be proved by rather simple arguments. These are shown in section 2. Complete proofs need more elaborate arguments given in section 4. This separation brings out more interesting differences among the three results.

For brevity we use the term *positive semidefinite matrix* to mean a positive semidefinite matrix in the rest of the paper.

2. The easier parts. Schur's theorem says that the Schur product $A \circ B$ of two positive matrices is positive. Two matrices X and Y are said to be *congruent* if $Y = T^* X T$ for some nonsingular matrix T . If X is positive, then so is every matrix congruent to it.

Let z_{ij} be the numbers defined by (4). For all real numbers t we have

$$z_{ij} = \frac{1}{(\lambda_i + \lambda_j)^2} - \frac{1}{1 - \frac{(2-t)\lambda_i \lambda_j}{(\lambda_i + \lambda_j)^2}}.$$

Since $4\lambda_i \lambda_j / (\lambda_i + \lambda_j)^2 \leq 1$, we can expand this as an infinite series

$$(8) \quad z_{ij} = \frac{1}{(\lambda_i + \lambda_j)^2} \sum_{n=0}^{\infty} (2-t)^n \frac{\lambda_i^n \lambda_j^n}{(\lambda_i + \lambda_j)^{2n}},$$

which is convergent, provided $|2-t| < 4$. The matrix with entries $1/(\lambda_i + \lambda_j)^2$ is positive, being the Schur product of the Cauchy matrix with itself. The matrix with entries $\lambda_i \lambda_j / (\lambda_i + \lambda_j)^2$ is congruent to this matrix (via a congruence by the diagonal matrix $T = \text{diag}(\lambda_1, \dots, \lambda_n)$). Each n -fold Schur product of this matrix with itself is again positive. Hence, if $t < 2$, the matrix Z is positive, being a sum of several positive matrices. This proves the sufficiency of the condition in Theorem 1.1, the part first established by Kwong [12]. It is interesting, and perhaps surprising, that the sufficient condition $-2 < t \leq 2$ turns out to be necessary as well.

In the same way we can write the numbers y_{ij} defined by (5) as

$$y_{ij} = \frac{1}{(\lambda_i + \lambda_j)^3} - \frac{1}{1 - \frac{(3-t)(\lambda_i^2 \lambda_j + \lambda_i \lambda_j^2)}{(\lambda_i + \lambda_j)^3}}.$$

Since $4(\lambda_i^2 \lambda_j + \lambda_i \lambda_j^2) / (\lambda_i + \lambda_j)^3 \leq 1$ this can be expanded as an infinite series

$$(9) \quad y_{ij} = \frac{1}{(\lambda_i + \lambda_j)^3} \sum_{n=0}^{\infty} (3-t)^n \left(\frac{\lambda_i^2 \lambda_j + \lambda_i \lambda_j^2}{(\lambda_i + \lambda_j)^3} \right)^n,$$

which is convergent, provided $|3 - t| < 4$, i.e., provided $-1 < t < 7$. Note that

$$\frac{\lambda_i^2 \lambda_j + \lambda_i \lambda_j^2}{(\lambda_i + \lambda_j)^3} = \frac{\lambda_i \lambda_j}{(\lambda_i + \lambda_j)^2},$$

and we have observed that the matrix with these numbers as its ij entries is positive. By the same arguments as we used for the matrix Z we can conclude that the matrix Y is positive for $-1 < t \leq 3$. This is where the similarity with Z ends. The matrices Y turn out to be positive for $t > 3$ as well.

More interesting things happen when we apply these considerations to the matrix W defined by (6). Since

$$(10) \quad 8(\lambda_i^3 \lambda_j + \lambda_i \lambda_j^3) \leq (\lambda_i + \lambda_j)^4,$$

we have

$$(11) \quad w_{ij} = \frac{1}{(\lambda_i + \lambda_j)^4} \sum_{n=0}^{\infty} (4 - t)^n \left(\frac{\lambda_i^3 \lambda_j + \lambda_i \lambda_j^3}{(\lambda_i + \lambda_j)^4} \right)^n,$$

provided $|4 - t| < 8$, i.e., provided $-4 < t < 12$. To proceed further we need to know whether the matrix G with entries

$$(12) \quad g_{ij} = \frac{\lambda_i^3 \lambda_j + \lambda_i \lambda_j^3}{(\lambda_i + \lambda_j)^4}$$

is positive. From (10) it follows that any 2×2 matrix G with entries given by (12) is positive. Thus from the identity (11) we can conclude that for $-4 < t \leq 4$ the matrix W is positive in the special case $n = 2$. We will see in section 4 that the matrix G is not always positive in general for all n . So this argument does not carry us further. We have to rely on other arguments to decide whether W is positive in all dimensions. It turns out that this is always the case for $t < 4$ but is always the case for $t \geq 4$. In other words the matrix W is positive for $t \geq 4$, and for arbitrary n this restriction is necessary. For $n = 2$, the matrix W is positive for all $-4 < t \leq 4$ as well.

Finally let us consider the matrix V defined by (7). Since $16\lambda_i^2 \lambda_j^2 \leq (\lambda_i + \lambda_j)^4$ we can write

$$(13) \quad v_{ij} = \frac{1}{(\lambda_i + \lambda_j)^4} \sum_{n=0}^{\infty} (6 - t)^n \left(\frac{\lambda_i^2 \lambda_j^2}{(\lambda_i + \lambda_j)^4} \right)^n,$$

provided $|6 - t| < 16$, i.e., provided $-10 < t < 22$. Our arguments using the Cauchy matrix, congruence, and Schur products allow us to deduce from this series expansion that V is positive if $-10 < t \leq 6$. (This fact has been noted by Kwong [12] as well.) It turns out that in this case again the condition $-10 < t \leq 6$ is necessary to ensure the positivity of all matrices V for all orders n .

3. Some Fourier transforms. Fourier transforms of some functions are needed for our arguments. First we indicate how they are calculated using contour integrations. See [14, p. 116] for the theory behind the method that we use. For a function $f \in L_1(\mathbb{R})$ we use the notation $\hat{f}$ for the function

$$(14) \quad \hat{f}(\xi) = \int_{-\infty}^{\infty} f(x) e^{i\xi x} dx.$$

If f is an even function, then

$$(15) \quad \hat{f}(\xi) = \int_{-\infty}^{\infty} f(x) \cos \xi x \, dx.$$

Let f_1 be the function

$$(16) \quad f_1(x) = \frac{1}{\cosh x + t}, \quad -1 < t < 1.$$

The integral (15) corresponding to this can be calculated by the method of residues. Let

$$(17) \quad \varphi(z) = \frac{e^{i\xi z}}{\cosh z + t}.$$

If $z = x + iy$, then $\cosh z = \cosh x \cos y + i \sinh x \sin y$. So the poles of φ are at the points $z = i \arccos(-t) = i(\pi \pm \arccos t)$. Choose the rectangular contour $[-R, R, R + 2\pi i, -R + 2\pi i]$, where $R > 0$. There are two poles of φ inside this rectangle: $z_1 = i(\pi + \arccos t)$, $z_2 = i(\pi - \arccos t)$. Using the periodicity of the exponential function we get by the method of residues

$$\int_{-\infty}^{\infty} \frac{e^{i\xi x}}{\cosh x + t} dx = \frac{2\pi i}{1 - e^{-\xi 2\pi}} \left[\frac{e^{i\xi z_1}}{\sinh z_1} + \frac{e^{i\xi z_2}}{\sinh z_2} \right].$$

Substitute the values of z_1 and z_2 . Use the relations $\sinh iz = i \sin z$ and $\sin(\pi \pm \arccos t) = \mp \sqrt{1 - t^2}$. A little calculation leads to the result that

$$(18) \quad \hat{f}_1(\xi) = \frac{2\pi \sinh(\xi \arccos t)}{\sqrt{1 - t^2} \sinh \xi \pi}$$

is the Fourier transform of the function (16).

Now consider the function

$$(19) \quad f_2(x) = \frac{1}{\cosh x + t}, \quad t > 1.$$

In this case the function φ defined as in (17) has poles at $z = \pm \operatorname{arccosh} t + ik\pi$, where k is an odd integer. For large R , the rectangular contour we chose above has two of these poles inside it. These are $z_1 = \operatorname{arccosh} t + i\pi$ and $z_2 = -\operatorname{arccosh} t + i\pi$. Again by the method of residues,

$$\int_{-\infty}^{\infty} \frac{e^{i\xi x}}{\cosh x + t} dx = \frac{2\pi i}{1 - e^{-\xi 2\pi}} \left[\frac{e^{i\xi z_1}}{\sinh z_1} + \frac{e^{i\xi z_2}}{\sinh z_2} \right].$$

Substitute the values of z_1 and z_2 . For $t \geq 1$ we have $\operatorname{arccosh} t = \log(t + \sqrt{t^2 - 1})$ and $\sinh(\operatorname{arccosh} t) = \sqrt{t^2 - 1}$. A little calculation shows that the Fourier transform of the function f_2 in (19) is

$$(20) \quad \hat{f}_2(\xi) = \frac{2\pi \sin(\xi \operatorname{arccosh} t)}{\sqrt{t^2 - 1} \sinh \xi \pi}.$$

Now consider the function

$$(21) \quad f_3(x) = \frac{1}{\cosh x + \sigma},$$

where σ is a complex number. For σ varying in the plane slit along $(-\infty, 1]$, f and its Fourier transform are analytic in σ . Hence we have

$$(22) \quad \hat{f}_3(\xi) = \begin{cases} \frac{2\pi \sinh(\xi \arccos \sigma)}{\sqrt{1-\sigma^2} \sinh \xi \pi} & \text{for } |\sigma| < 1, \\ \frac{2\pi \sin(\xi \operatorname{arccosh} \sigma)}{\sqrt{\sigma^2-1} \sinh \xi \pi} & \text{for } |\sigma| > 1, \sigma \notin (-\infty, 1]. \end{cases}$$

If $\sigma \notin \mathbb{R}$, both formulas make sense and give the same value for $\hat{f}(\xi)$. The special case when σ is purely imaginary will be needed in our calculation. Let

$$(23) \quad f_4(x) = \frac{1}{\cosh x + ia}, \quad a \in \mathbb{R}.$$

Use the formula (22) and the relation $\arccos(ia) = \frac{\pi}{2} - i \operatorname{arcsinh} a$. A little calculation shows

$$(24) \quad \hat{f}_4(\xi) = \frac{\pi}{\sqrt{1+a^2}} \left[\frac{\cos(\xi \operatorname{arcsinh} a)}{\cosh \frac{\xi \pi}{2}} - i \frac{\sin(\xi \operatorname{arcsinh} a)}{\sinh \frac{\xi \pi}{2}} \right].$$

(When $\sigma = 1$, we can obtain an expression for $\hat{f}_3(\xi)$ by taking limits in either of the formulas (22). We get $\hat{f}(\xi) = 2\pi\xi/\sinh \xi\pi$ in this case.)

4. Proofs of Theorems 1.2–1.4. Following Bhatia and Parthasarathy [4] we convert the question of positivity of the matrices under consideration to that of positive definiteness of some functions on the real line.

Put $\lambda_i = e^{x_i}$, $x_i \in \mathbb{R}$. Then (5) is equal to

$$\begin{aligned} y_{ij} &= \frac{1}{e^{3x_i} + e^{3x_j} + t(e^{2x_i+x_j} + e^{x_i+2x_j})} \\ &= \frac{1}{e^{3x_i/2}} \left(\frac{1}{e^{3(x_i-x_j)/2} + e^{3(x_j-x_i)/2} + t(e^{(x_i-x_j)/2} + e^{(x_j-x_i)/2})} \right) \frac{1}{e^{3x_j/2}}. \end{aligned}$$

Thus the matrix Y is congruent to one with entries

$$\frac{1}{\cosh\left(\frac{3(x_i-x_j)}{2}\right) + t \cosh\left(\frac{x_i-x_j}{2}\right)}.$$

To say that all such matrices are positive for $t > -1$ is to say that the function

$$f(x) = \frac{1}{\cosh 3x + t \cosh x}$$

on the real line is positive definite for $t > -1$. Our argument in section 2 shows that this is certainly the case for $-1 < t \leq 3$. Using the identity $\cosh 3x = 4 \cosh^3 x - 3 \cosh x$, we have

$$f(x) = \frac{1}{4 \cosh^3 x + (t-3) \cosh x}.$$

The positive definiteness of this function for $t > 3$ is equivalent to the following assertion.

PROPOSITION 4.1. *Let a be a real number.*

$$(25) \quad f(x) = \frac{1}{\cosh^3 x + a^2 \cosh x}$$

By a theorem of Bochner, f is positive definite if and only if its Fourier transform $\hat{f}(\xi) \geq 0$. The function f in (25) can be expressed also as

$$f(x) = \frac{1}{a^2} \left[\frac{1}{\cosh x} - \operatorname{Re} \frac{1}{\cosh x + ia} \right].$$

The Fourier transform of this can be found using (24); it is

$$\hat{f}(\xi) = \frac{\pi}{a^2 \cosh \frac{\xi\pi}{2}} \left[1 - \frac{\cos(\xi \operatorname{arcsinh} a)}{\sqrt{1+a^2}} \right].$$

It is clear that $\hat{f}(\xi) \geq 0$. $\square$

As explained above, this proves Theorem 1.2.

Using the same argument, we write (6) as

$$w_{ij} = \frac{1}{e^{2x_i}} \left(\frac{1}{e^{2(x_i-x_j)} + e^{2(x_j-x_i)} + t(e^{x_i-x_j} + e^{x_j-x_i}) + 6} \right) \frac{1}{e^{2x_j}}.$$

The assertion of Theorem 1.3 is equivalent to the statement that the function

$$f(x) = \frac{1}{\cosh 2x + t \cosh x + 3}$$

is positive definite for $t \geq 4$ but not for $-4 < t < 4$. Using the identity $\cosh 2x = 2 \cosh^2 x - 1$, we see that this is equivalent to the following.

PROPOSITION 4.2. *Let t be a real number.*

$$(26) \quad f(x) = \frac{1}{\cosh^2 x + t \cosh x + 1}$$

Let $t \geq 2$ or $t \leq -2$.

Let $t > 2$. Let α, β be the roots of the polynomial $x^2 + tx + 1$. Then α, β are negative real numbers and their product is 1. We can write the function f in (26) as

$$(27) \quad f(x) = \frac{1}{\sqrt{t^2 - 4}} \left[\frac{1}{\cosh x + a} - \frac{1}{\cosh x + b} \right],$$

where $0 < a < 1$ and $b = \frac{1}{a} > 1$. (The numbers a, b are the negatives of α and β .) The Fourier transform of the expression in the square brackets in (27) is calculated using (22). This is equal to

$$(28) \quad \begin{aligned} & \frac{2\pi}{\sinh \xi\pi} \left[\frac{\sinh(\xi \arccos a)}{\sqrt{1-a^2}} - \frac{\sinh(\xi \operatorname{arccosh} b)}{\sqrt{b^2-1}} \right] \\ &= \frac{2\pi}{\sinh \xi\pi} \frac{1}{\sqrt{1-a^2}} \left[\sinh(\xi \arccos a) - a \sin \left(\xi \operatorname{arccosh} \frac{1}{a} \right) \right]. \end{aligned}$$

We want to show that this is nonnegative for all ξ . As a function of ξ this expression is even. So it suffices to show that for $\xi \geq 0$ the expression inside the square brackets on the right-hand side of (28) is nonnegative.

The function $g(a) = \arccos a - a \operatorname{arccosh} \frac{1}{a}$ has the following properties: $g(0) = \pi/2$, $g(1) = 0$, and $g'(a) = -\operatorname{arccosh} \frac{1}{a} < 0$. Hence $g(a) > 0$ for $0 < a < 1$. So for all $\xi \geq 0$ we have

$$\sinh(\xi \arccos a) \geq \xi \arccos a \geq a \xi \operatorname{arccosh} \frac{1}{a} \geq a \sin \left(\xi \operatorname{arccosh} \frac{1}{a} \right).$$

This proves that the expression (28) is nonnegative. So the function (26) is positive definite for all $t > 2$. By continuity this is true for $t = 2$ as well.

When $t = 0$, the function f in (26) reduces to

$$f(x) = \frac{1}{\cosh^2 x + 1} = -\operatorname{Im} \frac{1}{\cosh x + i}.$$

The Fourier transform of this function can be read off from (24). It is

$$\hat{f}(\xi) = \frac{\pi \sin(\xi \operatorname{arcsinh} 1)}{\sqrt{2} \sinh \frac{\xi \pi}{2}}.$$

This function is negative for some ξ , and hence f is not positive definite in this case.

More elaborate arguments are needed to decide the case $-2 < t < 2$, $t \neq 0$. In this case let

$$\alpha = \frac{-t + i\sqrt{4-t^2}}{2}.$$

Then α and $\beta = \bar{\alpha}$ are the two roots of the polynomial $x^2 + tx + 1$. They are complex numbers of modulus 1. We have

$$(29) \quad \alpha = e^{i\theta} = \cos \theta + i \sin \theta, \quad 0 < \theta < \pi.$$

The case $\theta = \pi/2$ corresponds to $t = 0$, which we have already settled. The function f in (26) is equal to

$$f(x) = \frac{2}{\sqrt{4-t^2}} \operatorname{Im} \frac{1}{\cosh x - \alpha}.$$

We wish to prove that if

$$g(x) = \frac{1}{\cosh x - \alpha},$$

where α is as in (29), then the imaginary part of $\hat{g}(\xi)$ is negative for some ξ .

Use the formula (22) valid by an analytic continuation for all points σ on the unit circle except -1 . Our problem reduces to showing that for every point $\alpha \neq \pm 1$ on the unit circle, the expression

$$(30) \quad \operatorname{Im} \frac{\sinh(\xi \arccos \alpha)}{\sqrt{1-\alpha^2}}$$

is negative for some values of $\xi > 0$.

In the next few lines we write the principal values of the functions involved. A small calculation shows that for α as in (29),

$$(31) \quad \sqrt{1-\alpha^2} = \sqrt{\sin \theta + \sin^2 \theta} - i \operatorname{sgn}(\sin 2\theta) \sqrt{\sin \theta - \sin^2 \theta}$$

and

$$(32) \quad \sinh(\xi \arccos \alpha) = -i \sin \left(\xi \log(\alpha + i\sqrt{1-\alpha^2}) \right).$$

Let $\gamma = \alpha + i\sqrt{1-\alpha^2}$. A calculation shows

$$\gamma = \cos \theta + \operatorname{sgn}(\sin 2\theta) \sqrt{\sin \theta - \sin^2 \theta} + i(\sin \theta + \sqrt{\sin \theta + \sin^2 \theta}).$$

The conditions on θ guarantee that $\eta = \log \gamma$ is not a real number. The quantity (30) is equal to

$$(33) \quad \operatorname{Im} \frac{-i \sin(\xi \eta)}{\sqrt{1-\alpha^2}} = -\operatorname{Re} \frac{\sin(\xi \eta)}{\sqrt{1-\alpha^2}}.$$

Mapping properties of the function $\sin$ from the z -plane into the w -plane are well known. This function maps the strip $|\operatorname{Re} z| < \pi/2$ bijectively onto the w -plane slit along a part of the real line $(-\infty, -1] \cup [1, \infty)$. Horizontal lines in the z -plane are mapped onto ellipses with major axes contained in the u -axis and vertical lines are mapped onto hyperbolas with vertices on the u -axis in the w -plane. From these properties it is easy to see that the quantity (33) is negative for some values of ξ .

This completes the proof of Proposition 4.2, and that, in turn, proves Theorem 1.3. $\square$

These arguments can be repeated mutatis mutandis to prove Theorem 1.4. A little calculation shows that the statement of Theorem 1.4 is equivalent to the assertion that the function

$$\frac{1}{2 \cosh 2x + 8 \cosh x + t}$$

is positive definite if and only if $-10 < t \leq 6$. This is the same as saying that the function

$$f(x) = \frac{1}{\cosh 2x + 4 \cosh x + t}$$

is positive definite if and only if $-5 < t \leq 3$. Using the identity $\cosh 2x = 2 \cosh^2 x - 1$, we see that this is equivalent to the following assertion.

PROPOSITION 4.3. *Let t be a real number. The function*

$$(34) \quad f(x) = \frac{1}{\cosh^2 x + 2 \cosh x + t}$$

is positive definite if and only if $-3 < t \leq 1$.

The arguments in section 2 show that the function (34) is positive definite if $-3 < t \leq 1$. Let $t > 1$ and let

$$\alpha = -1 + i\sqrt{t-1}.$$

Then α and $\beta = \bar{\alpha}$ are the two roots of the polynomial $x^2 + 2x + t$. We have $|\alpha|^2 = t > 1$. As in the proof of Proposition 4.2 our problem reduces to showing that when α is a complex number with $|\alpha| > 1$, and α is not real, then the imaginary part of the Fourier transform of the function

$$f(x) = \frac{1}{\cosh x - \alpha}$$

is negative at some points. The proof of this is similar to that given in Proposition 4.2. $\square$

In section 2 we observed that when $n = 2$, the matrix G defined by (12) is positive. This matrix would have been positive for all n if the function

$$f(x) = \frac{\cosh 2x}{\cosh^4 x} = \frac{2}{\cosh^2 x} - \frac{1}{\cosh^4 x}$$

were positive definite. The Fourier transform of the function $1/\cosh^r x$ is known [9, p. 138]. Using that formula we see that

$$\hat{f}(\xi) = 2 \left| \Gamma \left(1 + \frac{i\xi}{2} \right) \right|^2 \left(1 - \frac{1}{3} \left| 1 + \frac{i\xi}{2} \right|^2 \right).$$

This is negative for large ξ .

5. Operator inequalities. Each of the positivity results in section 2 leads to inequalities for norms of operators. Let $\|A\|$ stand for the norm of A as a linear operator on the Hilbert space $\mathbb{C}^n$, and $\|A\|$ for any unitarily invariant norm—one that has the property $\|A\| = \|UAV\|$ for any unitary matrices U, V , and is normalized so that $\|A\| = \|A\|$ for any operator A of rank one. If A is positive, then for every B we have [10, p. 343]

$$(35) \quad \|A \circ B\| \leq \max_i a_{ii} \|B\|.$$

Let A, B be positive matrices, and let ν be any real number, $0 \leq \nu \leq 1$. Then for all X and all unitarily invariant norms

$$(36) \quad \|A^\nu X B^{1-\nu} + A^{1-\nu} X B^\nu\| \leq \|AX + XB\|.$$

For the norm $\|\cdot\|$ alone this was proved by Heinz [6] and used to derive many important results in perturbation theory. The general case, its history, and relations to other inequalities may be found in [1, Chapter IX].

In [4] Bhatia and Parthasarathy showed that for any positive real numbers $\lambda_1, \dots, \lambda_n$ and for $-1 < \alpha < 1$, the matrix P with entries

$$(37) \quad p_{ij} = \frac{\lambda_i^\alpha + \lambda_j^\alpha}{\lambda_i + \lambda_j}$$

is positive, and they used this fact and (35) to give a proof of (36). Following this method some quaint generalizations of (36) may be obtained using the matrices in (8), (9), and (13). For example, we can prove that

$$(38) \quad (1+t) \|A^\nu X B^{1-\nu} + A^{1-\nu} X B^\nu\| \leq \|AX + XB + t(A^{2/3} X B^{1/3} + A^{1/3} X B^{2/3})\|$$

for $-1 < t \leq 3$ and $\frac{1}{3} \leq \nu \leq \frac{2}{3}$.

To see this, write for $-1 < \alpha < 1$ and $-1 < t \leq 3$ the identity

$$(39) \quad \frac{\lambda_i^\alpha + \lambda_j^\alpha}{\lambda_i^3 + \lambda_j^3 + t(\lambda_i^2 \lambda_j + \lambda_i \lambda_j^2)} = \frac{\lambda_i^\alpha + \lambda_j^\alpha}{(\lambda_i + \lambda_j)} \frac{1}{(\lambda_i + \lambda_j)^2} \sum_{n=0}^{\infty} (3-t)^n \left(\frac{\lambda_i^2 \lambda_j + \lambda_i \lambda_j^2}{(\lambda_i + \lambda_j)^3} \right)^n.$$

The matrix whose ij entries are given by (39) is a sum of Schur products of several positive matrices. Hence it is positive. Multiply it on the left and the right by a diagonal matrix $\text{diag}(\lambda_1^r, \dots, \lambda_n^r)$, where $1 < r < 2$. Then put $\alpha = 3 - 2r$. This shows that the matrix with its ij entries given by

$$(40) \quad (1+t) \frac{\lambda_i^{3-r} \lambda_j^r + \lambda_i^r \lambda_j^{3-r}}{\lambda_i^3 + \lambda_j^3 + t(\lambda_i^2 \lambda_j + \lambda_i \lambda_j^2)}, \quad -1 < t \leq 3,$$

is positive and all its diagonal entries are equal to 1. Using the arguments in [2], one gets from this the inequality

$$(1+t) \||A^r X B^{3-r} + A^{3-r} X B^r\|| \leq \||A^3 X + X B^3 + t(A^2 X B + A X B^2)\||$$

for $-1 < t \leq 3$, $1 < r < 2$. Replace A and B by their cube roots and put $\nu = r/3$. This gives (38).

The choice $t = 0$ and $\nu = 1/2$ reduces (38) to the arithmetic-geometric mean inequality

$$\||A^{1/2} X B^{1/2}\|| \leq \frac{1}{2} \||AX + XB\||,$$

first proved by Bhatia and Davis [2], the special case $X = I$ having been proved earlier by Bhatia and Kittaneh [3].

Similar inequalities may be obtained by applying these considerations to (8) and to (13). The first of these was noted by Zhan [15]. One could also use the Schur product of the matrix (37) with itself. For example, one can write

$$\frac{(\lambda_i^\alpha + \lambda_j^\alpha)^2}{\lambda_i^2 + \lambda_j^2 + t\lambda_i \lambda_j} = \left(\frac{\lambda_i^\alpha + \lambda_j^\alpha}{\lambda_i + \lambda_j} \right)^2 \sum_{n=0}^{\infty} (2-t)^n \frac{\lambda_i^n \lambda_j^n}{(\lambda_i + \lambda_j)^{2n}}$$

for $-1 < \alpha < 1$ and $-2 < t \leq 2$ and get other operator inequalities.

The full power of this method has been exploited by Hiai and Kosaki in a series of interesting papers [7, 8, 11] and in the monograph [9], to which the reader should turn for a large collection of operator mean inequalities.

Acknowledgment. The first author thanks the Department of Mathematics, Kuwait University, for their generous hospitality during his visit that facilitated this work.

REFERENCES

- [1] R. BHATIA, *Matrix Analysis*, Springer-Verlag, New York, 1997.
- [2] R. BHATIA AND C. DAVIS, *More matrix forms of the arithmetic-geometric mean inequality*, SIAM J. Matrix Anal. Appl., 14 (1993), pp. 132–136.
- [3] R. BHATIA AND F. KITANEH, *On the singular values of a product of operators*, SIAM J. Matrix Anal. Appl., 11 (1990), pp. 272–277.
- [4] R. BHATIA AND K. R. PARTHASARATHY, *Positive definite functions and operator inequalities*, Bull. London Math. Soc., 32 (2000), pp. 214–228.
- [5] S. K. GODUNOV, *Modern Aspects of Linear Algebra*, Transl. Math. Monogr. 175, AMS, Providence, RI, 1998.
- [6] E. HEINZ, *Beiträge zur Störungstheorie der Spektralzerlegung*, Math. Ann., 123 (1951), pp. 415–438.
- [7] F. HIAI AND H. KOSAKI, *Comparison of various means for operators*, J. Funct. Anal., 163 (1999), pp. 300–323.

- [8] F. HIAI AND H. KOSAKI, *Means of matrices and comparison of their norms*, Indiana Univ. Math. J., 48 (1999), pp. 899–936.
- [9] F. HIAI AND H. KOSAKI, *Means of Hilbert Space Operators*, Lecture Notes in Math. 1820, Springer-Verlag, Berlin, 2003.
- [10] R. A. HORN AND C. R. JOHNSON, *Topics in Matrix Analysis*, Cambridge University Press, Cambridge, UK, 1990.
- [11] H. KOSAKI, *Arithmetic-geometric mean and related inequalities for operators*, J. Funct. Anal., 156 (1998), pp. 429–451.
- [12] M. K. KWONG, *On the definiteness of the solutions of certain matrix equations*, Linear Algebra Appl., 108 (1988), pp. 177–197.
- [13] A. M. LYAPUNOV, *The General Problem of the Stability of Motion*, English translation, Taylor and Francis, London, 1992.
- [14] D. S. MITRINOVIC AND J. D. KECKIC, *The Cauchy Method of Residues*, D. Reidel, Dordrecht, The Netherlands, 1984.
- [15] X. ZHAN, *Inequalities for unitarily invariant norms*, SIAM J. Matrix Anal. Appl., 20 (1998), pp. 466–470.

ON THE PSEUDOSPECTRA OF MATRIX POLYNOMIALS*

PETER LANCASTER[†] AND PANAYIOTIS PSARRAKOS[‡]

Abstract. The pseudospectra of a matrix polynomial $P(\lambda)$ are sets of complex numbers that are eigenvalues of matrix polynomials which are near to $P(\lambda)$; i.e., their coefficients are within some fixed magnitude of the coefficients of $P(\lambda)$. Pseudospectra provide important insights into the sensitivity of eigenvalues under perturbations and have several applications. First, qualitative properties concerning boundedness and connected components of pseudospectra are obtained. Then an accurate continuation algorithm for the numerical determination of the boundary of pseudospectra of matrix polynomials is devised and illustrated. This algorithm is based on a prediction-correction scheme.

Key words. matrix polynomial, eigenvalue, singular value, perturbation, ε -pseudospectrum, boundary, stability

AMS subject classifications. 65F15, 65F35, 93D09

DOI. 10.1137/S0895479804441420

1. Introduction and definitions. Let $\mathbb{C}^{n \times n}$ be the algebra of all $n \times n$ complex matrices and consider the

$$(1.1) \quad P(\lambda) = A_m \lambda^m + A_{m-1} \lambda^{m-1} + \cdots + A_1 \lambda + A_0,$$

where λ is a complex variable and $A_j \in \mathbb{C}^{n \times n}$, $j = 0, 1, \dots, m$, with $\det A_m \neq 0$. The study of matrix (and operator) polynomials has a long history, especially with regard to their spectral analysis; see [7, 8, 13, 16]. If $A_j^* = A_j$ ($j = 0, 1, \dots, m$), i.e., if all the coefficients of $P(\lambda)$ are Hermitian, then $P(\lambda)$ is said to be a *self-adjoint matrix polynomial*.

A scalar $\lambda_0 \in \mathbb{C}$ is said to be an *eigenvalue* of the matrix polynomial $P(\lambda)$ in (1.1) if the system $P(\lambda_0)v = 0$ has a nonzero solution $v_0 \in \mathbb{C}^n$. This solution v_0 is known as an *eigenvector* of $P(\lambda)$ corresponding to λ_0 . The set of all eigenvalues of $P(\lambda)$ is the *spectrum* of $P(\lambda)$, namely, $\sigma(P) = \{\lambda \in \mathbb{C} : \det P(\lambda) = 0\}$. Since $\det A_m \neq 0$, $\sigma(P)$ contains no more than nm distinct eigenvalues.

Consider the spectrum of perturbations of the matrix polynomial $P(\lambda)$ in (1.1) of the form

$$P_\Delta(\lambda) = (A_m + \Delta_m)\lambda^m + (A_{m-1} + \Delta_{m-1})\lambda^{m-1} + \cdots + (A_1 + \Delta_1)\lambda + A_0 + \Delta_0,$$

where the matrices $\Delta_0, \Delta_1, \dots, \Delta_m \in \mathbb{C}^{n \times n}$ are arbitrary. A *pseudospectrum* (introduced by Tisseur and Higham [22]) is defined as follows: For a given $\varepsilon > 0$ and a given set of nonnegative weights $\mathbf{w} = \{w_0, w_1, \dots, w_m\}$ (written $\mathbf{w} \geq 0$), the ε -*pseudospectrum* of $P(\lambda)$ with respect to $\mathbf{w}$ is defined to be

$$\sigma_{\varepsilon, \mathbf{w}}(P) = \{\lambda \in \mathbb{C} : \det P_\Delta(\lambda) = 0, \|\Delta_j\| \leq \varepsilon w_j, j = 0, 1, \dots, m\},$$

*Received by the editors February 19, 2004; accepted for publication (in revised form) December 2, 2004; published electronically July 13, 2005.

<http://www.siam.org/journals/simax/27-1/44142.html>

[†]Department of Mathematics and Statistics, University of Calgary, Calgary AB T2N 1N4, Canada (lancaste@math.ucalgary.ca).

[‡]Department of Mathematics, National Technical University, Zografou Campus, 15780 Athens, Greece (ppsarr@math.ntua.gr).

where $\|\cdot\|$ is any subordinate matrix norm. The parameters $w_0, w_1, \dots, w_m \geq 0$ allow freedom in how perturbations are measured: for example, in an absolute sense when $w_0 = w_1 = \dots = w_m = 1$, or in a relative sense when $w_j = \|A_j\|$ ($j = 0, 1, \dots, m$). Also, different values for the w_j admit different levels of confidence in the coefficients A_j . Note also that when $\varepsilon = 0$, $\sigma_{0,\mathbf{w}}(P) = \sigma(P)$.

Defining the associated compact set of perturbations of $P(\lambda)$,

$$\mathcal{B}(P, \varepsilon, \mathbf{w}) = \{P_\Delta(\lambda) : \|\Delta_j\| \leq \varepsilon w_j, j = 0, 1, \dots, m\},$$

the ε -pseudospectrum of $P(\lambda)$ can also be expressed in the form

$$\sigma_{\varepsilon,\mathbf{w}}(P) = \{\lambda \in \mathbb{C} : \det P_\Delta(\lambda) = 0, P_\Delta(\lambda) \in \mathcal{B}(P, \varepsilon, \mathbf{w})\}.$$

Observe that if $P(\lambda) = I\lambda - A$ for some $A \in \mathbb{C}^{n \times n}$, then $\sigma(P)$ coincides with the spectrum of A , $\sigma(A)$, in the usual sense. Furthermore, if we set $\mathbf{w} = \{w_0, w_1\} = \{1, 0\}$, then $\sigma_{\varepsilon,\mathbf{w}}(P)$ coincides with the ε -pseudospectrum of the matrix A , that is,

$$\sigma_\varepsilon(A) = \{\lambda \in \mathbb{C} : \lambda \in \sigma(A + \Delta_0), \|\Delta_0\| \leq \varepsilon\}.$$

The literature on pseudospectra of matrices (and operators) and their applications is extensive; see [2, 4, 15, 25] and the references therein. Also, the special case $n = 1$ is well understood. Thus, the ε -pseudospectrum of the scalar polynomial $p(\lambda) = a_m \lambda^m + a_{m-1} \lambda^{m-1} + \dots + a_1 \lambda + a_0$ coincides with the root neighborhood of $p(\lambda)$ introduced by Mosier [20] and the ε -pseudozero set of $p(\lambda)$ investigated by Toh and Trefethen [23].

The case for further development of algorithms for matrix polynomials rests largely on the pervasive second (and higher) degree polynomials used in the analysis of vibrating systems. Efficiencies are gained by avoiding linearizations and, as with problem areas already developed, pseudospectra give valuable insights into the sensitivities of spectra and, particularly, can be expected to clarify the effects of clustered and multiple eigenvalues.

The following lemma is one of the main tools used in this paper. Here, $s_{\min}(\cdot)$ denotes the minimum singular value of a complex matrix. The $q_{\mathbf{w}}(\cdot)$ is the matrix norm subordinate to the Euclidean vector norm and is consistently used throughout the remainder of this work. First define the scalar polynomial

$$q_{\mathbf{w}}(\lambda) = w_m \lambda^m + w_{m-1} \lambda^{m-1} + \dots + w_1 \lambda + w_0.$$

LEMMA 1.1. *Let $P(\lambda) = a_m \lambda^m + a_{m-1} \lambda^{m-1} + \dots + a_1 \lambda + a_0$ be a scalar polynomial and $\mathbf{w} = \{w_0, w_1, \dots, w_m\}$ a vector of nonnegative scalars. Then*

$$(1.2) \quad \sigma_{\varepsilon,\mathbf{w}}(P) = \{\lambda \in \mathbb{C} : s_{\min}(P(\lambda)) \leq \varepsilon q_{\mathbf{w}}(|\lambda|)\}.$$

Proof. This is just a special case of the important Lemma 2.1 of [22], which applies for any subordinate matrix norm. Here, one needs only that the spectral norm is unitarily invariant. $\square$

As the eigenvalues of $P_\Delta(\lambda)$ are continuous with respect to the entries of the coefficient matrices, it follows from the lemma that the boundary of the ε -pseudospectrum can be written in the form

$$(1.3) \quad \partial \sigma_{\varepsilon,\mathbf{w}}(P) = \{\lambda \in \mathbb{C} : s_{\min}(P(\lambda)) = \varepsilon q_{\mathbf{w}}(|\lambda|)\},$$

and $\partial \sigma_{\varepsilon,\mathbf{w}}(P)$ depends continuously on ε .

By using efficient algorithms for computing singular values, equations (1.2) and (1.3) become the main tools for estimation of the ε -pseudospectrum of a matrix polynomial. In particular, Tisseur and Higham (see [22]) obtain a graphical representation of $\sigma_{\varepsilon, \mathbf{w}}(P)$ by evaluating $s_{\min}(P(z))$ on a grid of points in the complex plane. One of our main objectives is the design of an alternative algorithm using a curve-tracing technique.

First of all, however, the geometry and the connected components of pseudospectra of matrix polynomials are studied. In the next two sections, basic boundedness properties of $\sigma_{\varepsilon, \mathbf{w}}(P)$ are obtained, as well as eigenvalue inclusion properties of the connected components of $\sigma_{\varepsilon, \mathbf{w}}(P)$. Section 4 contains the development of a curve-tracing algorithm for computing the boundary of pseudospectra. Our objective is to demonstrate that curve-following procedures for graphing pseudospectra for the classical eigenvalue problem (see [1, 2]) can be extended to apply directly to matrix polynomials with nonsingular leading coefficients. Numerical examples are given in the last section to demonstrate the feasibility of the method. Optimization of the algorithm with respect to efficiency is not considered here. Generalization of the method to admit matrix polynomials with singular leading coefficients (using ideas developed in [9]) should also be possible.

2. General properties. If the matrix polynomial $P(\lambda)$ in (1.1) is real (i.e., all its coefficients are real matrices) or self-adjoint, then it is well known that the spectrum of $P(\lambda)$ is symmetric with respect to the real axis. This symmetry also holds for the pseudospectra of matrix polynomials.

PROPOSITION 2.1. Let $P(\lambda) = A_m \lambda^m + \cdots + A_1 \lambda + A_0$ be an $n \times n$ matrix polynomial with $\varepsilon > 0$ and $\mathbf{w} \geq 0$. Suppose $\mu \in \sigma_{\varepsilon, \mathbf{w}}(P)$. Then there is a matrix polynomial $(A_m + \Delta_m) \lambda^m + \cdots + (A_1 + \Delta_1) \lambda + A_0 + \Delta_0 \in \mathcal{B}(P, \varepsilon, \mathbf{w})$ such that

$$\det[(A_m + \Delta_m) \mu^m + \cdots + (A_1 + \Delta_1) \mu + A_0 + \Delta_0] = 0.$$

If the coefficients of $P(\lambda)$ are real, then

$$\begin{aligned} & \det \left[\overline{(A_m + \Delta_m) \mu^m + \cdots + (A_1 + \Delta_1) \mu + A_0 + \Delta_0} \right] \\ &= \det [(A_m + \overline{\Delta_m}) \overline{\mu}^m + \cdots + (A_1 + \overline{\Delta_1}) \overline{\mu} + A_0 + \overline{\Delta_0}] = 0, \end{aligned}$$

where $(A_m + \overline{\Delta_m}) \lambda^m + \cdots + (A_1 + \overline{\Delta_1}) \lambda + A_0 + \overline{\Delta_0}$ also lies in $\mathcal{B}(P, \varepsilon, \mathbf{w})$. Hence, $\overline{\mu} \in \sigma_{\varepsilon, \mathbf{w}}(P)$.

Similarly, if all the coefficients of $P(\lambda)$ are Hermitian, then

$$\begin{aligned} & \det [(A_m + \Delta_m) \mu^m + \cdots + (A_1 + \Delta_1) \mu + A_0 + \Delta_0]^* \\ &= \det [(A_m + \Delta_m^*) \overline{\mu}^m + \cdots + (A_1 + \Delta_1^*) \overline{\mu} + A_0 + \Delta_0^*] = 0, \end{aligned}$$

where $(A_m + \Delta_m^*) \lambda^m + \cdots + (A_1 + \Delta_1^*) \lambda + A_0 + \Delta_0^* \in \mathcal{B}(P, \varepsilon, \mathbf{w})$. Thus, $\overline{\mu}$ lies in $\sigma_{\varepsilon, \mathbf{w}}(P)$. $\square$

Notice that if there is a perturbation $P_{\Delta}(\lambda) \in \mathcal{B}(P, \varepsilon, \mathbf{w})$ with identically zero determinant, then $\sigma_{\varepsilon, \mathbf{w}}(P)$ coincides with the whole complex plane so, a priori, the pseudospectrum may be unbounded. On the other hand, since the leading coefficient A_m is nonsingular, the matrix polynomial $P(\lambda)$ has exactly nm (finite) eigenvalues, counting multiplicities, so that for ε sufficiently small, the pseudospectrum must be

bounded and consist of no more than nm connected components. By extending a technique of Li and Rodman [14, Theorem 2.3], the next result establishes a necessary and sufficient condition for $\sigma_{\varepsilon, \mathbf{w}}(P)$ to be bounded (see also Proposition 3.5 of [10] for a similar result on stability radii of matrix polynomials).

THEOREM 2.2. *Let $P(\lambda) = (A_m + \Delta_m)\lambda^m + \cdots + (A_1 + \Delta_1)\lambda + A_0 + \Delta_0$ be a matrix polynomial of degree m with $A_m \in \mathbb{C}^{n \times n}$ and $\Delta_m \in \mathbb{C}^{n \times n}$. Suppose that $\sigma_{\varepsilon, \mathbf{w}}(P)$ is bounded. Then $\det(A_m + \Delta_m) \neq 0$ whenever $\|\Delta_m\| \leq \varepsilon w_m$, and*

$$\zeta_\varepsilon = \min\{|\det(A_m + \Delta_m)| : \|\Delta_m\| \leq \varepsilon w_m\} > 0.$$

Since the set $\mathcal{B}(P, \varepsilon, \mathbf{w})$ is compact, there is an $M_\varepsilon > 0$ such that for any perturbation of $P(\lambda)$,

$$P_\Delta(\lambda) = (A_m + \Delta_m)\lambda^m + \cdots + (A_1 + \Delta_1)\lambda + A_0 + \Delta_0 \in \mathcal{B}(P, \varepsilon, \mathbf{w}),$$

and for any $\lambda \in \mathbb{C}$ with $|\lambda| > M_\varepsilon$, we have

$$|\det P_\Delta(\lambda) - \det(A_m + \Delta_m)\lambda^{nm}| < \zeta_\varepsilon |\lambda^{nm}| \leq |\det(A_m + \Delta_m)\lambda^{nm}|$$

(keeping in mind that $\det P_\Delta(\lambda)$ is a scalar polynomial with leading term $\det(A_m + \Delta_m)\lambda^{nm}$). Hence, $\det P_\Delta(\lambda) \neq 0$, i.e., $\sigma_{\varepsilon, \mathbf{w}}(P) \subseteq \{\lambda \in \mathbb{C} : |\lambda| \leq M_\varepsilon\}$; $\sigma_{\varepsilon, \mathbf{w}}(P)$ is bounded.

To prove the converse, assume that $\sigma_{\varepsilon, \mathbf{w}}(P)$ is bounded but there is a

$$P_{\hat{\Delta}}(\lambda) = (A_m + \hat{\Delta}_m)\lambda^m + \cdots + (A_1 + \hat{\Delta}_1)\lambda + A_0 + \hat{\Delta}_0 \in \mathcal{B}(P, \varepsilon, \mathbf{w})$$

with $\det(A_m + \hat{\Delta}_m) = 0$. Then at least one of the coefficients of the scalar polynomial $\det P_{\hat{\Delta}}(\lambda)$ is nonzero; otherwise $\sigma_{\varepsilon, \mathbf{w}}(P) = \mathbb{C}$, a contradiction. Let the coefficient of λ^τ ($\tau \in \{0, 1, \dots, nm - 1\}$) in $\det P_{\hat{\Delta}}(\lambda)$ be nonzero and denote it by β_τ . Construct a sequence $\{\hat{\Delta}_{m,k}\}_{k \in \mathbb{N}} \subset \mathbb{C}^{n \times n}$ such that $\lim_{k \rightarrow \infty} \hat{\Delta}_{m,k} = \hat{\Delta}_m$ and

$$\det(A_m + \hat{\Delta}_{m,k}) \neq 0 \quad \text{and} \quad \|\hat{\Delta}_{m,k}\| \leq \varepsilon w_m \quad (k = 1, 2, \dots).$$

Clearly, for a fixed $\delta > 0$, $|\det(A_m + \hat{\Delta}_{m,k})| < \delta$ for all sufficiently large k . Since $\sigma_{\varepsilon, \mathbf{w}}(P)$ is bounded, the $(nm - \tau)$ th elementary symmetric function of the roots of $\det[(A_m + \hat{\Delta}_{m,k})\lambda^m + \cdots + (A_1 + \hat{\Delta}_1)\lambda + A_0 + \hat{\Delta}_0]$, which is equal to $\pm \beta_\tau / \det(A_m + \hat{\Delta}_{m,k})$, is bounded for all k . This contradicts the construction of the sequence $\{\hat{\Delta}_{m,k}\}_{k \in \mathbb{N}}$. $\square$

Theorem 2 of [20] can be generalized as follows.

THEOREM 2.3. *Let $P(\lambda) = (A_m + \Delta_m)\lambda^m + \cdots + (A_1 + \Delta_1)\lambda + A_0 + \Delta_0$ be a matrix polynomial of degree m with $A_m \in \mathbb{C}^{n \times n}$ and $\Delta_m \in \mathbb{C}^{n \times n}$. Suppose $\sigma_{\varepsilon, \mathbf{w}}(P)$ is bounded. It follows from Theorem 2.2 that for any perturbation of $P(\lambda)$,*

$$P_\Delta(\lambda) = (A_m + \Delta_m)\lambda^m + \cdots + (A_1 + \Delta_1)\lambda + A_0 + \Delta_0 \in \mathcal{B}(P, \varepsilon, \mathbf{w}),$$

$\det(A_m + \Delta_m) \neq 0$; i.e., the leading coefficient of the polynomial $\det P_\Delta(\lambda)$ is nonsingular. As a consequence, $P_\Delta(\lambda)$ has exactly nm eigenvalues, counting multiplicities, as does every member of the family of matrix polynomials

$$P_{\Delta,t}(\lambda) = (1-t)P(\lambda) + tP_\Delta(\lambda); \quad t \in [0, 1].$$

Moreover, for any $t \in [0, 1]$, $P_{\Delta,t}(\lambda)$ belongs to $\mathcal{B}(P, \varepsilon, \mathbf{w})$ and its eigenvalues lie in $\sigma_{\varepsilon, \mathbf{w}}(P)$.

The coefficients of the scalar polynomial $\det P_{\Delta,t}(\lambda)$ are continuous functions of $t \in [0, 1]$. Hence, by the continuity of the zeros of $\det P_{\Delta,t}(\lambda)$ with respect to its coefficients, as t varies from 0 to 1, the eigenvalues of $P_{\Delta,t}(\lambda)$ trace continuous paths from the eigenvalues of $P(\lambda)$ ($= P_{\Delta,0}(\lambda)$) to the eigenvalues of $P_\Delta(\lambda)$ ($= P_{\Delta,1}(\lambda)$). Thus, if $P(\lambda)$ has k eigenvalues (counting multiplicities) in a connected component $\mathcal{G}$ of $\sigma_{\varepsilon, \mathbf{w}}(P)$ and its $nm - k$ remaining eigenvalues are isolated in $\sigma_{\varepsilon, \mathbf{w}}(P) \setminus \mathcal{G}$, then this is true for the eigenvalues of every $P_{\Delta,t}(\lambda)$, $t \in [0, 1]$. Consequently, $P_\Delta(\lambda)$ has exactly k eigenvalues in $\mathcal{G}$, counting multiplicities.

Finally, note that each bounded connected component of $\sigma_{\varepsilon, \mathbf{w}}(P)$ contains at least one eigenvalue of $P(\lambda)$, and by the above discussion, it contains at least one eigenvalue of the perturbation $P_\Delta(\lambda)$. Hence, $\sigma_{\varepsilon, \mathbf{w}}(P)$ cannot have more than nm connected components. $\square$

Since the origin lies in $\sigma_{\varepsilon w_m}(A_m)$ if and only if $s_{\min}(A_m) \leq \varepsilon w_m$, we also have the following corollary.

COROLLARY 2.4. *If $\varepsilon > 0$, $\varepsilon w_m < s_{\min}(A_m)$, $\sigma_{\varepsilon, \mathbf{w}}(P)$ contains at most nm connected components.*

3. Matrix polynomials with bounded numerical range. The numerical range of the matrix polynomial $P(\lambda)$ is defined by

$$W(P) = \{\lambda \in \mathbb{C} : v^* P(\lambda) v = 0, v \in \mathbb{C}^n, v^* v = 1\}$$

(see, e.g., [14, 17, 18]), and it is always closed and contains $\sigma(P)$. For the linear pencil $I\lambda - A$ ($A \in \mathbb{C}^{n \times n}$), $W(I\lambda - A)$ coincides with the classical numerical range (also known as the F -range) of the matrix A , $F(A) = \{v^* A v : v \in \mathbb{C}^n, v^* v = 1\}$, which is always compact and convex [11]. The ε -pseudospectrum of A is defined by $\hat{\rho}(A) = \min\{|\lambda| : \lambda \in \partial F(A)\}$. By Theorem V.3.2 of [12], the ε -pseudospectrum of A lies in the region

$$F(A) + S(0, \varepsilon) = \{\lambda \in \mathbb{C} : \text{dist}(\lambda, F(A)) \leq \varepsilon\},$$

where $\text{dist}(\lambda, F(A))$ denotes the distance between point λ and the numerical range $F(A)$. Thus, $F(A) + S(0, \varepsilon)$ is often used as an initial region for the estimation of $\sigma_\varepsilon(A)$.

The numerical range $W(P)$ is bounded if and only if $0 \notin F(A_m)$, and in this case, it has no more than m connected components [14]. If $\mathcal{G}$ is a bounded connected component of $W(P)$, then for any unit vector $v \in \mathbb{C}^n$, the number of zeros of the scalar polynomial $v^* P(\lambda) v$ in $\mathcal{G}$, counting multiplicities, does not depend on v [16, Lemma 26.8]; i.e., it is constant. If we denote this constant number by $c(\mathcal{G})$, then $P(\lambda)$ has exactly $nc(\mathcal{G})$ eigenvalues in $\mathcal{G}$, counting multiplicities [18, Theorem 2.1]. Now Theorem V.3.2 of [12] can be generalized to the case of matrix polynomials. Although this result is mainly of theoretical interest, note that estimates of the inner numerical radius are available in [3].

THEOREM 3.1. Let $P(\lambda) = A_m \lambda^m + \cdots + A_1 \lambda + A_0$ be an $n \times n$ matrix polynomial with $\xi \leq m$ clusters $\mathcal{G}_1, \mathcal{G}_2, \dots, \mathcal{G}_\xi$ of eigenvalues. Let $\varepsilon > 0$ and $\mathbf{w} \geq 0$.

$$\sigma_{\varepsilon, \mathbf{w}}(P) \subseteq \left\{ \lambda \in \mathbb{C} : \prod_{j=1}^{\xi} \text{dist}(\lambda, \mathcal{G}_j)^{c(\mathcal{G}_j)} \leq \frac{\varepsilon q_{\mathbf{w}}(|\lambda|)}{\hat{r}(A_m)} \right\},$$

Suppose that $\text{dist}(\lambda_0, \mathcal{G}_j) = 0$ and $\lambda_0 \in \mathcal{G}_j$. Then there exist a perturbation of $P(\lambda)$,

$$P_{\Delta}(\lambda) = (A_m + \Delta_m)\lambda^m + \cdots + (A_1 + \Delta_1)\lambda + A_0 + \Delta_0 \in \mathcal{B}(P, \varepsilon, \mathbf{w}),$$

and a unit vector $v_0 \in \mathbb{C}^n$ such that $P_{\Delta}(\lambda_0)v_0 = 0$. Hence,

$$v_0^*(\Delta_m \lambda_0^m + \cdots + \Delta_1 \lambda_0 + \Delta_0)v_0 = -v_0^*P(\lambda_0)v_0$$

and, consequently,

$$\begin{aligned} \sum_{j=0}^m |v_0^* \Delta_j v_0| |\lambda_0|^j &\geq \left| \sum_{j=0}^m (v_0^* \Delta_j v_0) \lambda_0^j \right| = |v_0^* P(\lambda_0) v_0| \\ &= |v_0^* A_m v_0| \prod_{j=1}^m |\lambda_0 - \lambda_j(v_0)|, \end{aligned}$$

where $\lambda_1(v_0), \lambda_2(v_0), \dots, \lambda_m(v_0)$ are the zeros of the polynomial $v_0^* P(\lambda) v_0$. Since $\varepsilon w_j \geq \|\Delta_j\| \geq |v_0^* \Delta_j v_0|$, $j = 0, 1, \dots, m$, it follows that

$$\begin{aligned} \varepsilon q_{\mathbf{w}}(|\lambda_0|) &\geq \sum_{j=0}^m \|\Delta_j\| |\lambda_0|^j \geq \sum_{j=0}^m |v_0^* \Delta_j v_0| |\lambda_0|^j \\ &\geq \hat{r}(A_m) \prod_{j=1}^m |\lambda_0 - \lambda_j(v_0)| \geq \hat{r}(A_m) \prod_{j=1}^{\xi} \text{dist}(\lambda_0, \mathcal{G}_j)^{c(\mathcal{G}_j)}. \end{aligned}$$

The proof is complete. $\square$

4. A curve-tracing algorithm. For fixed $\varepsilon > 0$ and $\mathbf{w} = \{w_0, w_1, \dots, w_m\} \geq 0$, the boundary of the ε -pseudospectrum of the matrix polynomial $P(\lambda) = A_m \lambda^m + \cdots + A_1 \lambda + A_0$ is given by (1.3). In this section, we describe a prediction-correction continuation methodology for the computation of $\partial \sigma_{\varepsilon, \mathbf{w}}(P)$, extending an algorithm of Brühl [1, 2] for matrices. As noted in section 1, for our purposes, the pseudospectrum is that defined in terms of the spectral norm.

For convenience, define the function

$$g_P(x, y) = s_{\min}(P(x + iy)); \quad x, y \in \mathbb{R}.$$

When there is no fear of confusion, we write $g_P(\lambda) = s_{\min}(P(\lambda))$ for $\lambda \in \mathbb{C}$.

Our approach is based on the following result of Sun [21] concerning the differentiability of simple singular values.

THEOREM 4.1. Let $F(\mathbf{b}) : \mathbb{R}^d \mapsto \mathbb{C}^{n \times n}$ be a holomorphic function with $\mathbf{b}_0 = (b_0^{(1)}, b_0^{(2)}, \dots, b_0^{(d)}) \in \mathbb{R}^d$ such that s_0 is a simple singular value of $F(\mathbf{b}_0)$.

$$\begin{aligned}
& F(\mathbf{b}_0) = u_0 + v_0 \\
& \mathcal{N} = \mathbf{b}_0 \\
& s(\mathbf{b}) = F(\mathbf{b}), \quad u(\mathbf{b}) = u_0, \quad v(\mathbf{b}) = v_0 \\
& s(\mathbf{b}_0) = s_0, \quad u(\mathbf{b}_0) = u_0, \quad v(\mathbf{b}_0) = v_0 \\
& \mathcal{N} = \mathbf{b}_0 \\
& s(\mathbf{b}) = \mathbf{b}_0 \\
& \frac{\partial s(\mathbf{b}_0)}{\partial b^{(j)}} = \operatorname{Re} \left(u_0^* \frac{\partial F(\mathbf{b}_0)}{\partial b^{(j)}} v_0 \right); \quad j = 1, 2, \dots, d.
\end{aligned}$$

The next corollary is a direct consequence of this theorem.

COROLLARY 4.2. $\lambda_0 = x_0 + iy_0 \in \mathbb{C} \setminus \sigma(P)$, $s_{\min}(P(\lambda_0)) = u_0 + v_0$, $g_P(x, y) = (x_0, y_0) \in \mathbb{R}^2$,

$$\nabla g_P(x_0, y_0) = \left(\operatorname{Re} \left(u_0^* \frac{\partial P(x_0 + iy_0)}{\partial x} v_0 \right), \operatorname{Re} \left(u_0^* \frac{\partial P(x_0 + iy_0)}{\partial y} v_0 \right) \right).$$

The basic continuation method investigated here for finding points on the implicitly defined curve

$$\partial \sigma_{\varepsilon, \mathbf{w}}(P) = \{\lambda \in \mathbb{C} : g_P(\lambda) - \varepsilon q_{\mathbf{w}}(|\lambda|) = 0\}$$

consists of an initial step to find a starting point on the curve followed by a sequence of “predictor” steps tangential to the boundary and “corrector” steps to go back to the boundary.

For calculation of a first point on the boundary of $\sigma_{\varepsilon, \mathbf{w}}(P)$, let $\lambda_0 = x_0 + iy_0 \in \sigma_{\varepsilon, \mathbf{w}}(P) \setminus \sigma(P)$ and $d_0 \in \mathbb{C}$ be nonzero, and consider the function

$$h(t) = g_P(\lambda_0 + t d_0) - \varepsilon q_{\mathbf{w}}(|\lambda_0 + t d_0|); \quad t \in \mathbb{R}.$$

Then use Newton’s method to find a solution of $h(t) = 0$ along the straight line $\{\lambda_0 + t d_0 : t \in \mathbb{R}\}$ in the complex plane. Without loss of generality, it may be assumed that the initial value of t is $t_0 = 0$. Moreover, we assume that g_P is differentiable at λ_0 and the gradient $\nabla g_P(x_0, y_0)$ given by Corollary 4.2 is nonzero.

Then the first Newton iterate is

$$t_1 = t_0 - \frac{h(t_0)}{h'(t_0)} = - \frac{g_P(\lambda_0) - \varepsilon q_{\mathbf{w}}(|\lambda_0|)}{h'(0)},$$

where

$$\begin{aligned}
h'(0) &= \frac{\partial [g_P(\lambda_0) - \varepsilon q_{\mathbf{w}}(|\lambda_0|)]}{\partial d_0} = \frac{\partial [g_P(x_0 + iy_0) - \varepsilon q_{\mathbf{w}}(\sqrt{x_0^2 + y_0^2})]}{\partial d_0} \\
&= (\operatorname{Re} d_0) \left[\operatorname{Re} \left(u_0^* \frac{\partial P(x_0 + iy_0)}{\partial x} v_0 \right) - \varepsilon \frac{\partial q_{\mathbf{w}}(\sqrt{x_0^2 + y_0^2})}{\partial x} \right] \\
&\quad + (\operatorname{Im} d_0) \left[\operatorname{Re} \left(u_0^* \frac{\partial P(x_0 + iy_0)}{\partial y} v_0 \right) - \varepsilon \frac{\partial q_{\mathbf{w}}(\sqrt{x_0^2 + y_0^2})}{\partial y} \right].
\end{aligned}$$

For brevity, write the partial derivatives on the right in the form

$$\operatorname{Re} \left(u_0^* \frac{\partial P(x_0 + iy_0)}{\partial x} v_0 \right) - \varepsilon \frac{\partial q_{\mathbf{w}} \left(\sqrt{x_0^2 + y_0^2} \right)}{\partial x} = \mathcal{R}_{\varepsilon, \mathbf{w}}(\lambda_0, x)$$

and

$$\operatorname{Re} \left(u_0^* \frac{\partial P(x_0 + iy_0)}{\partial y} v_0 \right) - \varepsilon \frac{\partial q_{\mathbf{w}} \left(\sqrt{x_0^2 + y_0^2} \right)}{\partial y} = \mathcal{R}_{\varepsilon, \mathbf{w}}(\lambda_0, y).$$

Then

$$(4.1) \quad z_1 = \lambda_0 + t_1 d_0 = x_0 + iy_0 - \frac{g_P(x_0 + iy_0) - \varepsilon q_{\mathbf{w}} \left(\sqrt{x_0^2 + y_0^2} \right)}{(\operatorname{Re} d_0) \mathcal{R}_{\varepsilon, \mathbf{w}}(\lambda_0, x) + (\operatorname{Im} d_0) \mathcal{R}_{\varepsilon, \mathbf{w}}(\lambda_0, y)} d_0.$$

Since the point $\lambda_0 = x_0 + iy_0$ lies in $\sigma_{\varepsilon, \mathbf{w}}(P)$, for suitably chosen direction d_0 , we can estimate a boundary point of $\sigma_{\varepsilon, \mathbf{w}}(P)$ (in particular, an intersection point of $\partial \sigma_{\varepsilon, \mathbf{w}}(P)$ and $\{\lambda_0 + t d_0 : t \in \mathbb{R}\}$) by repeating (4.1) until the quantity $|s_{\min}(P(z)) - \varepsilon q_{\mathbf{w}}(|z|)|$ is small enough.

It is also worth noting that

$$(\operatorname{Re} d_0) \mathcal{R}_{\varepsilon, \mathbf{w}}(\lambda_0, x) + (\operatorname{Im} d_0) \mathcal{R}_{\varepsilon, \mathbf{w}}(\lambda_0, y) = \operatorname{Re} [\bar{d}_0 (\mathcal{R}_{\varepsilon, \mathbf{w}}(\lambda_0, x) + i \mathcal{R}_{\varepsilon, \mathbf{w}}(\lambda_0, y))],$$

and on choosing the direction $d_0 = \mathcal{R}_{\varepsilon, \mathbf{w}}(\lambda_0, x) + i \mathcal{R}_{\varepsilon, \mathbf{w}}(\lambda_0, y)$, equation (4.1) implies

$$(4.2) \quad z_1 = x_0 + iy_0 - \frac{g_P(x_0 + iy_0) - \varepsilon q_{\mathbf{w}} \left(\sqrt{x_0^2 + y_0^2} \right)}{\mathcal{R}_{\varepsilon, \mathbf{w}}(\lambda_0, x) - i \mathcal{R}_{\varepsilon, \mathbf{w}}(\lambda_0, y)}.$$

Assuming now that the point $z_{k-1} \in \partial \sigma_{\varepsilon, \mathbf{w}}(P)$ has been computed and τ_k is the corresponding step-length, the (tangential) prediction for the k th boundary point of $\sigma_{\varepsilon, \mathbf{w}}(P)$, z_k , is

$$\begin{aligned} \hat{z}_k &= z_{k-1} + \tau_k \left(i \frac{\nabla [g_P(z_{k-1}) - \varepsilon q_{\mathbf{w}}(|z_{k-1}|)]}{|\nabla [g_P(z_{k-1}) - \varepsilon q_{\mathbf{w}}(|z_{k-1}|)]|} \right) \\ &= z_{k-1} + \tau_k \left(i \frac{(\mathcal{R}_{\varepsilon, \mathbf{w}}(z_{k-1}, x), \mathcal{R}_{\varepsilon, \mathbf{w}}(z_{k-1}, y))}{|(\mathcal{R}_{\varepsilon, \mathbf{w}}(z_{k-1}, x), \mathcal{R}_{\varepsilon, \mathbf{w}}(z_{k-1}, y))|} \right); \end{aligned}$$

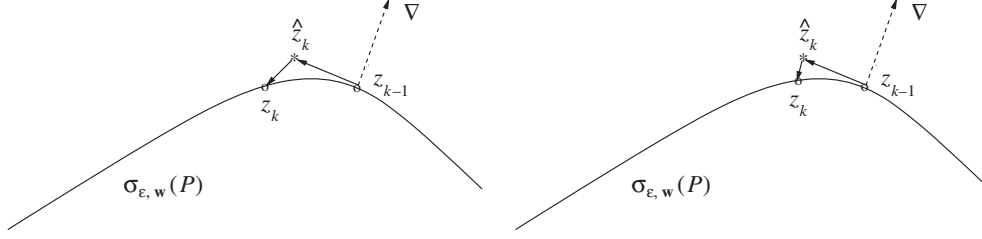
i.e., the direction tangential to $\partial \sigma_{\varepsilon, \mathbf{w}}(P)$ is chosen.

For sufficiently small τ_k , the correction step consists of a single Newton iterate with respect to the equation $g_P(\hat{z}_k + t d_k) - \varepsilon q_{\mathbf{w}}(|\hat{z}_k + t d_k|) = 0$, with an appropriate direction d_k and initial value $t_0 = 0$. It has been found that one Newton step gives adequate numerical performance, although the effect of taking more steps at this stage could be a subject for further investigation.

A natural choice for d_k is the gradient

$$\nabla [g_P(\hat{z}_k) - \varepsilon q_{\mathbf{w}}(|\hat{z}_k|)] = (\mathcal{R}_{\varepsilon, \mathbf{w}}(\hat{z}_k, x), \mathcal{R}_{\varepsilon, \mathbf{w}}(\hat{z}_k, y)).$$

In this case, step (4.2) is applied, and the estimation of z_k requires the computation of the singular values $s_{\min}(P(z_{k-1}))$ and $s_{\min}(P(\hat{z}_k))$ and their associated left and right singular vectors.

FIG. 4.1. Choosing the direction d_k in the correction step.

However, the computation of $s_{\min}(P(\hat{z}_k))$ and the corresponding singular vectors can be avoided (and the computational cost of the algorithm reduced by about a half) if the correction step is taken in the direction of

$$\nabla [g_P(z_{k-1}) - \varepsilon q_{\mathbf{w}}(|z_{k-1}|)] = (\mathcal{R}_{\varepsilon, \mathbf{w}}(z_{k-1}, x), \mathcal{R}_{\varepsilon, \mathbf{w}}(z_{k-1}, y))$$

and (4.2) is written in the form

$$z_k = \hat{z}_k - \frac{g_P(z_{k-1}) - \varepsilon q_{\mathbf{w}}(|z_{k-1}|)}{\mathcal{R}_{\varepsilon, \mathbf{w}}(z_{k-1}, x) - i\mathcal{R}_{\varepsilon, \mathbf{w}}(z_{k-1}, y)}.$$

In Figure 4.1, a graphical illustration of the directions $d_k = \nabla [g_P(\hat{z}_k) - \varepsilon q_{\mathbf{w}}(|\hat{z}_k|)]$ (left part) and $d_k = \nabla [g_P(z_{k-1}) - \varepsilon q_{\mathbf{w}}(|z_{k-1}|)]$ (right part) is given. The prediction points are marked with asterisks, the correction points are plotted as “o”, and ∇ denotes the gradient $\nabla [g_P(z_{k-1}) - \varepsilon q_{\mathbf{w}}(|z_{k-1}|)]$. Note that in the right part of the figure, the line segment $[\hat{z}_k, z_k]$ is parallel to ∇ .

ALGORITHM.

Input. The coefficients $A_0, A_1, \dots, A_m$ of the matrix polynomial $P(\lambda)$,
the parameter $\varepsilon > 0$ and the weights $w_0, w_1, \dots, w_m \geq 0$,
the number N of points to be determined on $\partial\sigma_{\varepsilon, \mathbf{w}}(P)$,
an approximation μ_0 of an eigenvalue of $P(\lambda)$,
the search direction d_0 for the initial point,
the relative accuracy tol for the initial point,
the step-length r .

Step 1. $\mu_0, \mu_1, \dots, \mu_N, \mu_{N+1}, \dots, \mu_{2N}, \mu_{2N+1}, \dots$

Set $z_{\text{new}} = \mu_0 + \varepsilon d_0$.

While $|s_{\min}(P(z_{\text{new}})) - \varepsilon q_{\mathbf{w}}(|z_{\text{new}}|)| > \varepsilon q_{\mathbf{w}}(|z_{\text{new}}|) tol$, repeat:

- (a) Set $z_{\text{old}} = z_{\text{new}}$.
- (b) Compute the minimum singular value $s_{\min}$ of $P(z_{\text{old}})$ and associated left and right singular vectors u_0 and v_0 .
- (c) Compute the gradient

$$\nabla [g_P(z_{\text{old}}) - \varepsilon q_{\mathbf{w}}(|z_{\text{old}}|)] = (\mathcal{R}_{\varepsilon, \mathbf{w}}(z_{\text{old}}, x), \mathcal{R}_{\varepsilon, \mathbf{w}}(z_{\text{old}}, y)).$$

- (d) Compute the next Newton iterate

$$z_{\text{new}} = z_{\text{old}} - \frac{s_{\min}(P(z_{\text{old}})) - \varepsilon q_{\mathbf{w}}(|z_{\text{old}}|)}{(\operatorname{Re} d_0)\mathcal{R}_{\varepsilon, \mathbf{w}}(z_{\text{old}}, x) + (\operatorname{Im} d_0)\mathcal{R}_{\varepsilon, \mathbf{w}}(z_{\text{old}}, y)} d_0.$$

End while

Set $z_1 = z_{\text{new}}$.

Step 2. $\tau_k = \tau_{k-1} + \tau_{k-1} \frac{\tau_{k-1}}{N-1}$ for $k = 2, 3, \dots, N$

For $k = 2, 3, \dots, N$, repeat:

- (a) Compute the minimum singular value $s_{\min}$ of $P(z_{k-1})$ and associated left and right singular vectors u_0 and v_0 .
- (b) Compute the gradient

$$\nabla [g_P(z_{k-1}) - \varepsilon q_{\mathbf{w}}(|z_{k-1}|)] = (\mathcal{R}_{\varepsilon, \mathbf{w}}(z_{k-1}, x), \mathcal{R}_{\varepsilon, \mathbf{w}}(z_{k-1}, y))$$

and the direction

$$d_k = i \frac{\mathcal{R}_{\varepsilon, \mathbf{w}}(z_{k-1}, x) + i \mathcal{R}_{\varepsilon, \mathbf{w}}(z_{k-1}, y)}{|\mathcal{R}_{\varepsilon, \mathbf{w}}(z_{k-1}, x) + i \mathcal{R}_{\varepsilon, \mathbf{w}}(z_{k-1}, y)|}.$$

- (c) Compute the predicted point

$$\hat{z}_k = z_{k-1} + d_k r.$$

- (d) Compute the corrected point

$$z_k = \hat{z}_k - \frac{s_{\min}(P(z_{k-1})) - \varepsilon q_{\mathbf{w}}(|z_{k-1}|)}{\mathcal{R}_{\varepsilon, \mathbf{w}}(z_{k-1}, x) - i \mathcal{R}_{\varepsilon, \mathbf{w}}(z_{k-1}, y)}.$$

End for

Output. The points $z_1, z_2, \dots, z_N$.

Of course, this algorithm tracks the boundary of that connected component of the pseudospectrum $\sigma_{\varepsilon, \mathbf{w}}(P)$ containing μ_0 . Consequently, for a complete picture of $\sigma_{\varepsilon, \mathbf{w}}(P)$, it may be necessary to repeat the procedure several times with different (appropriate) values of μ_0 .

Since the size of the step-lengths τ_k ($k = 2, 3, \dots, N$) used in the prediction step affects the accuracy and the computational cost of the algorithm, it is important to obtain criteria for their selection. This is an open problem, which is partially solved by Bekas and Gallopoulos [1]: an efficient choice of τ_k will undoubtedly depend mainly on the local shape of the curve $\partial\sigma_{\varepsilon, \mathbf{w}}(P)$. The present experiments are confined to a constant step-length, r (see (c) of Step 2). In particular, it is apparent that a procedure in which step-size is related to an estimate of the local curvature might be advantageous. This and related questions must, however, be deferred to future research.

As might be expected, difficulties appear near points of $\partial\sigma_{\varepsilon, \mathbf{w}}(P)$ where the function g_P is not differentiable and the minimum singular value is multiple, and also near points where the distance between distinct connected components of $\partial\sigma_{\varepsilon, \mathbf{w}}(P)$ becomes relatively small (see Example 5.2 below). In these cases, the curve-tracing algorithm may lose its path or retrace its own steps. Some of these difficulties can be resolved by choosing a smaller r (increasing the cost) and/or more suitable values for μ_0 and d_0 .

In spite of these apparent weaknesses, the authors' experiments suggest that the algorithm is remarkably robust.

An important feature of this technique is that it does not require a priori knowledge of the size or shape of $\sigma_{\varepsilon, \mathbf{w}}(P)$, since it determines the connected components of the pseudospectrum one after the other by using starting points close to eigenvalues. Moreover, the cost does not depend strongly on the degree of the matrix polynomial $P(\lambda)$. This parameter, m , appears only in the calculation of $P(z)$, $q(|z|)$, and their partial derivatives. The main cost of the algorithm comes from the computation of the

singular values $s_{\min}(P(z_{k-1}))$, $k = 2, 3, \dots, N$, and associated singular vectors. For this task, the suggested reference is [22], where five techniques (the Golub–Reinsch SVD algorithm, the transfer function approach, and the computation of solvents by using the generalized Schur decomposition, the Newton method, or the Bernoulli iteration) are compared in terms of flops and execution time (see also [15, 24]). In the examples of the next section, the SVD technique has been used.

5. Numerical examples. We give some examples to illustrate our results and the use of the proposed algorithm.

5.1 (a wing problem). Consider the quadratic matrix polynomial $Q(\lambda) = A_2\lambda^2 + A_1\lambda + A_0$ with

$$A_2 = \begin{bmatrix} 17.6 & 1.28 & 2.89 \\ 1.28 & 0.824 & 0.413 \\ 2.89 & 0.413 & 0.725 \end{bmatrix}, \quad A_1 = \begin{bmatrix} 7.66 & 2.45 & 2.1 \\ 0.23 & 1.04 & 0.223 \\ 0.6 & 0.756 & 0.658 \end{bmatrix},$$

$$\text{and } A_0 = \begin{bmatrix} 121 & 18.9 & 15.9 \\ 0 & 2.7 & 0.145 \\ 11.9 & 3.64 & 15.5 \end{bmatrix}.$$

The eigenproblem of $Q(\lambda)$ arose from a study of the oscillations of a wing in an airstream and originates in [6, section 10.11]. It has also been examined in section 5.3 of [13], and comparison with the recent treatment of the same example in [22] is interesting. The left part of Figure 5.1 indicates the boundaries of the ε -pseudospectra of $Q(\lambda)$ for $w_0 = w_1 = w_2 = 1$ (i.e., for perturbations measured in the absolute sense) and $\varepsilon = 0.01, 0.1, 0.15$, using the present path-tracing algorithm. For step-length $r = 0.1$, approximately 2000 boundary points are required.

Symmetry of the pseudospectra with respect to the real axis is apparent, confirming Proposition 2.1. The eigenvalues of $Q(\lambda)$ are $-0.88 \pm i 8.44$, $0.09 \pm i 2.52$, $-0.92 \pm i 1.76$, and they are plotted as “+”. As observed in [22], the eigenvalues $-0.88 \pm i 8.44$ are seen to be much more sensitive to these perturbations than the remaining eigenvalues of $Q(\lambda)$, and for $\varepsilon < s_{\min}(A_2) \cong 0.17$, $\sigma_{\varepsilon, \mathbf{w}}(Q)$ is bounded and consists of six connected components, confirming Corollary 2.4. On the other hand, in the right part of Figure 5.1, we see that the two connected components corresponding to the pair $-0.88 \pm i 8.44$ become an unbounded connected component with a “hole,” where the rest of $\sigma_{\varepsilon, \mathbf{w}}(Q)$ lies, when $\varepsilon = 0.18 > s_{\min}(A_2)$ (i.e., the exterior of the outermost 0.18-curve is in $\sigma_{0.18, \mathbf{w}}(Q)$). The four components of $\sigma_{\varepsilon, \mathbf{w}}(Q)$ that correspond to the eigenvalues $0.09 \pm i 2.52$, $-0.92 \pm i 1.76$ are magnified (using step-length $r = 0.03$) in the left part and the right part of Figure 5.2 for $\varepsilon = 0.01, 0.1, 0.15$ and for $\varepsilon = 0.18$, respectively.

The cost of the algorithm depends mainly on Step 2 since Step 1 usually demands a small number of Newton iterations. In the above example, for initial point $\mu_0 \cong -0.88 + i 8.44$, search direction $d_0 = 1$, and relative accuracy $tol = 10^{-3}$, the estimation of the first boundary point of $\sigma_{\varepsilon, \mathbf{w}}(Q)$ (for $\varepsilon = 0.1, 0.15, 0.18$) requires only four Newton iterations.

Notice also that in all the examples herein, the matrix polynomials are real and thus we can exploit the symmetry of their pseudospectra with respect to the real axis and need only compute the parts of the boundaries in the closed upper half-plane.

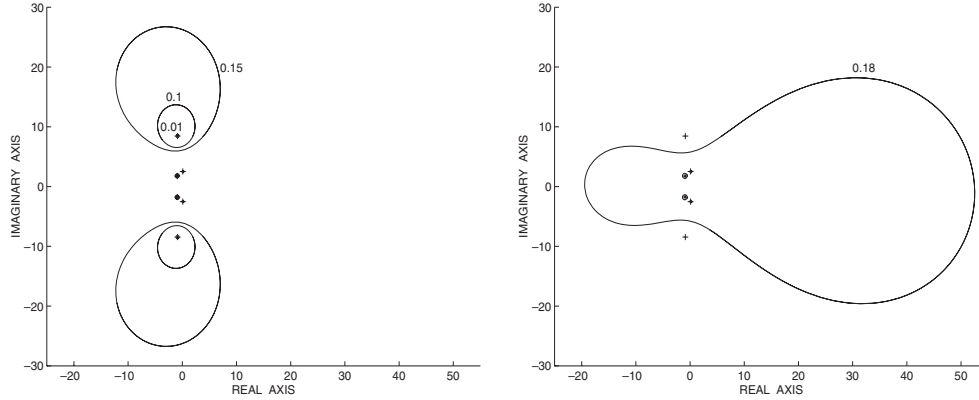
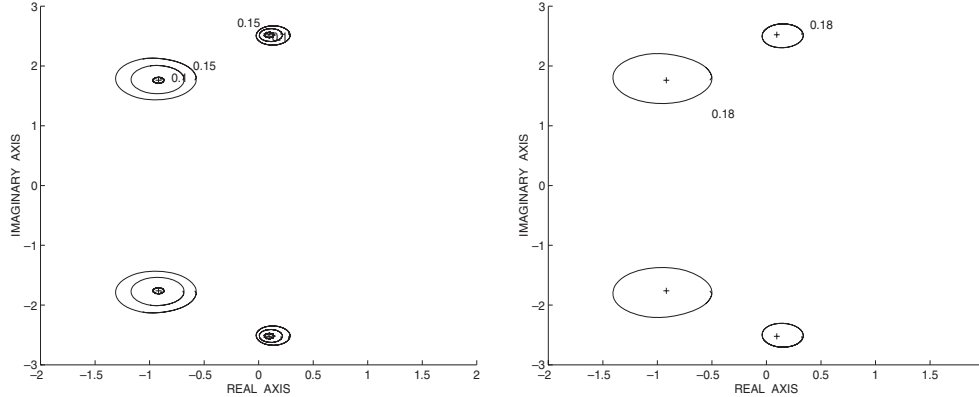
FIG. 5.1. The boundaries $\partial\sigma_{\varepsilon,\mathbf{w}}(Q)$ for $\varepsilon = 0.01, 0.1, 0.15$ and for $\varepsilon = 0.18$.

FIG. 5.2. The less sensitive eigenvalues.

5.2 (a vibrating system). The 3×3 self-adjoint matrix polynomial

$$P(\lambda) = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 2 & 0 \\ 0 & 0 & 5 \end{bmatrix} \lambda^2 + \begin{bmatrix} 0 & 0 & 0 \\ 0 & 3 & -1 \\ 0 & -1 & 6 \end{bmatrix} \lambda + \begin{bmatrix} 2 & -1 & 0 \\ -1 & 3 & 0 \\ 0 & 0 & 10 \end{bmatrix}$$

corresponds to a mass-spring model described in [5, Example 6]. The predicted boundaries of the ε -pseudospectra of $P(\lambda)$ for $w_0 = \|A_0\| = 10$, $w_1 = \|A_1\| = 6.3$, and $w_2 = \|A_2\| = 5$ (i.e., for perturbations measured in a relative sense) and for $\varepsilon = 0.02, 0.05, 0.1$ are shown in Figure 5.3. The eigenvalues of $P(\lambda)$, $-0.08 \pm i1.45$, $-0.75 \pm i0.86$, $-0.51 \pm i1.25$, are plotted as “+”.

Once again, the predicted symmetry with respect to the real axis is confirmed. Note also that all the pseudospectra are bounded and $s_{\min}(A_2) = 1 > \varepsilon w_2$, verifying Theorem 2.2.

Figure 5.4 shows how the close proximity of two connected components of the pseudospectrum may affect the curve-tracing algorithm. In particular, $\sigma_{0.06,\mathbf{w}}(P)$ consists of two connected components, one in the open upper half-plane and one in

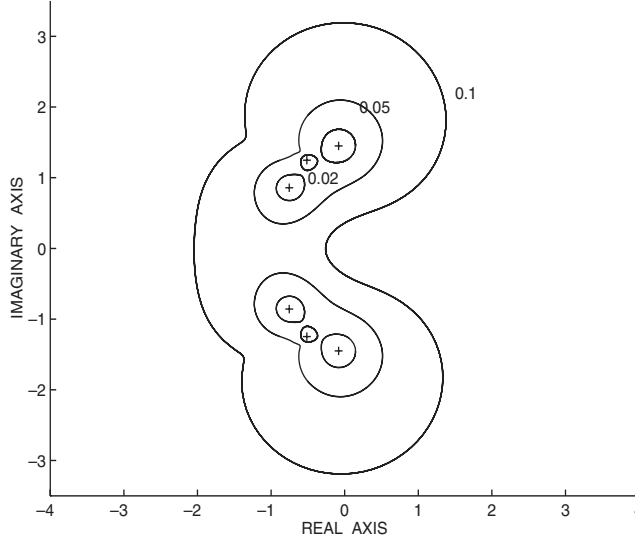
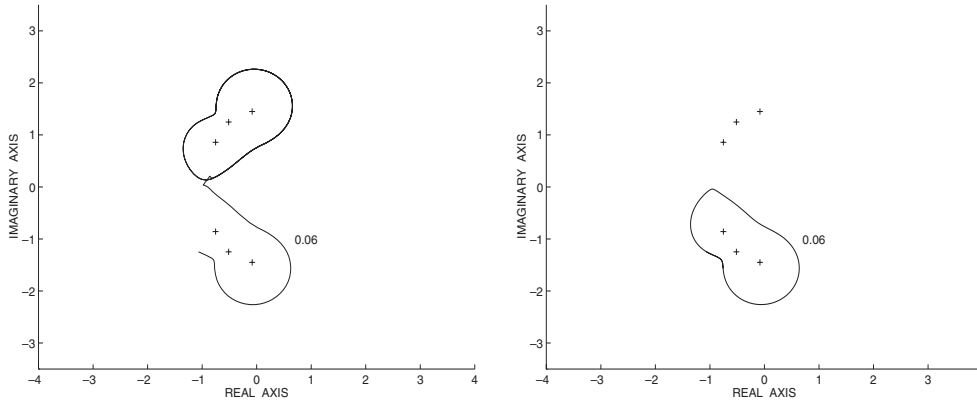


FIG. 5.3. A damped vibrating system.

FIG. 5.4. Step-lengths $r = 0.03$ and $r = 0.003$.

the open lower half-plane of $\mathbb{C}$. Using a starting point μ_0 close to the eigenvalue $-0.51 - i1.25$, direction $d_0 = -1$, and step-length $r = 0.03$, the algorithm sketches a part of the boundary of the lower connected component (in particular, 164 points), and on arriving at the top of the component, it “loses its way” and starts tracing the boundary of the upper connected component, where it remains forever (see the left part of the figure). If the step-length is decreased to $r = 0.003$, then the algorithm remains on the correct path and plots exactly the boundary of the lower component (right part of the figure). Since $r = 0.003$, approximately 2200 points are needed to complete the picture.

In our last example, we consider a quadratic matrix polynomial of larger size.

Example 5.3 (a gyroscopic system). Let B denote the 10×10 nilpotent matrix having ones on the subdiagonal and zeros elsewhere, and define the matrices $\hat{M} =$

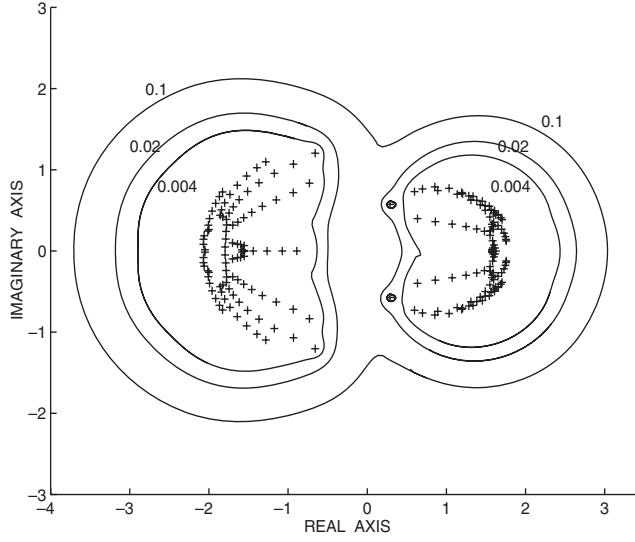


FIG. 5.5. A damped gyroscopic system.

$(4I_{10} + B + B^T)/6$, $\hat{G} = B - B^T$, and $\hat{K} = B + B^T - 2I_{10}$. Then, using the tensor product, we set

$$\begin{aligned} M &= I_{10} \otimes \hat{M} + 1.30\hat{M} \otimes I_{10}, \\ G &= 1.35I_{10} \otimes \hat{G} + 1.10\hat{G} \otimes I_{10}, \\ K &= I_{10} \otimes \hat{K} + 1.20\hat{K} \otimes I_{10}, \end{aligned}$$

and observe that $M = M^T$, $G = -G^T$, and $K = K^T$. Moreover, the matrix M is positive definite with $s_{\min}(M) \cong 0.8$ and K is negative definite. The 100×100 matrix polynomial $M\lambda^2 + G\lambda + K$ corresponds to an undamped gyroscopic system, and its eigenproblem has been examined in [19, Example 6.1]. Adding the tridiagonal damping matrix $D = \text{tridiag}\{-0.1, 0.3, -0.1\}$ to the linear term yields the matrix polynomial

$$R(\lambda) = M\lambda^2 + (G + D)\lambda + K.$$

The pseudospectra $\sigma_{\varepsilon, \mathbf{w}}(R)$ for $\mathbf{w} = \{1, 1, 1\}$ and for $\varepsilon = 0.004, 0.02, 0.1$ consist of four, two, and one bounded connected components, respectively, and their boundaries are shown in Figure 5.5. The eigenvalues of $R(\lambda)$ are plotted as “+”. As in the two previous examples, the results of section 2 are apparently confirmed.

Obviously, the classical grid method of [22] handles several ε ’s at once, while we have to repeat the path-tracing algorithm for each component and each ε . On the other hand, the new procedure is based on a one-dimensional grid (instead of a predefined two-dimensional grid), and in terms of the number of points at which it is necessary to compute $s_{\min}$, the curve-tracing method competes well with the grid method and is computationally less demanding when seeking a small number of boundary curves. In the above example, for step-length $r = 0.06$, the determination of the boundaries of $\sigma_{0.004, \mathbf{w}}(R)$, $\sigma_{0.02, \mathbf{w}}(R)$, and $\sigma_{0.1, \mathbf{w}}(R)$ in Figure 5.5 requires 292, 304, and 310 points, respectively. For the same step-length, the grid method demands a priori knowledge of the size of pseudospectra and a 117×84 grid of the rectangle $\Omega = [-4, 3] \times [-i2.5, i2.5]$, that is, 9828 grid points.

Acknowledgments. The authors are grateful to F. Tisseur and N. J. Higham for helpful comments.

REFERENCES

- [1] C. BEKAS AND E. GALLOPOULOS, *Cobra: Parallel path following for computing the matrix pseudospectrum*, Parallel Comput., 27 (2001), pp. 1879–1896.
- [2] M. BRÜHL, *A curve tracing algorithm for computing the pseudospectrum*, BIT, 36 (1996), pp. 441–454.
- [3] C.-H. CHENG AND N. HIGHAM, *The nearest definite pair for the Hermitian generalized eigenvalue problem*, Linear Algebra Appl., 302-303 (1999), pp. 63–76.
- [4] M. EMBREE AND L. N. TREFETHEN, *Generalizing eigenvalue theorems to pseudospectra theorems*, SIAM J. Sci. Comput., 23 (2001), pp. 583–590.
- [5] S. FALK, *Klassifikation gedämpfter Schwingungssysteme und Eingrenzung ihrer Eigenwerte*, Ing.-Arch., 29 (1960), pp. 436–444.
- [6] R. A. FRAZER, W. J. DUNCAN, AND A. R. COLLAR, *Elementary Matrices and Some Applications to Dynamics and Differential Equations*, Cambridge University Press, Cambridge, UK, 1938.
- [7] F. R. GANTMACHER, *The Theory of Matrices*, Chelsea, New York, 1959.
- [8] I. GOHBERG, P. LANCASTER, AND L. RODMAN, *Matrix Polynomials*, Academic Press, New York, 1982.
- [9] N. J. HIGHAM AND F. TISSEUR, *More on pseudospectra for polynomial eigenvalue problems and applications in control theory*, Linear Algebra Appl., 351-352 (2002), pp. 435–453.
- [10] D. HINRICHSSEN, N. K. SON, AND P. H. A. NGOC, *Stability radii of higher order positive difference systems*, Systems Control Lett., 49 (2003), pp. 377–388.
- [11] R. A. HORN AND C. R. JOHNSON, *Topics in Matrix Analysis*, Cambridge University Press, Cambridge, UK, 1991.
- [12] T. KATO, *Perturbation Theory of Linear Operators*, Springer-Verlag, Berlin, 1976.
- [13] P. LANCASTER, *Lambda-Matrices and Vibrating Systems*, Pergamon Press, Oxford, 1966; Dover, New York, 2002.
- [14] C.-K. LI AND L. RODMAN, *Numerical range of matrix polynomials*, SIAM J. Matrix Anal. Appl., 15 (1994), pp. 1256–1265.
- [15] S. H. LUI, *Computation of pseudospectra by continuation*, SIAM J. Sci. Comput., 18 (1997), pp. 565–573.
- [16] A. S. MARKUS, *Introduction to the Spectral Theory of Polynomial Operator Pencils*, Transl. Math. Monogr. 71, AMS, Providence, RI, 1988.
- [17] J. MAROULAS AND P. PSARRAKOS, *The boundary of numerical range of matrix polynomials*, Linear Algebra Appl., 267 (1997), pp. 101–111.
- [18] J. MAROULAS AND P. PSARRAKOS, *On the connectedness of numerical range of matrix polynomials*, Linear Algebra Appl., 280 (1998), pp. 97–108.
- [19] V. MEHRMANN AND D. WATKINS, *Structure-preserving methods for computing eigenpairs of large sparse skew-Hamiltonian/Hamiltonian pencils*, SIAM J. Sci. Comput., 22 (2001), pp. 1905–1925.
- [20] R. G. MOSIER, *Root neighborhoods of a polynomial*, Math. Comp., 47 (1986), pp. 265–273.
- [21] J.-G. SUN, *A note on simple non-zero singular values*, J. Comput. Math., 6 (1988), pp. 258–266.
- [22] F. TISSEUR AND N. J. HIGHAM, *Structured pseudospectra for polynomial eigenvalue problems, with applications*, SIAM J. Matrix Anal. Appl., 23 (2001), pp. 187–208.
- [23] K.-C. TOH AND L. N. TREFETHEN, *Pseudozeros of polynomials and pseudospectra of companion matrices*, Numer. Math., 68 (1994), pp. 403–425.
- [24] K.-C. TOH AND L. N. TREFETHEN, *Calculation of pseudospectra by the Arnoldi iteration*, SIAM J. Sci. Comput., 17 (1996), pp. 1–15.
- [25] L. N. TREFETHEN, *Pseudospectra of linear operators*, SIAM Rev., 39 (1997), pp. 383–406.

INTERLACING PROPERTIES FOR HERMITIAN MATRICES WHOSE GRAPH IS A GIVEN TREE*

C. M. DA FONSECA[†]

Abstract. We extend some interlacing properties of the eigenvalues of tridiagonal matrices to Hermitian matrices whose graph is a tree. We also give a graphical interpretation of the results. We use the work on matchings polynomials by Heilmann and Lieb.

Key words. eigenvalues, tree, interlacing

AMS subject classifications. 15A18, 15A57

DOI. 10.1137/S0895479804440359

1. Introduction. Several fields of applied mathematics have motivated the study of the behavior of eigenvalues of matrices of a special shape. Interest in the spectra of tridiagonal matrices has increased in recent years in different branches, e.g., numerical analysis, special functions, partial differential equations, and, naturally, linear algebra. Several fast and parallel algorithms for calculating these spectra have been designed (see, e.g., [1, 9, 12]).

Krishnakumar and Morf [12] established a divide-and-conquer approach for the solution of symmetric tridiagonal eigenvalue problems. This highly accurate method is based on a sequential doubling in the evaluation of the characteristic polynomial and a bisection method. In fact they consider partitioning a matrix and applying the three-term recurrence to the submatrices, yielding a recursion for which the order doubles at each step. A key feature of this approach is natural concurrency, which readily adapts to computing equipment with parallel architectures. In these circumstances, the computing time for all eigenvalues is $O(n \log n)$ instead of $O(n^2)$.

In 1992, Hill and Parlett [9] extended Cauchy's interlacing theorem and other interlacing results for eigenvalues of tridiagonal matrices and related them to the last component of eigenvectors. Later, Bar-On [1] generalized the main results of [9], giving new interlacing properties for eigenvalues of a tridiagonal symmetric matrix. Some relations relating the eigenvalues of a tridiagonal symmetric matrix to those of its leading and trailing submatrices were proved. These theoretical results have been generalized to specially structured symmetric matrices and applied to obtain fast and efficient parallel algorithms for locating the eigenvalues of matrices of very large order.

Kulkarni, Schmidt, and Tsui [13] considered the so-called tridiagonal pseudo-Toeplitz matrices: tridiagonal matrices which contain a Toeplitz matrix in the upper left block. They established a connection between the characteristic polynomials of these tridiagonal pseudo-Toeplitz matrices and the Chebyshev polynomials of the second kind, whereby one can locate the eigenvalues that fall in the intervals determined by the roots of some Chebyshev polynomials. They gave some special cases where the location of all eigenvalues can be determined. In fact, their results can be generalized to the broader class of Jacobi matrices.

*Received by the editors January 27, 2004; accepted for publication (in revised form) by R. Bhatia November 24, 2004; published electronically July 13, 2005. This work was supported by CMUC—Centro de Matemática da Universidade de Coimbra.

<http://www.siam.org/journals/simax/27-1/44035.html>

[†]Departamento de Matemática, Universidade de Coimbra, 3001-454 Coimbra, Portugal (cmf@mat.uc.pt).

Our aim here is to unify some important theoretical results which deal with the spectra of tridiagonal matrices, generalize them to matrices whose graph is a given tree, and bring the results together in one place. More precisely, we give the location of the eigenvalues of weighted trees in terms of spectra of an upper left block subtree and of a bottom right tridiagonal block. For this purpose, we derive some important identities for the characteristic polynomial of a weighted tree based on the work of Heilmann and Lieb [7], done in a different context. These authors, motivated by problems in quantum chemistry, established results which are the foundation of some recent developments on matchings polynomials (cf. [4, 5]). We conclude by mentioning some results on the multiplicity of the eigenvalues of weighted trees. Some examples are given.

2. Weighted graphs. A weighted finite graph G with n vertices is a pair (V, E) , where V is the vertex set $\langle n \rangle := \{1, \dots, n\}$ and E is the set $V \times V$; to each ordered pair $e = (i, j)$ we assign a complex number, the weight of e , and to (j, i) its conjugate. We say that the vertices i and j are adjacent, and write $i \sim j$, if (i, j) is an edge of G , with $i \neq j$. The symbol $\simeq$ means adjacent or equal. Throughout, we assume that the graph G has loops; i.e., (i, i) is an edge to which we assign a real number.

The (weighted adjacency) matrix of G is the Hermitian matrix of order n where the (i, j) -entry is the weight of the edge (i, j) .

Given an edge $e = (i, j)$ of G , $G \setminus e$ is obtained by deleting e but not the vertex i or j . In this case the matrix of $G \setminus e$ is equal to that of G , except for the (i, j) -entry and, by symmetry, the (j, i) -entry, which are replaced by zero.

If $A = (a_{ij})$ is a Hermitian matrix of order n , the (weighted) graph of A , $G = G(A)$, is determined entirely by the off-diagonal entries of A : the vertex set of G is $\langle n \rangle$ and i and j are adjacent if and only if $a_{ij} \neq 0$. If A is a 0-1 matrix, with main diagonal equal to zero, then A is the adjacency matrix of $G(A)$. We define

$$\mathcal{H}(G) = \{A \mid A = A^*, G(A) = G\}$$

as the set of all Hermitian matrices that share a common graph G . Since our study is dedicated to the eigenvalues of Hermitian matrices, we will also consider

$$\tilde{\mathcal{H}}(G) = \{DAD^{-1} \mid A \in \mathcal{H}(G), D \text{ is a scalar matrix}\},$$

i.e., the set of all matrices that are diagonally similar to Hermitian matrices whose graph is G .

A tree is a connected graph without cycles, and a (disconnected) forest is a graph of which each component is a tree.

3. The characteristic polynomial of a weighted tree. There are close connections between the theory of matchings in graphs and the characteristic polynomial of an adjacency matrix of a graph. Recall that the matchings polynomial of a graph G is $\mu(G, x) := \sum_k (-1)^k p(G, k) x^{n-2k}$, where $p(G, k)$ is the number of elements of a set of k disjoint edges, no two of which have a vertex in common.

Though the matchings polynomial of a graph has many interesting properties, the task of computing this polynomial for a given graph is hard. In general there is no easy way of computing $\mu(G, x)$. Thus the matchings polynomial is in this regard a more intractable object than the characteristic polynomial. Nonetheless, it is known that G is a forest if and only if both polynomials coincide, and there are simple recurrences that enable us to compute the matchings polynomials of small graphs with some facility. For example, the matchings polynomials of bipartite graphs

are essentially the same as “rook polynomials” (cf. [4]). An unexpected property of matchings polynomials is that all their zeros are real. In the paper [7] we can find three distinct proofs of this fact. Therefore if G is a tree, then all the eigenvalues of the adjacency matrix of G are real. Here we strengthen this result.

Given a graph G , the characteristic polynomial of $A(G)$,

$$\varphi(G, x) = \det(xI - A(G)),$$

is sometimes called the characteristic polynomial of G and simply denoted by $\varphi(G)$.

LEMMA 3.1. If F is a forest with components $T_1, \dots, T_\ell$, then $\varphi(F) = \varphi(T_1) \cdots \varphi(T_\ell)$.

Let us define $w_{ij}(A) = |a_{ij}|^2$ if $i \neq j$; otherwise, $w_{ii}(A) = a_{ii}$. Sometimes we abbreviate $w_{ij}(A)$ to w_{ij} . The next result provides a general recurrence relation between different characteristic polynomials.

LEMMA 3.2. Let $e = (i, j)$ be an edge of a tree T .

$$(3.1) \quad \varphi(T, x) = \varphi(T \setminus e, x) - w_{ij} \varphi(T \setminus ij, x).$$

Let E_{ij} be the matrix with ij -entry equal to 1, and all other entries equal to zero. Denote by E the sum $a_{ij}E_{ij} + \bar{a}_{ij}E_{ji}$. Notice that

$$A(T) = A(T \setminus e) + E.$$

Since the determinant is a multilinear function on the columns and T is a tree, we get (3.1). $\square$

THEOREM 3.3. Let i be a vertex of a tree T .

$$(3.2) \quad \varphi(T, x) = (x - w_{ii})\varphi(T \setminus i, x) - \sum_{k \sim i} w_{ki} \varphi(T \setminus ki, x).$$

The equality (3.2) can be derived by iterating the formula (3.1). $\square$

A routine induction argument, based on (3.2), gives us an expression for the derivative of the characteristic polynomial.

COROLLARY 3.4. Let T be a tree.

$$\varphi'(T, x) = \sum_{i \in V(T)} \varphi(T \setminus i, x).$$

We now state a result which we will use often. The general interlacing theorem between the eigenvalues of a Hermitian matrix and one of its principal submatrices is well known in the literature (e.g., [10]). It follows from the interlacing theorem.

THEOREM 3.5. Let T be a tree with n vertices. Let $A \in \mathcal{H}(T)$ be a Hermitian matrix with $A(T)$ as its principal submatrix of order $n-1$.

$$\lambda_1 \leq \lambda_2 \leq \cdots \leq \lambda_n.$$

Let i be a vertex of T . Let $\mu_1 \leq \mu_2 \leq \cdots \leq \mu_{n-1}$ be the eigenvalues of $A(T \setminus i)$.

$$\lambda_1 \leq \mu_1 \leq \lambda_2 \leq \mu_2 \leq \cdots \leq \mu_{n-1} \leq \lambda_n;$$

where $A(T)$ and $A(T \setminus i)$ are the principal submatrices of A of order n and $n-1$ respectively.

Theorem 3.5 has a well-known corollary for tridiagonal matrices already proved elsewhere (cf. [13]).

COROLLARY 3.6. Let P be a path with n vertices. Let $A \in \mathcal{H}(P)$ be a Hermitian matrix with A as its principal submatrix of order $n-1$.

4. Christoffel–Darboux identity. As we already pointed out, the work of Heilmann and Lieb [7] has had some important implications in the theory of matchings polynomials with a close connection to orthogonal polynomials (for more details, see, e.g., [2]). Here we derive some important identities for characteristic polynomials of matrices in the set $\mathcal{H}(T)$.

Let P_{ij} denote the (unique) path in a tree joining vertex i to j . Given a path P in T with more than one vertex, let us define $W(P) = \prod_{(i,j) \in P} w_{ij}(P)$, where the product is taken over the edges (i, j) of P , with $i < j$.

THEOREM 4.1 (Christoffel–Darboux identity). *Let T be a tree with n vertices. Then for any vertex $i \in V(T)$*

$$(4.1) \quad \varphi(T, x)\varphi(T \setminus i, y) - \varphi(T, y)\varphi(T \setminus i, x) = (x - y) \sum_{j=1}^n W(P_{ij})\varphi(T \setminus P_{ij}, x)\varphi(T \setminus P_{ij}, y).$$

For trees with one or two vertices the result is trivial. From (3.2), we have the equations

$$x\varphi(T \setminus i, x)\varphi(T \setminus i, y) = \varphi(T, x)\varphi(T \setminus i, y) + \sum_{k \sim i} w_{ki}\varphi(T \setminus ki, x)\varphi(T \setminus i, y)$$

and

$$y\varphi(T \setminus i, y)\varphi(T \setminus i, x) = \varphi(T, y)\varphi(T \setminus i, x) + \sum_{k \sim i} w_{ki}\varphi(T \setminus ki, y)\varphi(T \setminus i, x).$$

Subtracting the second equation from the first, we get

$$\begin{aligned} (x - y)\varphi(T \setminus i, y)\varphi(T \setminus i, x) &= \varphi(T, x)\varphi(T \setminus i, y) - \varphi(T, y)\varphi(T \setminus i, x) \\ &\quad - \sum_{k \sim i} w_{ki} [\varphi(T \setminus i, x)\varphi(T \setminus ki, y) - \varphi(T \setminus i, y)\varphi(T \setminus ki, x)]. \end{aligned}$$

Applying the hypothesis on $\varphi(T \setminus i, x)\varphi(T \setminus ki, y) - \varphi(T \setminus i, y)\varphi(T \setminus ki, x)$, with the convention $\varphi(\emptyset, x) = 1$, we get the result. $\square$

COROLLARY 4.2. *Let T be a tree with n vertices. Then for any vertex $i \in V(T)$*

$$(4.2) \quad \varphi'(T, x)\varphi(T \setminus i, x) - \varphi(T, x)\varphi'(T \setminus i, x) = \sum_{j=1}^n W(P_{ij})\varphi(T \setminus P_{ij}, x)^2.$$

Letting $y \rightarrow x$ in (4.1), we get (4.2) since we may write

$$\begin{aligned} &\varphi(T, x)\varphi(T \setminus i, y) - \varphi(T, y)\varphi(T \setminus i, x) \\ &= [\varphi(T, x) - \varphi(T, y)]\varphi(T \setminus i, y) - [\varphi(T \setminus i, x) - \varphi(T \setminus i, y)]\varphi(T, x). \quad \square \end{aligned}$$

If we consider the sum over all vertices of T in (4.2), then from Corollary 3.4 we get the following.

COROLLARY 4.3.

$$(4.3) \quad \varphi'(T, x)^2 - \varphi''(T, x)\varphi(T, x) = \sum_{i,j=1}^n W(P_{ij})\varphi(T \setminus P_{ij}, x)^2.$$

THEOREM 4.4. Let T be a tree with n vertices and let $i, j \in V(T)$

$$(4.4) \quad \varphi(T \setminus i, x) \varphi(T \setminus j, x) - \varphi(T \setminus ij, x) \varphi(T, x) = W(P_{ij}) \varphi(T \setminus P_{ij}, x)^2.$$

Once again the cases $n = 1$ and $n = 2$ are trivial. We use induction on n as in the propositions before. For that we only have to consider (3.2) and

$$\varphi(T \setminus j, x) = (x - w_{ii}) \varphi(T \setminus ij, x) - \sum_{k \sim i} w_{ki} \varphi(T \setminus kij, x). \quad \square$$

5. Location of the eigenvalues of weighted trees. When we consider the Jacobi matrix

$$J_{n+1} := \begin{pmatrix} \alpha_0 & \beta_0 & & & \\ \gamma_1 & \alpha_1 & \beta_1 & & \\ & \gamma_2 & \ddots & \ddots & \\ & & \ddots & \ddots & \beta_{n-1} \\ & & & \gamma_n & \alpha_n \end{pmatrix},$$

its off-diagonal entries enter into its characteristic equation only as products $\beta_i \gamma_{i+1}$. Thus it suffices to consider only matrices that have superdiagonal entries normalized to 1, i.e.,

$$(5.1) \quad \tilde{J}_{n+1} := \begin{pmatrix} \alpha_0 & 1 & & & \\ \gamma_1 & \alpha_1 & 1 & & \\ & \gamma_2 & \ddots & \ddots & \\ & & \ddots & \ddots & 1 \\ & & & \gamma_n & \alpha_n \end{pmatrix}.$$

From Corollary 3.6 we conclude the well-known result that J_n , and therefore $\tilde{J}_n$, with $\gamma_i > 0$, have n distinct real eigenvalues. Note that $\tilde{J}_n \in \mathcal{H}(P)$, where P is a path on n vertices. In particular, when we consider tridiagonal Toeplitz matrices, i.e., making α_i 's and γ_i 's all equal to α and γ , respectively, the characteristic polynomial of J_n is $p_n(x) = (\sqrt{\gamma})^n U_n(\frac{x-\alpha}{2\sqrt{\gamma}})$, where U_n is the Chebyshev polynomial of second kind of degree n , which satisfies the three-term recurrence relations

$$xU_n(x) = U_{n+1}(x) + U_{n-1}(x)$$

for all $n = 1, 2, \dots$, with initial conditions $U_0(x) = 1$ and $U_1(x) = 2x$. It is well known (see, e.g., [2]) that each U_n also satisfies

$$U_n(x) = \frac{\sin(n+1)\theta}{\sin \theta}, \quad x = \cos \theta \quad (0 \leq \theta < \pi)$$

for all $n = 0, 1, 2, \dots$, from which one easily deduces the orthogonality relations

$$\int_{-1}^1 U_n(x) U_m(x) \sqrt{1-x^2} dx = \frac{\pi}{2} \delta_{n,m}.$$

LEMMA 5.1. $\alpha_i = \alpha$ and $\gamma_i = \gamma$ for $i = 1, \dots, n$. (5.1)

$$\lambda_\ell = \alpha - 2\sqrt{\gamma} \cos\left(\frac{\ell\pi}{n+1}\right)$$

for $\ell = 1, 2, \dots, n$.

We note that if $\gamma < 0$, then $p_n(x) = (i\sqrt{-\gamma})^n U_n(\frac{x-\alpha}{2i\sqrt{-\gamma}})$, where $i = \sqrt{-1}$.

Given a weighted tree T on $m+n$ vertices, let $A^{m,n}$ be the matrix diagonally similar to the Hermitian matrix $A(T)$ partitioned in the following way:

$$(5.2) \quad A^{m,n} = \left(\begin{array}{c|cccc} A(T^m) & & & & \\ \hline & 1 & & & \\ & \alpha_1 & 1 & & \\ & \gamma_2 & & \ddots & \\ & & \ddots & \ddots & 1 \\ & & & \gamma_n & \alpha_n \end{array} \right),$$

where T^m is a subtree of T . We assume that the tridiagonal right block is unreduced, with $\gamma_i > 0$. Using the Laplace expansion, we get

$$(5.3) \quad \varphi(T, x) = \varphi(T^m, x)\varphi(P^n, x) - \gamma_1 \varphi(T^m \setminus m, x)\varphi(P^n \setminus m+1, x),$$

where P^n is the path in T on the n vertices $\{m+1, \dots, m+n\}$. Suppose θ is an eigenvalue of $A^{m,n}$. If θ is not a common zero of $\varphi(T^m \setminus m, x)$ and $\varphi(P^n, x)$, then

$$(5.4) \quad \frac{\varphi(T^m, \theta)}{\varphi(T^m \setminus m, \theta)} = \gamma_1 \frac{\varphi(P^n \setminus m+1, \theta)}{\varphi(P^n, \theta)}.$$

Notice that given any weighted tree, if we define

$$p(x) := \frac{\varphi(T, x)}{\varphi(T \setminus i, x)},$$

then

$$p'(x) := \frac{\varphi'(T, x)\varphi(T \setminus i, x) - \varphi(T, x)\varphi'(T \setminus i, x)}{(\varphi(T \setminus i, x))^2},$$

which is positive by (4.2), i.e., $p(x)$ is strictly increasing. If we denote

$$f(x) = \frac{\varphi(T^m, x)}{\varphi(T^m \setminus m, x)} \quad \text{and} \quad g(x) = \gamma_1 \frac{\varphi(P^n \setminus m+1, x)}{\varphi(P^n, x)},$$

then $f(x)$ is strictly increasing and $g(x)$ is strictly decreasing.

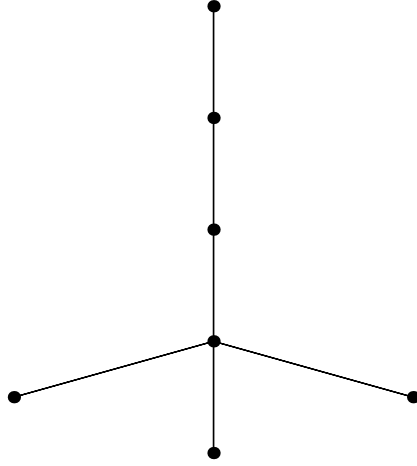
A similar analysis can be easily done if instead of the right Jacobi block in (5.2) we consider $A(T^n)$ for a more general tree T^n on n vertices.

Consider the extended sets of eigenvalues of $A(T^m \setminus m)$,

$$\eta = \{-\infty = \eta_0 < \eta_1 \leq \dots \leq \eta_{m-1} < \eta_m = +\infty\},$$

$$-\infty < \xi_1 \leq \eta_1 \leq \cdots \leq \eta_{m-2} \leq \xi_{m-1} \leq \eta_{m-1} \leq \xi_m < +\infty$$
$$-\infty < \zeta_1 < \rho_1 < \cdots < \rho_{n-2} < \zeta_{n-1} < \rho_{n-1} < \zeta_n < +\infty.$$

In fact, this result generalizes some other examples given, for example, in [9, 13].

FIG. 1. *Graph of A.*

Consider now the matrix

$$A = \left(\begin{array}{cccc|ccc} 1 & 0 & 0 & i & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & -2 & -1 & 0 & 0 & 0 \\ -i & 1 & -1 & 1 & -\sqrt{2} & 0 & 0 \\ \hline 0 & 0 & 0 & -\sqrt{2} & 2 & -1+i & 0 \\ 0 & 0 & 0 & 0 & -1-i & -1 & -2 \\ 0 & 0 & 0 & 0 & 0 & -2 & 1 \end{array} \right).$$

The graph of A is shown in Figure 1.

The eigenvalues of

$$\begin{pmatrix} 1 & 0 & 0 & i \\ 0 & 0 & 0 & 1 \\ 0 & 0 & -2 & -1 \\ -i & 1 & -1 & 1 \end{pmatrix}$$

are

$$\xi_1 = -2.376078978,$$

$$\xi_2 = -0.595187944,$$

$$\xi_3 = 0.595187944,$$

$$\xi_4 = 2.376078978,$$

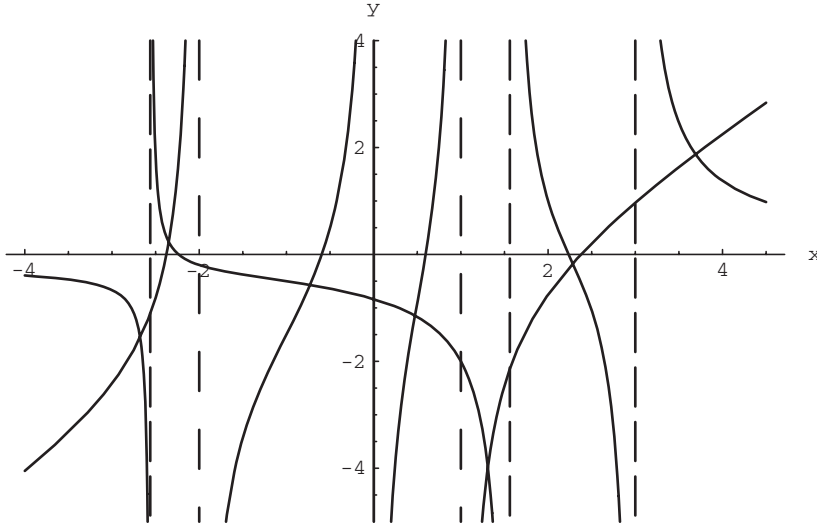
and the eigenvalues of

$$\begin{pmatrix} -1 & -2 \\ -2 & 1 \end{pmatrix}$$

are

$$\rho_1 = -2.236067977,$$

$$\rho_2 = 2.236067977.$$

FIG. 2. Intersection of graphics of $f(x)$ and of $g(x)$.

Notice that

$$-\infty < \xi_1 < \rho_1 < \xi_2 < \xi_3 < \rho_2 < \xi_4 < +\infty.$$

The intersection of graphics of $f(x)$ and of $g(x)$ is shown in Figure 2. Finally, if we evaluate directly the eigenvalues of A , we get

$$\begin{aligned}\theta_1 &= -2.677387677, \\ \theta_2 &= -2.349613205, \\ \theta_3 &= -0.730287931, \\ \theta_4 &= 0.473357313, \\ \theta_5 &= 1.309649492, \\ \theta_6 &= 2.282754449, \\ \theta_7 &= 3.691527560.\end{aligned}$$

7. Multiplicities of eigenvalues. For any square matrix A , let $m_A(\theta)$ denote the (algebraic) multiplicity of θ as an eigenvalue of A . The interlacing established in Theorem 3.5 has a straightforward consequence for $A \in \mathcal{H}(T)$, where T is a tree:

$$|m_{A(T \setminus i)}(\theta) - m_{A(T)}(\theta)| \leq 1.$$

Suppose $m_{A(T)}(\theta) > 0$. A vertex i of T is θ -essential and θ -positive if $m_{A(T \setminus i)}(\theta) = m_{A(T)}(\theta) - 1$ and $m_{A(T \setminus i)}(\theta) = m_{A(T)}(\theta) + 1$, respectively. If θ is understood, we shall omit the mention of θ .

Note that each $A \in \mathcal{H}(T)$ has at least one essential vertex. Indeed, the multiplicity of θ as zero of $\varphi'(T, x)$ is $m_{A(T)}(\theta) - 1$. If $m_{A(T \setminus i)}(\theta) \geq m_{A(T)}(\theta)$ for all vertices i in T , then by Corollary 3.4 the multiplicity of θ as a zero of $\varphi'(T, x)$ is at least $m_{A(T)}(\theta)$.

The next theorem states that if P is a path in a tree T , then $\varphi(T \setminus P, x)/\varphi(T, x)$ has only simple poles.

THEOREM 7.1. Let P be a path in T and let $A \in \mathcal{H}(T)$. Then θ is a zero of $\varphi(T \setminus P, x)$ if and only if θ is a zero of $\varphi(T, x)$ with multiplicity at least $m_A(\theta) - 1$.

$$(7.1) \quad m_{A(T \setminus P)}(\theta) \geq m_{A(T)}(\theta) - 1.$$

Suppose θ is an eigenvalue of $A(T)$ with $m_A(\theta) > 1$. Then θ is a zero of $\varphi'(T, x)^2 - \varphi''(T, x)\varphi(T, x)$ with multiplicity at least $2m_A(\theta) - 2$. From (4.3), θ is a zero of nonnegative summation $\sum_{i,j=1}^n W(P_{ij})\varphi(T \setminus P_{ij}, x)^2$, and therefore θ has multiplicity for each $\varphi(T \setminus P_{ij}, x)$ of at least $m_A(\theta) - 1$. $\square$

In general, we say that a path P is P -essential if $m_{A(T \setminus P)}(\theta) = m_{A(T)}(\theta) - 1$. Given any θ -essential vertex i , with $\theta \neq a_{ii}$, there is an adjacent vertex j , such that the path ij is essential. For, if $m_{A(T \setminus ik)}(\theta) \geq m_{A(T)}(\theta)$ for all $k \sim i$, then by (3.2), $m_{A(T \setminus P)}(\theta) \geq m_{A(T)}(\theta)$, which is a contradiction.

We also point out that if a path P_{ij} is essential for some vertex j , then i is an essential vertex. In fact, suppose that $m_{A(T \setminus i)}(\theta) \geq m_{A(T)}(\theta)$. For any $j (\neq i)$, the multiplicity of θ as a zero of $\varphi(T \setminus i, x)\varphi(T \setminus j, x) - \varphi(T \setminus ij, x)\varphi(T, x)$ is at least $2m_{A(T)}(\theta) - 1$. It is in fact, by (4.4), at least $2m_{A(T)}(\theta)$, and therefore $m_{A(T \setminus P_{ij})}(\theta) \geq m_{A(T)}(\theta)$.

There are some recent results on the multiplicities of eigenvalues of a tree inspired by matching theory (cf. [4]). For example, Johnson and Leal Duarte [11] proved that the maximum multiplicity of any single eigenvalue among all matrices in $\mathcal{H}(T)$ is equal to the smallest number of vertex disjoint paths of T that cover all the vertices of T . Later, they also proved that the number of distinct eigenvalues of a Hermitian matrix whose graph is the tree T is at least the number of vertices in a longest path in T (cf. [14]).

LEMMA 7.2. Let T be a tree and let i be a vertex of T . Then i is θ -positive if and only if $m_{A(T \setminus ij)}(\theta) > m_{A(T)}(\theta)$ for all $j \sim i$.

Suppose i is θ -positive and $m_{A(T \setminus ij)}(\theta) > m_{A(T)}(\theta)$ for all $j \sim i$. Then (3.2) leads to a contradiction. Conversely, by a remark of the last section, the path ij , for any j , is not essential and, consequently, $m_{A(T \setminus ij)}(\theta) \geq m_{A(T)}(\theta)$. Suppose that $j \sim i$ is essential in $T \setminus i$. Then $m_{A(T \setminus ij)}(\theta) = m_{A(T \setminus i)}(\theta) - 1$. Hence i is positive. $\square$

LEMMA 7.3. Let i and j be adjacent vertices of T . Then $\varphi(T \setminus i, x) - \varphi(T \setminus ij, x) = \varphi(T, x) - \varphi(T \setminus i, x)$ if and only if i is θ -positive.

Suppose θ is a common zero of $\varphi(T \setminus i, x)$ and $\varphi(T \setminus ij, x)$. By Theorem 7.1, if $m_{A(T)}(\theta) > 1$, then θ is a common zero of $\varphi(T \setminus i, x)$ and $\varphi(T \setminus ij, x)$. If $m_{A(T \setminus i)}(\theta) > 1$, by interlacing we have $m_{A(T \setminus i)}(\theta) > 0$. Therefore, $m_{A(T)}(\theta) = m_{A(T \setminus i)}(\theta) = 1$. From Lemma 7.2, there is no adjacent vertex to i essential in $T \setminus i$. Thus $m_{A(T \setminus ij)}(\theta) \geq m_{A(T)}(\theta) = 1$, and θ is a zero of $\varphi(T \setminus ij, x)$, which is a contradiction. $\square$

Under the conditions of the above lemma, by interlacing, we conclude that $\varphi(T, x)$ and $\varphi(T \setminus i, x)$ have only simple zeros. By induction we also have the following corollary.

COROLLARY 7.4. Let H be a subtree of T and let P be a path in H . Then θ is a zero of $\varphi(T, x)$ if and only if θ is a zero of $\varphi(T \setminus i, x)$ with multiplicity at least $m_A(\theta) - 1$.

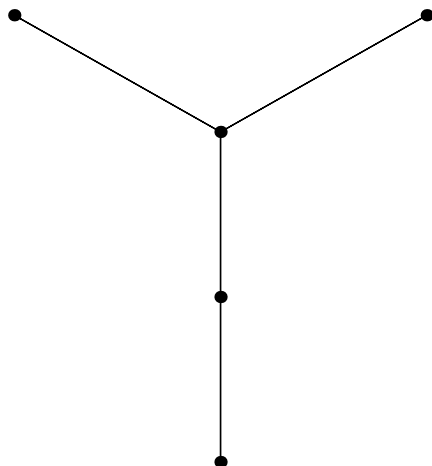


FIG. 3. A tree with 5 vertices.

Consider the matrix

$$\begin{pmatrix} 1 & 0 & i & 0 & 0 \\ 0 & 1 & -1 & 0 & 0 \\ -i & -1 & -2 & 1 & 0 \\ 0 & 0 & 1 & 2 & \sqrt{2} \\ 0 & 0 & 0 & -\sqrt{2} & 3 \end{pmatrix},$$

whose graph is the tree depicted in Figure 3.

The matrix has eigenvalue 1 with multiplicity 2. Therefore

$$\begin{pmatrix} 1 & 0 & i \\ 0 & 1 & -1 \\ -i & -1 & -2 \end{pmatrix} \quad \text{and} \quad \begin{pmatrix} 1 & 0 \\ 0 & 1 \end{pmatrix}$$

have at least one common eigenvalue.

REFERENCES

- [1] I. BAR-ON, *Interlacing properties of tridiagonal symmetric matrices with applications to parallel computing*, SIAM J. Matrix Anal. Appl., 17 (1996), pp. 548–562.
- [2] T. S. CHIHARA, *An Introduction to Orthogonal Polynomials*, Gordon and Breach, New York, 1978.
- [3] J. W. DEMMEL AND W. B. GRAGG, *On computing accurate singular values and eigenvalues of matrices with acyclic graphs*, Linear Algebra Appl., 185 (1993), pp. 203–218.
- [4] C. D. GODSIL, *Algebraic Combinatorics*, Chapman and Hall, New York, London, 1993.
- [5] C. D. GODSIL, *Algebraic matching theory*, Electron. J. Combin., 2 (1995), Research Paper 8.
- [6] F. HARARY, *The determinant of the adjacency matrix of a graph*, SIAM Rev., 4 (1962), pp. 202–210.
- [7] O. L. HEILMANN AND E. H. LIEB, *Theory of monomer-dimer systems*, Comm. Math. Phys., 25 (1972), pp. 190–232.
- [8] G. HEINIG AND K. ROST, *Algebraic Methods for Toeplitz-like Matrices and Operators*, Oper. Theory Adv. Appl. 13, Birkhäuser-Verlag, Basel, 1984.
- [9] R. O. HILL AND B. N. PARLETT, *Refined interlacing properties*, SIAM J. Matrix Anal. Appl., 13 (1992), pp. 239–247.

- [10] R. A. HORN AND C. R. JOHNSON, *Matrix Analysis*, Cambridge University Press, New York, 1985.
- [11] C. R. JOHNSON AND A. LEAL DUARTE, *The maximum multiplicity of an eigenvalue in a matrix whose graph is a tree*, Linear and Multilinear Algebra, 46 (1999), pp. 139–144.
- [12] A. KRISHNAKUMAR AND M. MORF, *Eigenvalues of a symmetric tridiagonal matrix: A divide-and-conquer approach*, Numer. Math., 48 (1986), pp. 349–368.
- [13] D. KULKARNI, D. SCHMIDT, AND S.-K. TSUI, *Eigenvalues of tridiagonal pseudo-Toeplitz matrices*, Linear Algebra Appl., 297 (1999), pp. 63–80.
- [14] A. LEAL DUARTE AND C. R. JOHNSON, *On the minimum number of distinct eigenvalues for a symmetric matrix whose graph is a given tree*, Math. Inequal. Appl., 5 (2002), pp. 175–180.
- [15] J. W. LEWIS, *Inversion of tridiagonal matrices*, Numer. Math., 38 (1982), pp. 333–345.
- [16] G. MEURANT, *A review on the inverse of symmetric tridiagonal and block tridiagonal matrices*, SIAM J. Matrix Anal. Appl., 13 (1992), pp. 707–728.
- [17] B. N. PARLETT, *The Symmetric Eigenvalue Problem*, Prentice-Hall, Englewood Cliffs, NJ, 1980.
- [18] S. PARTER, *On the eigenvalues and eigenvectors of a class of matrices*, J. Soc. Indust. Appl. Math., 8 (1960), pp. 376–388.
- [19] A. RONVEAUX AND W. VAN ASSCHE, *Upward of the Jacobi matrix for orthogonal polynomials*, J. Approx. Theory, 86 (1996), pp. 335–357.
- [20] G. WIENER, *Spectral multiplicity and splitting results for a class of qualitative matrices*, Linear Algebra Appl., 61 (1984), pp. 15–29.
- [21] J. H. WILKINSON, *The Algebraic Eigenvalue Problem*, Oxford University Press, London, 1965.

THE ACCURATE AND EFFICIENT SOLUTION OF A TOTALLY POSITIVE GENERALIZED VANDERMONDE LINEAR SYSTEM*

JAMES DEMMEL[†] AND PLAMEN KOEV[‡]

Abstract. Vandermonde, Cauchy, and Cauchy–Vandermonde totally positive linear systems can be solved extremely accurately in $O(n^2)$ time using Björck–Pereyra-type methods. We prove that Björck–Pereyra-type methods exist not only for the above linear systems but also for *any* totally positive linear system as long as the initial minors (i.e., contiguous minors that include the first row or column) can be computed accurately. Using this result we design a new $O(n^2)$ Björck–Pereyra-type method for solving generalized Vandermonde systems of equations by using a new algorithm for computing the Schur function. We present explicit formulas for the entries of the bidiagonal decomposition, the LDU decomposition, and the inverse of a totally positive generalized Vandermonde matrix, as well as algorithms for computing these entries to high relative accuracy.

Key words. high relative accuracy, generalized Vandermonde matrix, total positivity, bidiagonal decomposition, Schur function

AMS subject classifications. 65F05, 15A23

DOI. 10.1137/S0895479804440335

1. Introduction. We consider the class of (a_i, x_i) (TP) matrices, i.e., matrices with all minors positive, and in particular the TP₁ matrices defined as

$$(1.1) \quad G = \begin{bmatrix} x_1^{a_1} & x_1^{a_2} & \cdots & x_1^{a_n} \\ x_2^{a_1} & x_2^{a_2} & \cdots & x_2^{a_n} \\ \vdots & \vdots & \ddots & \vdots \\ x_n^{a_1} & x_n^{a_2} & \cdots & x_n^{a_n} \end{bmatrix},$$

where $0 \leq a_1 < a_2 < \cdots < a_n$ are integers and $0 < x_1 < x_2 < \cdots < x_n$ [12, p. 76].

We define the λ associated with G as the nonincreasing sequence of nonnegative integers

$$\lambda = (\lambda_1, \lambda_2, \dots, \lambda_n) = (a_n - (n-1), a_{n-1} - (n-2), \dots, a_1).$$

In terms of λ_i ,

$$G = [x_i^{j-1+\lambda_{n-j+1}}]_{1 \leq i, j \leq n}.$$

Throughout this paper G will denote this particular TP generalized Vandermonde matrix and $V = [x_i^{j-1}]_{1 \leq i, j \leq n}$ will denote the ordinary Vandermonde matrix. The latter corresponds to partition $\lambda = (0)$.

Our goal is to perform accurate and efficient matrix computations with G —we consider bidiagonal decomposition, linear equation solving, triangular decomposition, and inversion. Our inspiration comes from other algorithms for Vandermonde,

*Received by the editors January 29, 2004; accepted for publication (in revised form) by H. J. Woerdeman November 19, 2004; published electronically July 13, 2005.

<http://www.siam.org/journals/simax/27-1/44033.html>

[†]Computer Science Division and Mathematics Department, University of California, Berkeley, CA 94720 (demmelm@cs.berkeley.edu).

[‡]Department of Mathematics, Massachusetts Institute of Technology, Cambridge, MA 02139 (plamen@math.mit.edu).

Cauchy, and Cauchy–Vandermonde matrices [4, 18, 25] which use a method introduced by Björck and Pereyra [3] to

1. solve $n \times n$ linear systems $Ax = b$ in $O(n^2)$ time, and
2. be arbitrarily more accurate than the usual condition number $\kappa(A) \equiv \|A\|_2 \cdot \|A^{-1}\|_2$ would suggest.

Such accurate computation is justified, because in fact the condition number of these matrices, i.e., the sensitivity of the solution with respect to the parameters defining the matrices, can be arbitrarily smaller than $\kappa(A)$. So these accurate computations do in fact compute the answer as accurately as the data deserve.

We extend these earlier results as follows.

First, we observe that accurate and efficient Björck–Pereyra-type methods exist for TP matrix that permits accurate and efficient evaluation of its minors. Initial minors are minors of contiguous sets of rows and columns and include at least the first row and/or the first column [16]. This is based on an earlier result of Gasca and Peña: the entries of the bidiagonal factors of a matrix inverse arising from products and quotients of initial minors and so can be computed accurately and efficiently if these minors can be [16, 25].

Next, we must consider the initial minors of generalized Vandermonde matrices. It is well known that

$$(1.2) \quad \det G = s_\lambda(x_1, x_2, \dots, x_n) \cdot \det V = s_\lambda(x_1, x_2, \dots, x_n) \cdot \prod_{i>j} (x_i - x_j)$$

is the product of the $s_\lambda(x_1, x_2, \dots, x_n)$ and $\det V = \prod_{i>j} (x_i - x_j)$ [23]. The other minors of G have analogous forms.

Our second and most important contribution is the application of our new fast and accurate algorithm for evaluating Schur functions [7] to accurately compute all the needed minors of G in order to obtain

- an accurate bidiagonal decomposition of G^{-1} (section 3);
- a new Björck–Pereyra-type algorithm for solving $Gy = b$ (section 4);
- an accurate G^{-1} (section 5);
- an accurate LDU decomposition of G (section 6).

These computations can be performed in time that grows like a low order polynomial in n and $|\lambda| \equiv \sum_i \lambda_i$.

The accuracy of our algorithms is due to the fact that they can suffer strictly bounded or no subtractive cancellation (depending on the choice of algorithm from [7] for computing the Schur function accurately—see section 3). This means all rounding errors accumulate slowly, and the final computed values are quite accurate, yielding very small error bounds (section 7).

In contrast, previous algorithms for generalized Vandermonde matrices may be arbitrarily less accurate than ours, or may require arbitrarily high precision arithmetic to achieve the same accuracy. We compare the accuracy of our new Björck–Pereyra-type algorithm to that of the traditional algorithms in section 8.

In what follows we will use MATLAB [26] notation for vectors and minors of a matrix. For example, the submatrix of A consisting of rows i through j and columns k through l will be denoted as $A(i : j, k : l)$ or, equivalently, $A_{i:j, k:l}$. The submatrix of A consisting of rows i through j and columns $k, k+1, \dots, l$ and s will be denoted as $A(i : j, [k : l, s])$. Similarly, $x_{1:n}$ will mean the vector $(x_1, x_2, \dots, x_n)$, etc.

2. Related work. Björck and Pereyra proposed an $O(n^2)$ method for solving a Vandermonde linear system $Vy = b$ in their landmark paper [3]. Their idea was to

use Newton interpolation, which written in matrix form is simply a decomposition

$$(2.1) \quad V^{-1} = U^{(1)}U^{(2)} \dots U^{(n-1)}L^{(n-1)}L^{(n-2)} \dots L^{(1)}$$

of V^{-1} as a product of upper ($U^{(i)}$) and lower ($L^{(i)}$) bidiagonal matrices. Björck and Pereyra noticed that their method sometimes produced solutions with surprising accuracy despite the ill conditioning of V . Higham [17, 18] showed that when the nodes x_i are positive and sorted (i.e., when V is TP) no subtractive cancellation occurs in forming the solution if b has an alternating sign pattern.¹ This is a consequence of the fact that in floating point arithmetic the relative accuracy is preserved when multiplying, dividing, or adding quantities with the same sign, or forming $x_i \pm x_j$, where x_i and x_j are exact initial data. In fact all entries of $L^{(i)}$ and $U^{(i)}$ are either 0, 1, $-x_i$, or $1/(x_i - x_j)$, $i > j$ [18].

Björck–Pereyra-type methods were also derived for Cauchy [4] and Cauchy–Vandermonde matrices [25].

Explicit formulas for the entries of the LU decomposition of an ordinary Vandermonde matrix were obtained in [28] and for the entries of the LU decomposition and the inverse of a generalized Vandermonde matrix in [29].

The total positivity of a matrix A is not revealed by its entries—one would need to verify that the n^2 initial minors are positive to conclude that all 4^n minors are and thus A is TP [10]. Yet to design structure-exploiting algorithms, we first look for a structure-revealing representation. It turns out that the TP structure is revealed by the entries of the bidiagonal decomposition of A (or equivalently the bidiagonal decomposition of A^{-1}). This bidiagonal decomposition is obtained from a process called *Newton interpolation* [13, 15, 16], which reduces A to upper triangular form using only n rows for elimination (unlike Gaussian elimination, which uses a fixed pivot row). The entries of the bidiagonal factors in this decomposition intrinsically parameterize the set of TP matrices [2, 8, 9, 10, 16]. They determine the entries of the inverse [17], the LU decomposition, the solution (if the right-hand side has an alternating sign pattern) [17], the eigenvalues, and the SVDs [22] to high relative accuracy.

The numerical solution of a generalized Vandermonde linear system was discussed in [29]. In that paper Van de Vel shows how to compute an LU decomposition of a TP generalized Vandermonde matrix G without suffering any subtractive cancellation, thus computing each entry of L and U to high relative accuracy. He says, “The Björck–Pereyra modification is essentially based on Newton’s formula for polynomial interpolation, and hence is not applicable, since we work with a set of powers² $j_1, \dots, j_n$, which are not ‘in line’.” We demonstrate that Björck–Pereyra-type methods are in fact applicable to G . To derive those, one must use the bidiagonal decomposition of G^{-1} and not Newton’s method. Since the Björck–Pereyra-type methods yield more accurate solutions than LU-based algorithms (see section 7), this paper represents a substantial advance over the results in [29].

3. Bidiagonal decomposition of G^{-1} . In this section we present explicit formulas for the decomposition of G^{-1} as the product of bidiagonal matrices and show how the entries of the latter can be computed efficiently to high relative accuracy. This is perhaps the most important decomposition of G from a numerical point of view:

¹The preservation of the relative accuracy in the process of Newton interpolation with positive and increasing interpolation nodes was, however, first noticed by Kahan and Farkas in 1963 [19].

²We call these $a_1, \dots, a_n$ in (1.1).

- It reveals the total positivity of G [16].
- It yields a Björck–Pereyra-type algorithm for the solution to the TP generalized Vandermonde linear system $Gy = b$ (section 4).
- It allows for the computation of the eigenvalues and SVDs of G to high relative accuracy in $O(n^3)$ additional time [22].

We start with a result due to Gasca and Peña [13, Lemma 2.6(1)].

PROPOSITION 3.1. *Let $A = (A_{i,j})_{i,j=1}^n$ be a nonsingular $n \times n$ matrix.*

$$(3.1) \quad A^{-1} = U^{(1)}U^{(2)} \dots U^{(n-1)}D^{-1}L^{(n-1)}L^{(n-2)} \dots L^{(1)},$$

$$\begin{aligned} & \text{where } L^{(k)} = (L_{i+1,i}^{(k)})_{i=1}^n, \quad U^{(k)} = (U_{i,i+1}^{(k)})_{i=1}^n, \quad k = 1, 2, \dots, n-1, \\ & \text{and } L_{i+1,i}^{(k)} = U_{i,i+1}^{(k)} = 0 \text{ for } i < k, \quad D = (D_{ii})_{i=1}^n. \end{aligned}$$

$$(3.2) \quad L_{i+1,i}^{(k)} = -\frac{\det A_{i-k+2:i+1,1:k}}{\det A_{i-k+2:i,1:k-1}} \cdot \frac{\det A_{i-k+1:i-1,1:k-1}}{\det A_{i-k+1:i,1:k}} \quad i \geq k;$$

$$(3.3) \quad U_{i,i+1}^{(k)} = -\frac{\det A_{1:k,i-k+2:i+1}}{\det A_{1:k-1,i-k+2:i}} \cdot \frac{\det A_{1:k-1,i-k+1:i-1}}{\det A_{1:k,i-k+1:i}} \quad i \geq k;$$

$$(3.4) \quad D_{ii} = \frac{\det A_{1:i,1:i}}{\det A_{1:i-1,1:i-1}}.$$

We now apply Proposition 3.1 to G . Let $i < j, k < l \in \{1, 2, \dots, n\}$ be such that $j - i = l - k$. Then from (1.2)

$$\begin{aligned} \det(G_{i:j,k:l}) &= \det(x_s^{r-1+\lambda_{n-r+1}})_{i \leq s \leq j, k \leq r \leq l} \\ &= \det(x_s^{r-1+\lambda_{n-r+1}})_{i \leq s \leq j, 1 \leq r \leq l-k+1} \cdot \prod_{s=i}^j x_s^{k-1} \\ &= \det(V_{i:j,1:l-k+1}) \cdot s_{(\lambda_{n-l+1}, \lambda_{n-l+2}, \dots, \lambda_{n-k+1})}(x_{i:j}) \cdot \prod_{s=i}^j x_s^{k-1}. \end{aligned}$$

Now for the bidiagonal decomposition (3.1) of G^{-1} we have, from (3.2), (3.3), and (3.4),

$$\begin{aligned} L_{i+1,i}^{(k)} &= -\frac{\det(V_{i-k+2:i+1,1:k}) \cdot s_{(\lambda_{n-k+1:n})}(x_{i-k+2:i+1})}{\det(V_{i-k+2:i,1:k-1}) \cdot s_{(\lambda_{n-k+2:n})}(x_{i-k+2:i})} \\ &\quad \times \frac{\det(V_{i-k+1:i-1,1:k-1}) \cdot s_{(\lambda_{n-k+2:n})}(x_{i-k+1:i-1})}{\det(V_{i-k+1:i,1:k}) \cdot s_{(\lambda_{n-k+1:n})}(x_{i-k+1:i})} \\ &= -\frac{s_{(\lambda_{n-k+1:n})}(x_{i-k+2:i+1})}{s_{(\lambda_{n-k+2:n})}(x_{i-k+2:i})} \cdot \frac{s_{(\lambda_{n-k+2:n})}(x_{i-k+1:i-1})}{s_{(\lambda_{n-k+1:n})}(x_{i-k+1:i})} \\ &\quad \times \prod_{j=i-k+1}^{i-1} \frac{x_{i+1} - x_{j+1}}{x_i - x_j}, \end{aligned} \quad (3.5)$$

$$(3.6) \quad D_{ii}^{-1} = \frac{\det(G_{1:i-1,1:i-1})}{\det(G_{1:i,1:i})} = \frac{s_{(\lambda_{n-i+2:n})}(x_{1:i-1})}{s_{(\lambda_{n-i+1:n})}(x_{1:i})} \cdot \prod_{j=1}^{i-1} \frac{1}{x_i - x_j},$$

$$\begin{aligned}
U_{i,i+1}^{(k)} &= - \frac{\det(V_{1:k,1:k}) \cdot s_{(\lambda_{n-i:n-i+k-1})}(x_{1:k}) \prod_{s=1}^k x_s^{i-k+1}}{\det(V_{1:k-1,1:k-1}) \cdot s_{(\lambda_{n-i+1:n-i+k-1})}(x_{1:k-1}) \prod_{s=1}^{k-1} x_s^{i-k+1}} \\
&\quad \times \frac{\det(V_{1:k-1,1:k-1}) \cdot s_{(\lambda_{n-i+2:n-i+k})}(x_{1:k-1}) \prod_{s=1}^{k-1} x_s^{i-k}}{\det(V_{1:k,1:k}) \cdot s_{(\lambda_{n-i+1:n-i+k})}(x_{1:k}) \prod_{s=1}^k x_s^{i-k}} \\
(3.7) \quad &= -x_k \cdot \frac{s_{(\lambda_{n-i:n-i+k-1})}(x_{1:k})}{s_{(\lambda_{n-i+1:n-i+k-1})}(x_{1:k-1})} \cdot \frac{s_{(\lambda_{n-i+2:n-i+k})}(x_{1:k-1})}{s_{(\lambda_{n-i+1:n-i+k})}(x_{1:k})}.
\end{aligned}$$

3.1. Let $\lambda = (1)$ and $n = 4$. Then $s_{(1)}(x_1, x_2, \dots, x_i) = x_1 + x_2 + \dots + x_i$ and the decomposition (3.1) of G^{-1} is

$$\begin{aligned}
&\begin{bmatrix} 1 & x_1 & x_1^2 & x_1^4 \\ 1 & x_2 & x_2^2 & x_2^4 \\ 1 & x_3 & x_3^2 & x_3^4 \\ 1 & x_4 & x_4^2 & x_4^4 \end{bmatrix}^{-1} \\
&= \begin{bmatrix} 1 & -x_1 & & \\ & 1 & -x_1 & \\ & & 1 & -x_1 \\ & & & 1 \end{bmatrix} \begin{bmatrix} 1 & & & \\ & 1 & -x_2 & \\ & & 1 & -x_2 \\ & & & 1 \end{bmatrix} \begin{bmatrix} 1 & & & \\ & 1 & & -x_3(x_1 + x_2 + x_3) \\ & & 1 & \\ & & & 1 \end{bmatrix} \\
&\quad \times \text{diag} \left(1, x_2 - x_1, (x_3 - x_2)(x_3 - x_1), (x_4 - x_3)(x_4 - x_2)(x_4 - x_1) \sum_{i=1}^4 x_i \right)^{-1} \\
&\quad \times \begin{bmatrix} 1 & & & \\ & 1 & & \\ & & 1 & \\ & & & 1 \end{bmatrix} \begin{bmatrix} 1 & & & \\ & -\frac{x_3 - x_2}{x_2 - x_1} & & \\ & & -\frac{x_4 - x_3}{x_3 - x_2} & \\ & & & 1 \end{bmatrix} \begin{bmatrix} 1 & & & \\ & -1 & 1 & \\ & & -1 & 1 \\ & & & -1 & 1 \end{bmatrix}.
\end{aligned}$$

The Schur function can be computed to high relative accuracy as described in [7]. The expressions (3.5)–(3.7) involve only multiplications, divisions, and/or forming $x_i - x_j$, and thus are computed to high relative accuracy.

ALGORITHM 3.1 (bidiagonal decomposition of G^{-1}). *Input:* $x = (x_1, x_2, \dots, x_n)$. *Output:* G^{-1} . [7]

```

d = ones(n)
for i = 2 : n
    for j = i - 1 : -1 : 1
        dij = di,j+1 · (xi - xj)
    for k = 1 : n - 1
        for i = n - 1 : -1 : k
            Li+1,i(k) = -  $\frac{s_{(\lambda_{n-k+1:n})}(x_{i-k+2:i+1})}{s_{(\lambda_{n-k+2:n})}(x_{i-k+2:i})} \cdot \frac{s_{(\lambda_{n-k+2:n})}(x_{i-k+1:i-1})}{s_{(\lambda_{n-k+1:n})}(x_{i-k+1:i})} \cdot \frac{d_{i+1,i-k+2}}{d_{i,i-k+1}}$ 
        for i = 1 : n
            Dii-1 =  $\frac{s_{(\lambda_{n-i+2:n})}(x_{1:i-1})}{s_{(\lambda_{n-i+1:n})}(x_{1:i})} \cdot \frac{1}{d_{i1}}$ 
        for k = n - 1 : -1 : 1
            for i = k : n - 1
                Ui,i+1(k) = -xk ·  $\frac{s_{(\lambda_{n-i:n-i+k-1})}(x_{1:k})}{s_{(\lambda_{n-i+1:n-i+k-1})}(x_{1:k-1})} \cdot \frac{s_{(\lambda_{n-i+2:n-i+k})}(x_{1:k-1})}{s_{(\lambda_{n-i+1:n-i+k})}(x_{1:k})}$ 
            end
        end
    end
end

```

Algorithm 3.1 implements (3.5)–(3.7) straightforwardly. For efficiency we precompute $d_{ij} = (x_i - x_{i-1})(x_i - x_{i-2}) \dots (x_i - x_j)$ for all $i > j$ at the cost of n^2 floating point operations.

Let ϵ denote machine precision. We now briefly discuss the two algorithms from [7] for computing the Schur function to high relative accuracy.

- The first algorithm evaluates the $s_\mu(x_{1:n})$ [23, (3.5), p. 41]

$$(3.8) \quad s_\mu = \det(e_{\mu'_i - i + j})_{1 \leq i, j \leq \mu_1}$$

in $(|\mu|\mu_1 \log \mu_1 + \log \frac{1}{\epsilon})$ -digit arithmetic, then rounds the result back to $\log \frac{1}{\epsilon}$ digits, i.e., to precision ϵ . Here e_k are the elementary symmetric polynomials and μ' is the conjugate partition ($\mu'_i = \#(\mu_j \geq i)$). The amount of subtractive cancellation in evaluating (3.8) is bounded, and by using just $|\mu|\mu_1 \log \mu$ extra digits, s_μ is computed to high relative accuracy. The cost of computing $s_\mu(x_{i:j})$ is $O((j-i)|\mu| + \mu_1^3)(|\mu|\mu_1)^{1+\rho})$, where ρ is tiny and accounts for certain logarithmic functions. Of this cost $O((j-i)|\mu|(|\mu|\mu_1)^{1+\rho})$ is for computing the e_i and the rest is for computing the determinant in (3.8), all done in $(|\mu|\mu_1 \log \mu_1 + \log \frac{1}{\epsilon})$ -digit arithmetic. The functions $e_k(x_{i:j})$, $1 \leq k \leq |\lambda|$, $1 \leq i \leq j \leq n$, need only be computed once. Since all Schur functions $s_\mu(x_{i:j})$ in Algorithm 3.1 satisfy $j-i < n$ and $\mu \leq \lambda$, the overall cost of Algorithm 3.1 is bounded by

$$(3.9) \quad O(n^2 + n^2|\lambda|(|\lambda|\lambda_1)^{1+\rho} + n^2\lambda_1^3(|\lambda|\lambda_1)^{1+\rho}) \leq O(n^2|\lambda|^{2+\rho}\lambda_1^{3+\rho}).$$

- The second algorithm uses the recursive formula [23, (5.9), p. 72]

$$(3.10) \quad s_\lambda(x_{1:n}) = \sum_{\mu \leq \lambda} s_\mu(x_{1:n-1}) \cdot x_n^{|\lambda| - |\mu|},$$

which involves only additions of positive quantities and multiplications, and is therefore always accurate. There is no need to use extra precision. In the process of computing $s_\lambda(x_{1:n})$ using (3.10) one ends up computing also $s_\nu(x_{1:m})$ for all $\nu \leq \lambda$ and $m \leq n$. In practice we compute $s_\lambda(x_{i:n})$ for $i = 1, 2, \dots, n-1$ and obtain s_μ Schur functions that appear in (3.5)–(3.7) (see [21, Algorithm 4.4.1, p. 69] for an implementation). The complexity of Algorithm 3.1 can then be bounded by

$$O\left(n^2 e^{5.2|\lambda|^{1/2}}\right).$$

This bound is asymptotically worse than (3.9), but this approach does not require the use of extended precision.

4. Björck–Pereyra-type method for solving $Gy = b$. We now present our main algorithm for solving a generalized Vandermonde linear system.

ALGORITHM 4.1 (Björck–Pereyra-type method for solving $Gy = b$). G is a generalized Vandermonde matrix, $Gy = b$, $b \in \mathbb{C}^n$, $O(n^2)$ operations.

$$y = G^{-1}b = U^{(1)}U^{(2)} \dots U^{(n-1)}D^{-1}L^{(n-1)}L^{(n-2)} \dots L^{(1)}b$$

```

for  $k = 1 : n - 1$ 
  for  $i = n - 1 : -1 : k$ 
     $b_{i+1} = b_{i+1} - b_i \cdot L_{i+1,i}^{(k)}$ 
  for  $i = 1 : n$   $b_i = b_i \cdot D_{ii}^{-1}$ 
  for  $k = n - 1 : -1 : 1$ 
    for  $i = k : n - 1$ 
       $b_i = b_i - b_{i+1} \cdot U_{i,i+1}^{(k)}$ 

```

When b has an alternating sign pattern ($\text{sign}(b_i) = (-1)^i$ or $\text{sign}(b_i) = -(-1)^i$) then Algorithm 4.1 is subtraction-free, so that y is very accurate. Algorithm 4.1 is the natural generalization of the Björck–Pereyra algorithm [3] for ordinary Vandermonde matrices. A MATLAB implementation of Algorithm 4.1 is available from [20].

5. The inverse of G . In this section we present explicit expressions for the entries of G^{-1} and describe two methods for computing these entries to high relative accuracy.

The $(n-1) \times (n-1)$ minor of G , obtained by removing row k and column l , is a generalized Vandermonde determinant. We use Cramer’s rule to obtain explicit formulas for the entries of G^{-1} :

$$\begin{aligned}
 (G^{-1})_{lk} &= (-1)^{k+l} \frac{\det(x_i^{j-1+\lambda_{n-j+1}})_{1 \leq i, j \leq n; i \neq k, j \neq l}}{\det G} \\
 &= (-1)^{k+l} \frac{s_{\lambda^{(k,l)}}(x_1, \dots, \hat{x}_k, \dots, x_n) \cdot \det V(x_1, \dots, \hat{x}_k, \dots, x_n)}{s_{\lambda}(x_1, \dots, x_n) \cdot \det V} \\
 (5.1) \quad &= (-1)^{n-l} \frac{s_{\lambda^{(k,l)}}(x_1, \dots, \hat{x}_k, \dots, x_n)}{s_{\lambda}(x_1, \dots, x_n)} \cdot \prod_{i=1, i \neq k}^n \frac{1}{x_k - x_i},
 \end{aligned}$$

where $\lambda^{(k,l)} = (\lambda_1 + 1, \lambda_2 + 1, \dots, \lambda_{n-l} + 1, \lambda_{n-l+2}, \dots, \lambda_n)$. The “hats” indicate the omission of the symbols they cover. The formula (5.1) was also obtained in [29], where the Schur function is referred to as a “D-determinant.”

Using (5.1) we can compute G^{-1} to high relative accuracy componentwise by employing the techniques from [7] for computing the Schur function. The last factor in (5.1), $\prod_{i=1, i \neq k}^n \frac{1}{x_k - x_i}$, is computed to high relative accuracy.

Alternatively, G^{-1} can be computed by multiplying out (3.1). Because of the checkerboard sign pattern of the factors in (3.1), no subtractive cancellation can occur in this multiplication, obtaining each entry of G^{-1} to high relative accuracy.

Both methods scale cubically with n and as a low order polynomial in $|\lambda|$.

6. The LDU decomposition of G . In this section we present explicit formulas for the entries of the LDU decomposition of G resulting from Gaussian elimination with no pivoting. We also present two methods for computing these entries to high relative accuracy.

Let $G = LDU$, where L and U are unit lower and upper triangular matrices, respectively. Then [11]

$$D_{ii} = \frac{\det G_{1:i,1:i}}{\det G_{1:i-1,1:i-1}}; \quad U_{ij} = \frac{\det G_{1:i,[1:i-1,j]}}{\det G_{1:i,1:i}}; \quad \text{and} \quad L_{ij} = \frac{\det G_{[1:j-1,i],1:j}}{\det G_{1:j,1:j}}.$$

All minors in the above expressions are generalized Vandermonde determinants; thus

$$\begin{aligned}\det G_{1:i,1:i} &= (\det V_{1:i,1:i}) \cdot s_{(\lambda_{n-i+1}, \dots, \lambda_p)}(x_{1:i}); \\ \det G_{[1:i-1,j],1:i} &= (\det V_{[1:i-1,j],1:i}) \cdot s_{(\lambda_{n-i+1}, \dots, \lambda_p)}(x_{[1:i-1,j]}); \\ \det G_{1:i,[1:i-1,j]} &= (\det V_{1:i,[1:i-1,j]}) \cdot s_{(\lambda_{n-j+1+j-i}, \lambda_{n-i+2}, \dots, \lambda_p)}(x_{1:i}).\end{aligned}$$

Therefore

$$(6.1) \quad D_{ii} = \frac{s_{(\lambda_{n-i+1}, \dots, \lambda_p)}(x_{1:i})}{s_{(\lambda_{n-i+2}, \dots, \lambda_p)}(x_{1:i-1})} \prod_{j=1}^{i-1} (x_i - x_j);$$

$$(6.2) \quad U_{ij} = \frac{s_{(\lambda_{n-j+1+j-i}, \lambda_{n-i+2}, \dots, \lambda_p)}(x_{1:i})}{s_{(\lambda_{n-i+1}, \dots, \lambda_p)}(x_{1:i})}, \quad i < j;$$

$$(6.3) \quad L_{ij} = \frac{s_{(\lambda_{n-j+1}, \dots, \lambda_p)}(x_{[1:j-1,i]})}{s_{(\lambda_{n-j+1}, \dots, \lambda_p)}(x_{1:j})} \prod_{k=1}^{j-1} \frac{x_i - x_k}{x_j - x_k}, \quad i > j.$$

The formulas (6.1)–(6.3) can be used to compute each entry of L , D , and U to high relative accuracy if the techniques from [7] are used for computing the Schur function. The total cost will be $O(n^2)$, where the constant hidden in the big- O notation grows like a low order polynomial in $|\lambda|$ and can be found in [7].

Alternatively, the entries of L , D , and U can be computed starting with (3.1). For instance,

$$L = (L^{(1)})^{-1} (L^{(2)})^{-1} \dots (L^{(n-1)})^{-1}.$$

Since $L^{(i)}$ is bidiagonal, each multiplication by $(L^{(i)})^{-1}$ can be performed using only additions and multiplications of positive numbers in $O(n^2)$ time. The total cost of that approach is $O(n^3)$, in addition to computing (3.1), and yields each entry of L , D , and U to high relative accuracy.

The formulas (6.1)–(6.3) were also obtained in [29] along with a subtraction-free procedure for computing the entries of L , D , and U to high relative accuracy. No complexity analysis was performed in [29].

Our interest in the LDU decomposition of TP matrices stems from the fact that an accurate LDU decomposition of a TP matrix yields a solution to a TP linear system with small componentwise relative backward error [5]. Our Björck–Pereyra-type method from section 3 goes a step further by producing a solution that also has a small rel_gap_x error when the right-hand side has an alternating sign pattern.

We consider only the LDU decomposition from Gaussian elimination with no pivoting, since pivoting is not needed to ensure stability when solving a TP linear system [14] (see also parts 3 and 4 of Proposition 7.2).

7. Perturbation theory and error analysis. We will now show that the entries of the bidiagonal decomposition (3.1) of G^{-1} , each entry of L , D , and U in the LDU decomposition of G , and each entry of G^{-1} may be very well conditioned functions of the data x_i despite the fact that $\kappa(G)$ may be very large.

THEOREM 7.1. $\hat{x}_i = x_i(1 + \delta_i)$, $|\delta_i| < \delta < 1/|\lambda|$, $x_i > 0$, $L^{(k)} = D^{-1} U^{(k)}$, $G^{-1} = L D U$, $G = \hat{G}$, $\text{rel\_gap}_x = \min_{i \neq j} \frac{|x_i - x_j|}{|x_i| + |x_j|}$.

$$\text{rel\_gap}_x \equiv \min_{i \neq j} \frac{|x_i - x_j|}{|x_i| + |x_j|}.$$

$$|\hat{z} - z| \leq \frac{(4|\lambda| + 2n/\text{rel\_gap}_x) \delta}{1 - (4|\lambda| + 2n/\text{rel\_gap}_x) \delta} |z|.$$

The Schur function $s_\mu(x_{i:j})$, $1 \leq i \leq j \leq n$, $\mu \leq \lambda$, is a polynomial of degree $|\mu| \leq |\lambda|$ with positive coefficients. Therefore [7]

$$(7.1) \quad |s_\mu(\hat{x}_{1:i}) - s_\mu(x_{1:i})| \leq \frac{\delta|\mu|}{1 - \delta|\mu|} s_\mu(x_{1:i}),$$

which, combined with formulas (3.5)–(3.7), (5.1), and (6.1)–(6.3), implies the desired result. $\square$

Therefore the quantity

$$(7.2) \quad \kappa^{\text{struct}}(G) = 4|\lambda| + 2n/\text{rel\_gap}_x$$

is the correct estimate of $\kappa(G)$ of G , which can be arbitrarily smaller than $\kappa(G)$.

We now turn our attention to the accuracy of the computed solution.

PROPOSITION 7.2 (see [1, 4, 18]). *Let $G \in \mathbb{R}^{n \times n}$ be TP, $L^{(i)} = U^{(i)} D^{-1}$, $\xi \ll 1$, $\hat{y} = G^{-1}b$, $Gy = b$, $\epsilon = \|G - \hat{G}\|$, $\epsilon \leq 4$.*

1. $|y - \hat{y}| \leq O(\epsilon + \xi) |G^{-1}| |b|$
2. $b_i - \hat{y}_i \leq (-1)^i b_i$, $i = 1, 2, \dots, n$

$$|y - \hat{y}| \leq O(\epsilon + \xi) |y|.$$

3. $|b - G\hat{y}| \leq O(\epsilon + \xi) |G| |\hat{y}| + O((\epsilon + \xi)^2)$.

4. $\hat{G}\hat{y} = b$, $\hat{G} = G + \hat{G} - G$, [6, p. 37] [27], $|G - \hat{G}| \leq O(\epsilon + \xi) |G| + O((\epsilon + \xi)^2)$.

A proof is available in [1, 4, 18]; we just mention some details. Since G is TP, the bidiagonal factors in (3.1) have positive diagonals and nonpositive offdiagonals. Thus the product of all bidiagonals may be formed, obtaining G^{-1} without suffering any subtractive cancellation, proving part 1. If the right-hand side has an alternating sign pattern, then the solution is also formed without subtractive cancellation, proving part 2. Parts 3 and 4 are equivalent [27] and follow from the fact that the inverse of a small relative componentwise perturbation of a bidiagonal matrix is a small relative componentwise perturbation of the inverse of the bidiagonal matrix.

8. Numerical experiments. We performed extensive numerical experiments and confirmed the correctness and cost of our algorithms. We selected two of those tests and report them below.

TABLE 8.1

Solving a 12×12 generalized Vandermonde linear system, $\lambda = (6, 3, 2, 1, 1)$. The digits that disagree with the results from Mathematica are underlined.

Nodes x_i	Mathematica	Algorithm 3.1	Traditional
2.0594766805425224	-1.0555833300893850e12	-1.055583330089384 <u>8</u> e12	-1.070849603216741 <u>2</u> e12
2.0940348689727863	4.1382711534659341e12	4.138271153465931 <u>6</u> e12	4.1982342784471191e12
2.1568348650743552	-7.1458326298047432e12	-7.145832629804740 <u>2</u> e12	-7.249574915129184 <u>6</u> e12
2.3449660139029209	7.0736899709677744e12	7.073689970967772 <u>5</u> e12	7.176586047671326 <u>2</u> e12
2.3472440864992130	-4.3450312586851128e12	-4.3450312586851128e12	-4.4083608888447139e12
2.4163531061463885	1.6416125390612351e12	1.641612539061234 <u>9</u> e12	1.6655874274458679e12
2.6563511105968578	-3.2574038855981763e11	-3.2574038855981757e11	-3.3050732345287665e11
2.7184145910794202	1.5677076891114130e10	1.567707689111412 <u>8</u> e10	1.590744471811275 <u>1</u> e10
2.7423051335090571	-2.9091221498536110e09	-2.9091221498536110e09	-2.951959663177880 <u>3</u> e09
2.8840219496751516	4.3989708280861162e07	4.3989708280861154e07	4.4640199653213106e07
2.9550986332751350	-5.8977191662033915e05	-5.8977191662033915e05	-5.985301929325286 <u>8</u> e05
2.9582142929074067	1.1347682330511120e02	1.134768233051112 <u>2</u> e02	1.151764159134706 <u>3</u> e02

We tested Algorithm 3.1 as follows. We chose a partition $\lambda = (6, 3, 2, 1, 1)$, $n = 12$, and uniformly distributed random nodes $x_i \in [2, 3]$, which we then sorted in increasing order, and a right-hand side

$$b = (-1, 1, -1, 1, -1, 1, -1, 1, -1, 1, -1, 1)^T$$

with alternating signs. This is equivalent to interpolating the data b with a polynomial of the form

$$f(x) = a_0 + a_1x + a_2x^2 + a_3x^3 + a_4x^4 + a_5x^5 + a_6x^6 + a_7x^8 + a_8x^9 + a_9x^{11} + a_{10}x^{13} + a_{11}x^{17}$$

at the nodes x_i , $i = 1, 2, \dots, 12$. This resulted in a TP generalized Vandermonde matrix G that was quite ill conditioned: $\kappa(G) \approx 3.7 \cdot 10^{18}$. The choice of a right-hand side with alternating signs guarantees that the solution will be computed with no subtractive cancellation with Algorithm 3.1.

We computed the solution to $Gz = b$ using Algorithm 3.1 and also using 100-digit arithmetic in `mpfr` [30] and got the same result to 14 digits, a loss of at most two digits of accuracy in each entry. Note that we generated the random numbers x_i in MATLAB, output them as decimal numbers to a file, read them into `mpfr`, converted them to 100 decimal digit big floats, and then formed the explicit matrix which we then solved with. In other words, we more than likely made some errors in the 16th place just in forming the matrix in `mpfr`. The smallest relative gap between the x_i 's is $\text{rel\_gap}_x \approx 10^{-3}$ and so the structured condition number (7.2) is only $\kappa^{\text{struct}} \approx 10^4$. Thus the problem is well conditioned, and we could expect the solution computed by `mpfr` and the one computed by Algorithm 3.1 to differ in at most four digits in the last place. The results in Table 8.1 confirmed our claims that the problem is well conditioned, even though $\kappa(G) \approx 3.7 \cdot 10^{18}$ and the traditional algorithm computed a solution that was only good in the first digit. Since Algorithm 3.1 computes the Schur function at every step, this test also confirms the correctness of the algorithm for computing the Schur function from [7].

Acknowledgments. The authors would like to thank Jose-Javier Martínez for pointing out to us the relevance of the paper [24] and especially Nick Higham for suggesting the reference [8] and for several helpful discussions in the preparation of this paper.

REFERENCES

- [1] P. ALONSO, M. GASCA, AND J. M. PEÑA, *Backward error analysis of Neville elimination*, Appl. Numer. Math., 23 (1997), pp. 193–204.
- [2] A. BERENSTEIN, S. FOMIN, AND A. ZELEVINSKY, *Parametrizations of canonical bases and totally positive matrices*, Adv. Math., 122 (1996), pp. 49–149.
- [3] Å. BJÖRCK AND V. PEREYRA, *Solution of Vandermonde systems of equations*, Math. Comp., 24 (1970), pp. 893–903.
- [4] T. BOROS, T. KAILATH, AND V. OLSHEVSKY, *A fast parallel Björck-Pereyra-type algorithm for solving Cauchy linear equations*, Linear Algebra Appl., 302/303 (1999), pp. 265–293.
- [5] C. DE BOOR AND A. PINKUS, *Backward error analysis for totally positive linear systems*, Numer. Math., 27 (1977), pp. 485–490.
- [6] J. W. DEMMEL, *Applied Numerical Linear Algebra*, SIAM, Philadelphia, 1997.
- [7] J. DEMMEL AND P. KOEV, *Accurate and efficient evaluation of Schur and Jack functions*, Math. Comp., to appear.
- [8] S. M. FALLAT, *Bidiagonal factorizations of totally nonnegative matrices*, Amer. Math. Monthly, 108 (2001), pp. 697–712.
- [9] S. FOMIN AND A. ZELEVINSKY, *Double Bruhat cells and total positivity*, J. Amer. Math. Soc., 12 (1999), pp. 335–380.
- [10] S. FOMIN AND A. ZELEVINSKY, *Total positivity: Tests and parametrizations*, Math. Intelligencer, 22 (2000), pp. 23–33.
- [11] F. GANTMACHER, *The Theory of Matrices*, AMS Chelsea Publishing, Providence, RI, 1998.
- [12] F. GANTMACHER AND M. KREIN, *Oscillation Matrices and Kernels and Small Vibrations of Mechanical Systems*, revised ed., AMS Chelsea Publishing, Providence, RI, 2002.
- [13] M. GASCA AND J. M. PEÑA, *Total positivity and Neville elimination*, Linear Algebra Appl., 165 (1992), pp. 25–44.
- [14] M. GASCA AND J. M. PEÑA, *Scaled pivoting in Gauss and Neville elimination for totally positive systems*, Appl. Numer. Math., 13 (1993), pp. 345–355.
- [15] M. GASCA AND J. M. PEÑA, *A matricial description of Neville elimination with applications to total positivity*, Linear Algebra Appl., 202 (1994), pp. 33–53.
- [16] M. GASCA AND J. M. PEÑA, *On factorizations of totally positive matrices*, in Total Positivity and Its Applications, Kluwer Academic, Dordrecht, The Netherlands, 1996, pp. 109–130.
- [17] N. J. HIGHAM, *Error analysis of the Björck-Pereyra algorithms for solving Vandermonde systems*, Numer. Math., 50 (1987), pp. 613–632.
- [18] N. J. HIGHAM, *Stability analysis of algorithms for solving confluent Vandermonde-like systems*, SIAM J. Matrix Anal. Appl., 11 (1990), pp. 23–41.
- [19] W. KAHAN AND I. FARKAS, *Algorithms* 167–169, Comm. ACM, 6 (1963), pp. 164–165; see also the *certification*, Comm. ACM, 6 (1963), p. 523.
- [20] P. KOEV, <http://www-math.mit.edu/~plamen>.
- [21] P. KOEV, *Accurate and Efficient Computations with Structured Matrices*, Ph.D. dissertation, University of California, Berkeley, CA, 2002.
- [22] P. KOEV, *Accurate eigenvalues and SVDs of totally nonnegative matrices*, SIAM J. Matrix Anal. Appl., 27 (2005), pp. 1–23.
- [23] I. G. MACDONALD, *Symmetric Functions and Hall Polynomials*, 2nd ed., Oxford University Press, New York, 1995.
- [24] J. J. MARTÍNEZ AND J. M. PEÑA, *Factorizations of Cauchy-Vandermonde matrices*, Linear Algebra Appl., 284 (1998), pp. 229–237.
- [25] J. J. MARTÍNEZ AND J. M. PEÑA, *Fast algorithms of Björck-Pereyra type for solving Cauchy-Vandermonde linear systems*, Appl. Numer. Math., 26 (1998), pp. 343–352.
- [26] *MATLAB Reference Guide*, The MathWorks, Inc., Natick, MA, 1992.
- [27] W. OETTLI AND W. PRAGER, *Compatibility of approximate solution of linear equations with given error bounds for coefficients and right hand sides*, Numer. Math., 6 (1964), pp. 405–409.
- [28] H. ORUÇ AND G. M. PHILLIPS, *Explicit factorization of the Vandermonde matrix*, Linear Algebra Appl., 315 (2000), pp. 113–123.
- [29] H. VAN DE VEL, *Numerical treatment of a generalized Vandermonde system of equations*, Linear Algebra Appl., 17 (1977), pp. 149–179.
- [30] S. WOLFRAM, *Mathematica: A System for Doing Mathematics by Computer*, Addison-Wesley, Redwood City, CA, 1988.

THE EHRlich–ABERTH METHOD FOR THE NONSYMMETRIC TRIDIAGONAL EIGENVALUE PROBLEM*

DARIO A. BINI[†], LUCA GEMIGNANI[†], AND FRANÇOISE TISSEUR[‡]

Abstract. An algorithm based on the Ehrlich–Aberth iteration is presented for the computation of the zeros of $p(\lambda) = \det(T - \lambda I)$, where T is a real irreducible nonsymmetric tridiagonal matrix. The algorithm requires the evaluation of $p(\lambda)/p'(\lambda) = -1/\text{trace}(T - \lambda I)^{-1}$, which is done by exploiting the QR factorization of $T - \lambda I$ and the semiseparable structure of $(T - \lambda I)^{-1}$. The choice of initial approximations relies on a divide-and-conquer strategy, and some results motivating this strategy are given. Guaranteed a posteriori error bounds based on a running error analysis are proved. A Fortran 95 module implementing the algorithm is provided and numerical experiments that confirm the effectiveness and the robustness of the approach are presented. In particular, comparisons with the LAPACK subroutine `dhesqr` show that our algorithm is faster for large dimensions.

Key words. nonsymmetric eigenvalue problem, symmetric indefinite generalized eigenvalue problem, tridiagonal matrix, root finder, QR decomposition, divide and conquer

AMS subject classification. 65F15

DOI. 10.1137/S0895479803429788

1. Introduction. Real nonsymmetric tridiagonal eigenvalue problems arise as intermediate steps in a variety of eigenvalue problems. For example, the nonsymmetric eigenvalue problem can be reduced in a finite number of steps to nonsymmetric tridiagonal form [13], [18]. In the sparse case, the nonsymmetric Lanczos algorithm produces a nonsymmetric tridiagonal matrix. Other motivation for this work comes from the symmetric quadratic eigenvalue problem

$$(\lambda^2 M + \lambda C + K)x = 0,$$

with M, C , and K real symmetric matrices, which is frequently encountered in structural mechanics [40]. The standard way of dealing with this problem in practice is to reformulate it as a generalized eigenvalue problem (GEP) $Ax = \lambda Bx$ of twice the dimension, a process called linearization. Symmetry in the problem can be maintained with an appropriate choice of linearization [40], such as, for example,

$$A = \begin{bmatrix} 0 & K \\ K & C \end{bmatrix}, \quad B = \begin{bmatrix} K & 0 \\ 0 & -M \end{bmatrix}, \quad x = \begin{bmatrix} u \\ \lambda u \end{bmatrix}.$$

The resulting A and B are symmetric but not definite, and in general the pair (A, B) is indefinite. When the pair (A, B) is of small to medium size, it can be reduced to a symmetric tridiagonal-diagonal pair (S, D) using one of the procedures described by Tisseur [39]. This is the most compact form that can be obtained in a finite number

*Received by the editors June 5, 2003; accepted for publication (in revised form) November 15, 2004; published electronically July 13, 2005.

<http://www.siam.org/journals/simax/27-1/42978.html>

[†]Dipartimento di Matematica, Università di Pisa, via Buonarroti 2, 56127 Pisa (bini@dm.unipi.it, gemignani@dm.unipi.it). The work of the first author was supported by MIUR grant 2002014121. The work of the second author was supported by GNCS of Istituto Nazionale Di Alta Matematica.

[‡]Department of Mathematics, University of Manchester, Manchester, M13 9PL, England (ftisseur@ma.man.ac.uk, <http://www.ma.man.ac.uk/~ftisseur/>). The work of this author was supported by Engineering and Physical Sciences Research Council grant GR/R45079 and Nuffield Foundation grant NAL/00216/G.

of steps. For large and sparse matrices the pseudo-Lanczos algorithm of Parlett and Chen [31] applied to $A - \lambda B$ yields a projected problem $S - \lambda D$ with S symmetric tridiagonal and D diagonal. In both cases, the eigenvalues of the symmetric pair (S, D) are the same as the eigenvalues of the nonsymmetric tridiagonal matrix $T = D^{-1}S$.

Our aim is to derive a robust algorithm that computes all the eigenvalues of a real $n \times n$ nonsymmetric tridiagonal matrix T in $O(n^2)$ operations. The QR algorithm [20] does not preserve tridiagonal structure: the matrix T is considered as a Hessenberg matrix and the upper part of T is filled in along the iterations. Therefore the QR algorithm requires some extra storage and the eigenvalues are computed in $O(n^3)$ operations. Two alternatives are the LR algorithm [38] for nonsymmetric tridiagonal matrices and the HR algorithm [7], [8]. Both algorithms preserve the tridiagonal form of T but may be unstable since they use nonorthogonal transformations. Attempts to solve the nonsymmetric tridiagonal eigenvalue by generalizing Cuppen's divide-and-conquer algorithm have been unsuccessful because of a lack of good root finders and because deflation is not as advantageous as it is in the symmetric case [2], [27].

In this paper we propose a root finder for the characteristic polynomial of T based on the Ehrlich–Aberth method [1], [15]. This method approximates simultaneously all the zeros of a polynomial $p(z)$: given a vector $z^{(0)} \in \mathbb{C}^n$ of initial approximations to the zeros of $p(z)$, the Ehrlich–Aberth iteration generates a sequence $z^{(j)} \in \mathbb{C}^n$ which locally converges to the n -tuple of the roots of $p(z)$, according to the equation

$$(1.1) \quad z_j^{(k+1)} = z_j^{(k)} - \frac{\frac{p(z_j^{(k)})}{p'(z_j^{(k)})}}{1 - \frac{p(z_j^{(k)})}{p'(z_j^{(k)})} \sum_{k=1, k \neq j}^n \frac{1}{z_j^{(k)} - z_k^{(k)}}}, \quad j = 1:n.$$

The convergence is superlinear (cubic or even higher if the implementation is in the Gauss–Seidel style) for simple roots and linear for multiple roots. In practice, the Ehrlich–Aberth iteration has good global convergence properties, though no theoretical results seem to be known about global convergence. In principle, we cannot exclude the possibility that the Ehrlich–Aberth simultaneous iteration fails to converge or fails to approximate certain roots. However, in practice this has never been encountered (see [5], [6]).

Other techniques for simultaneous iterations such as Laguerre, Durand–Kerner, Euler-like, and Halley-like methods [17], [34], [35], [36], [37] could be used as well. However, our computational experience in polynomial root-finding indicates that compared with these iterations, the Ehrlich–Aberth method has the advantages of requiring a small number of iterations for convergence and having a small cost per iteration. Note that the quasi-Laguerre iteration in [14] relies on the eigenvalues to be real.

The main requirements when using the Ehrlich–Aberth method for computing the roots of $p(z)$ are

1. a fast, robust, and stable computation of the Newton correction $p(z)/p'(z)$;
2. a criterion for choosing the initial approximations to the zeros, $z^{(0)}$, so that the number of iterations needed for convergence is not too large.

For the first issue, Bini [4] shows that Horner's rule is an effective tool when $p(z)$ is expressed in terms of its coefficients. In this case the cost of each simultaneous iteration is $O(n^2)$ operations. Moreover, Horner's rule is backward stable and its computation provides a cheap criterion to test whether the given approximation is in the root-neighborhood (pseudospectrum) of the polynomial [4]. This makes the Ehrlich–Aberth method an effective tool for approximating polynomial roots [6], and

it is now part of the MPSolve package (Multiprecision Polynomial Solver) [5].

In our context, where $p(\lambda) = \det(T - \lambda I)$ is not available explicitly, we need a tool having the same features as Horner's rule, that is, a tool that allows us to compute in a fast, stable, and robust way the Newton correction $p(\lambda)/p'(\lambda)$. This issue is discussed in section 2, where we use the QR factorization of $T - \lambda I$ and the semiseparable structure of $(T - \lambda I)^{-1}$ to compute the Newton correction by means of the equation

$$\frac{p(\lambda)}{p'(\lambda)} = -\frac{1}{\text{trace}(T - \lambda I)^{-1}}.$$

The algorithm that we obtain in this way fulfills the desired requirements of robustness and stability; in particular, it does not suffer unduly from underflow or overflow.

For the second issue, concerning the choice of initial approximations, we rely on the specific features of our eigenvalue problem. More precisely, we apply a divide-and-conquer strategy where the initial approximations are obtained by computing the eigenvalues of two suitable tridiagonal matrices of sizes $m = \lceil n/2 \rceil$ and $n - m$. Even though there are no theoretical results guaranteeing convergence under this choice, we provide in section 3 some theoretical results that motivate this strategy.

The complete algorithm is described in section 4, where we also deal with the issues of computing eigenvectors and running error bounds. Numerical experiments in section 5 illustrate the robustness of our algorithm. In particular, our results show that in most cases our algorithm performs faster than the LAPACK subroutine `dhseqr` for $n \geq 800$ and the speed-up for $n = 1600$ ranges from 3 to 70. The implementation in Fortran 95 is available as a module at www.dm.unipi.it/~bini/software, file `eigen.v1.1.tgz`.

2. Computing the Newton correction. Our aim in this section is to derive a fast, robust, and stable method for computing the Newton correction $p(\lambda)/p'(\lambda)$, where $p(\lambda) = \det(T - \lambda I)$.

The tridiagonal matrix

$$(2.1) \quad T = \begin{bmatrix} \alpha_1 & \gamma_1 & & & 0 \\ \beta_1 & \alpha_2 & \gamma_2 & & \\ & \beta_2 & \ddots & \ddots & \\ & & \ddots & \alpha_{n-1} & \gamma_{n-1} \\ 0 & & & \beta_{n-1} & \alpha_n \end{bmatrix} \in \mathbb{R}^{n \times n}$$

is said to be *regular* or *non-singular* if $\beta_j \gamma_j \neq 0$ for $j = 1:n-1$. We denote by T_k the leading principal submatrix of T in rows and columns 1 through k and let $p_k = \det(T_k - \lambda I)$.

A natural approach is to compute $p(\lambda) = p_n(\lambda)$ and its derivative by using the recurrence

$$(2.2) \quad \begin{aligned} p_0(\lambda) &= 1, \quad p_1(\lambda) = \alpha_1, \\ p_k(\lambda) &= (\alpha_k - \lambda)p_{k-1}(\lambda) - \beta_{k-1}\gamma_{k-1}p_{k-2}(\lambda), \quad k = 2:n, \end{aligned}$$

obtained by expanding $\det(T_k - \lambda I_k)$ by its last row. Since this recurrence is known to suffer from overflow and underflow problems [30], we adopt a different strategy.

Assume that λ is not a zero of p , that is, $p(\lambda) \neq 0$. Then

$$(2.3) \quad \frac{p'(\lambda)}{p(\lambda)} = -\sum_{j=1}^n \frac{1}{\lambda_j - \lambda} = -\text{trace}((T - \lambda I)^{-1}) = -\sum_{j=1}^n \theta_j,$$

where θ_j is the j th diagonal element of $(T - \lambda I)^{-1}$.

In what follows, S denotes the shifted tridiagonal matrix

$$S := T - \lambda I.$$

If S is unreduced, S^{-1} can be characterized in terms of four vectors, u , v , y , and z [26], such that

$$(2.4) \quad (S^{-1})_{jk} = \begin{cases} u_j v_k & \text{if } j \leq k, \\ y_j z_k & \text{if } j \geq k. \end{cases}$$

Note that $u_j v_j = y_j z_j$. These four vectors can be computed in $O(n)$ operations, and it is tempting to use them for the computation of Newton's correction via

$$p(\lambda)/p'(\lambda) = -1/\sum_{j=1}^n u_j v_j = -1/\sum_{j=1}^n y_j z_j.$$

However, as illustrated in [25], u , v , y , and z can be extremely badly scaled and their computation can break down because of overflow and underflow.

In the next two subsections, we describe two robust and efficient approaches for computing the trace of the inverse of a tridiagonal matrix and discuss our choice.

2.1. Dhillon's approach. Dhillon [11] proposes an algorithm to compute the 1-norm of the inverse of a tridiagonal matrix S in $O(n)$ operations that is more reliable than Higham's algorithm [24] based on the compact representation (2.4). As a by-product, Dhillon's approach provides $\text{trace}(S^{-1})$. His algorithm relies on the computation of the two triangular factorizations

$$(2.5) \quad S = L_+ D_+ U_+, \quad S = U_- D_- L_-,$$

where L_+ and L_- are unit lower bidiagonal, U_+ and U_- are unit upper bidiagonal, while $D_+ = \text{diag}(d_1^+, \dots, d_n^+)$ and $D_- = \text{diag}(d_1^-, \dots, d_n^-)$. If these factorizations exist, the diagonal entries of S^{-1} denoted by θ_j , $j = 1:n$, can be expressed in terms of the diagonal factors D_+ and D_- through the recurrence

$$(2.6) \quad \theta_1 = 1/d_1^-, \quad \theta_{j+1} = \theta_j \frac{d_j^+}{d_{j+1}^-}, \quad j = 1:n-1.$$

Note that the triangular factorizations (2.5) may suffer element growth. They can also break down prematurely if a zero pivot is encountered, that is, if $d_j^+ = 0$ or $d_{j+1}^- = 0$ for some j . To overcome this latter drawback, Dhillon [11] makes use of IEEE floating point arithmetic, which permits computations with $\pm\infty$. With this approach, Dhillon's algorithm always returns an approximation of $\text{trace}(S^{-1})$. It is worth pointing out that besides the multiplicative formulae (2.6) for computing θ_j , alternative formulae are introduced in [12] having an additive form which may make them more robust with respect to overflow and underflow.

In our implementation of the Ehrlich–Aberth method the computation of $\det(S)$ is needed at the last stage of the algorithm to provide an error bound for the computed eigenvalues (see section 4). Dhillon's algorithm can be modified to deal with the computation of $\det(S)$ by using a block formulation. Its stability properties are related to those of the (block) LU factorization for tridiagonal matrices.

2.2. A QR factorization approach. In this section we present an alternative algorithm for the computation of $\text{trace}(S^{-1})$ in $O(n)$ operations that is based on the properties of QR factorizations of tridiagonal matrices. Our algorithm keeps the element growth under control and does not have any difficulty caused by overflow and underflow, so there is no need to augment the algorithm with tests for dealing with degenerate cases, as in [11]. In addition, it provides $\det(S)$. Its cost is about twice the cost of the algorithm based on the LU factorization described in section 2.1.

Recall that in our application S is the shifted tridiagonal matrix $T - \lambda I$, where T is given by (2.1). Since λ can be complex, S has real subdiagonal and superdiagonal elements and complex diagonal elements. We denote by G_j the $n \times n$ unitary Givens rotation equal to the identity matrix except in rows and columns j and $j + 1$, where

$$G_j([j, j+1], [j, j+1]) = \begin{bmatrix} \phi_j & \psi_j \\ -\bar{\psi}_j & \bar{\phi}_j \end{bmatrix}, \quad |\phi_j|^2 + |\psi_j|^2 = 1.$$

In the following we choose ψ_j real; this choice enables us to reduce the arithmetic complexity of the computation. Let $S = QR$ be the QR factorization of S obtained by means of Givens rotations, so that

$$(2.7) \quad G_{n-1} \cdots G_2 G_1 S = R \quad \text{and} \quad Q^* = G_{n-1} \cdots G_2 G_1.$$

Since S is tridiagonal, R is an upper triangular matrix of the form

$$R = \begin{bmatrix} r_1 & s_1 & t_1 & & 0 \\ & \ddots & \ddots & \ddots & \\ & & r_{n-2} & s_{n-2} & t_{n-2} \\ & & & r_{n-1} & s_{n-1} \\ 0 & & & & r_n \end{bmatrix} \in \mathbb{C}^{n \times n}.$$

If $\phi_1 = \bar{\alpha}_1 \tau_1$ and $\psi_1 = \beta_1 \tau_1$ with $\tau_1 = 1/\sqrt{|\alpha_1|^2 + \beta_1^2}$, then

$$S_1 := G_1 S = \left[\begin{array}{c|ccc} r_1 & s_1 & t_1 & 0 & \dots \\ 0 & \tilde{\alpha}_2 & \tilde{\gamma}_2 & & 0 \\ \vdots & \beta_2 & \alpha_3 & \gamma_3 & \\ \vdots & 0 & \ddots & \ddots & \ddots \end{array} \right]$$

with

$$\begin{aligned} r_1 &= \phi_1 \alpha_1 + \psi_1 \beta_1, & s_1 &= \phi_1 \gamma_1 + \psi_1 \alpha_2, & t_1 &= \psi_1 \gamma_2, \\ \tilde{\alpha}_2 &= -\psi_1 \gamma_1 + \bar{\phi}_1 \alpha_2, & \tilde{\gamma}_2 &= \bar{\phi}_1 \gamma_2. \end{aligned}$$

Recursively applying the same transformation to the $(n-1) \times (n-1)$ trailing principal submatrix of S_1 yields the factorization (2.7), where

$$(2.8) \quad \begin{aligned} \tau_j &= 1/\sqrt{|\tilde{\alpha}_j|^2 + \beta_j^2}, & \phi_j &= \bar{\tilde{\alpha}}_j \tau_j, & \psi_j &= \beta_j \tau_j, \\ r_j &= \phi_j \tilde{\alpha}_j + \psi_j \beta_j, & s_j &= \phi_j \tilde{\gamma}_j + \psi_j \alpha_{j+1}, & t_j &= \psi_j \gamma_{j+1}, \\ \tilde{\alpha}_{j+1} &= -\psi_j \tilde{\gamma}_j + \bar{\phi}_j \alpha_{j+1}, & \tilde{\gamma}_{j+1} &= \bar{\phi}_j \gamma_{j+1} \end{aligned}$$

for $j = 1:n-1$, with $\tilde{\alpha}_1 = \alpha_1$ and $\tilde{\gamma}_1 = \gamma_1$. Note that all the ψ_j are real, and if S is unreduced, the ψ_j are nonzero.

The following result of [19] concerns the semiseparable structure of Q^* and is crucial in the computation of the diagonal entries of S^{-1} in $O(n)$ arithmetic operations.

THEOREM 2.1. . . . $S \in \mathbb{C}^{n \times n}$ $S = QR$.

(2.8)

$$(2.9) \quad \begin{aligned} D &= \text{diag}(1, -\psi_1, \psi_1\psi_2, \dots, (-1)^{n-1}\psi_1\psi_2 \cdots \psi_{n-1}), \\ u &= D^{-1}[1, \bar{\phi}_1, \bar{\phi}_2, \dots, \bar{\phi}_{n-1}]^T, \\ v &= D[\phi_1, \phi_2, \phi_3, \dots, \phi_{n-1}, 1]^T. \end{aligned}$$

$$Q^* = \begin{bmatrix} v_1 u_1 & \psi_1 & & & 0 \\ v_2 u_1 & v_2 u_2 & \psi_2 & & \\ \vdots & \vdots & \vdots & \ddots & \\ \vdots & \vdots & \vdots & v_{n-1} u_{n-1} & \psi_{n-1} \\ v_n u_1 & v_n u_2 & \cdots & v_n u_{n-1} & v_n u_n \end{bmatrix}.$$

For simplicity, we assume that S is nonsingular so that R^{-1} exists. Let w be the solution of the system $Rw = v$, where v is defined in (2.9). Then, using the structure of Q^* in Theorem 2.1 and the fact that R is triangular, the j th diagonal elements of S^{-1} , θ_j , is given by

$$\theta_j = e_j^* S^{-1} e_j = e_j^* R^{-1} Q^* e_j = u_j e_j^* R^{-1} v = u_j e_j^* w = u_j w_j,$$

and hence

$$\text{trace}(S^{-1}) = \sum_{j=1}^n u_j w_j.$$

Observe that the computation of u and v by means of (2.9) generates underflow and overflow problems: since the diagonal entries of D are products of the ψ_j with $|\psi_j| \leq 1$, then for large n , D may have diagonal entries that underflow to zero and inverting D would generate overflow. A way of avoiding this drawback is to scale the system $Rw = v$ with the diagonal matrix D of Theorem 2.1. This yields $\hat{R}\hat{w} = \hat{v}$, where

$$(2.10) \quad \hat{R} = D^{-1}RD, \quad \hat{w} = D^{-1}w, \quad \hat{v} = D^{-1}v = [\phi_1, \dots, \phi_{n-1}, 1]^T.$$

With this scaling, no accumulation of products of ψ_j is needed. The entries of the matrix $\hat{R}$ are given by

$$(2.11) \quad \hat{r}_j = r_j, \quad \hat{s}_j = -\psi_j s_j, \quad \hat{t}_j = \psi_j \psi_{j+1} t_j$$

and their computation does not generate overflow since $|\psi_j| \leq 1$. Underflow in the computation of $\hat{s}_j$ and $\hat{t}_j$ is not a problem since their inverses are not needed in the solution of $\hat{R}\hat{w} = \hat{v}$. The only terms that must be inverted in the computation of $\hat{w}$ are the diagonal elements of $\hat{R}$.

We now show that if overflow is encountered in the evaluation of $\hat{w}$, then R and therefore $S = T - \lambda I$ are numerically singular. In that case we have detected an eigenvalue. First observe that since R is upper triangular and $\hat{R}^{-1} = D^{-1}R^{-1}D$, the elements of $\hat{R}^{-1}$ are obtained by multiplying the corresponding elements of R^{-1} by

suitable products of ψ_j . Since $|\psi_j| \leq 1$ for any j , we deduce that $|\widehat{R}^{-1}| \leq |R^{-1}|$, where the inequality holds componentwise and $|R|$ denotes the matrix with elements $|r_{jk}|$. In this way we find that

$$(2.12) \quad \|\widehat{R}^{-1}\|_\infty \leq \|R^{-1}\|_\infty.$$

Second, since $\|\widehat{v}\|_\infty = 1$, we have $\|\widehat{w}\|_\infty \leq \|\widehat{R}^{-1}\|_\infty \|\widehat{v}\|_\infty = \|\widehat{R}^{-1}\|_\infty$, which complemented with (2.12) yields $\|\widehat{w}\|_\infty \leq \|R^{-1}\|_\infty$. Therefore, overflow in $\widehat{w}$ implies overflow in R^{-1} ; that is, $T - \lambda I$ is numerically singular.

Let

$$(2.13) \quad \widehat{u} = [1, \overline{\phi}_1, \overline{\phi}_2, \dots, \overline{\phi}_{n-1}]^T.$$

Note that because $|\phi_j|^2 + |\psi_j|^2 = 1$, the components of $\widehat{u}$ are all bounded by 1 in modulus. Since $w = D\widehat{w}$ and $u = D^{-1}\widehat{u}$, we find that $u_j w_j = \widehat{u}_j \widehat{w}_j$, $j = 1:n$, so that

$$(2.14) \quad \text{trace}(S^{-1}) = \sum_{j=1}^n \widehat{u}_j \widehat{w}_j$$

and

$$(2.15) \quad \det(S) = \prod_{j=1}^n r_j.$$

The relative error due to cancellation in computing (2.14) is bounded by a multiple of the unit roundoff times

$$(2.16) \quad |\widehat{u}|^T |\widehat{w}| / |\widehat{u}^T \widehat{w}| := \zeta,$$

where $|\widehat{u}|$ and $|\widehat{w}|$ denote the vectors whose components are the moduli of $\widehat{u}$ and $\widehat{w}$. By performing an asymptotic analysis of the order of $|\zeta|$ when $\lambda \rightarrow \mu$, where μ is a simple eigenvalue of T with right and left eigenvectors x and y , respectively, we find that

$$(2.17) \quad (T - \lambda I)^{-1} = (\mu - \lambda)^{-1} y x^T + O(1),$$

with $x^T y \neq 0$. Therefore, since the diagonal elements of $S^{-1} = (T - \lambda I)^{-1}$ are $\widehat{u}_j \widehat{w}_j$, $j = 1:n$, one has $\widehat{u}_j \widehat{w}_j = (\mu - \lambda)^{-1} x_j y_j + O(1)$. This implies that $|\widehat{u}|^T |\widehat{w}| = |\mu - \lambda|^{-1} |y|^T |x| + O(1)$ and $\widehat{u}^T \widehat{w} = (\mu - \lambda)^{-1} y^T x + O(1)$ so that

$$(2.18) \quad |\widehat{u}|^T |\widehat{w}| / |\widehat{u}^T \widehat{w}| = |y|^T |x| / |y^T x| + O(|\mu - \lambda|).$$

Recall that for ϵ in a neighborhood of 0, the matrix $T + \epsilon F$ has the eigenvalue $\mu + \epsilon y^T F x / (y^T x) + O(\epsilon^2)$ (see, for instance, [20]).

Moreover, if $F = \text{diag}(f_i)$ with $|f_i| = 1$ such that $y_i f_i x_i$ is real nonnegative, then $\epsilon y^T F x / (y^T x) = \epsilon |y|^T |x| / (y^T x)$; that is, the ratio $|y|^T |x| / |y^T x|$ is the condition number of the eigenvalue μ with respect to a particular diagonal perturbation. Therefore, if the values $\widehat{u}_i$ and $\widehat{w}_i$ are computed in a numerically stable way, then the accuracy of the computed value of $\sum_{j=1}^n \widehat{u}_j \widehat{w}_j$ is consistent with the conditioning of the eigenvalue. We can perform a similar asymptotic analysis when the multiplicity ν of μ is greater than 1, since $(T - \lambda I)^{-1} = (\mu - \lambda)^{-\nu} x y^T + O((\mu - \lambda)^{-\nu+1})$. However, in this

case the condition $x^T y \neq 0$ is not generally satisfied. Concerning the accuracy of the computation of the diagonal elements of R , we refer the reader to section 4.1.

The analysis based on (2.18) is a “first order” analysis and keeps its validity only from the asymptotic point of view as $\lambda \rightarrow \mu$. Indeed, for a finite value of λ this analysis may lose its validity if, say, $|x^T y|$ is much smaller than $|\mu - \lambda|$ so that the $O(|\mu - \lambda|)$ term in (2.18) is no longer negligible. On the other hand, this analysis is useful for designing a heuristic stopping criterion.

Equations (2.8) and (2.10)–(2.14) constitute our algorithm for the computation of $p'(\lambda)/p(\lambda) = \text{trace}((T - \lambda I)^{-1}) = \text{trace}(S^{-1})$, which we summarize below in pseudocode. The function *Givens* constructs ϕ_j and ψ_j and guards against the risk of overflow. We refer to Bindel et al. [3] for a detailed explanation on how this function should be implemented.

```

function  $\tau = \text{trace\_Tinv}(\beta, \alpha, \gamma)$ 
% Compute  $\tau = \text{trace}(S^{-1})$ , where  $S = \text{tridiag}(\beta, \alpha, \gamma)$  is  $n \times n$  tridiagonal
% with real off-diagonals and complex diagonal.
 $a = \alpha_1, g = \gamma_1, u_1 = 1$ 
% Computes vectors  $\hat{r}, \hat{s}, \hat{t}$  in (2.11),  $\hat{u}$  in (2.13) and  $\hat{v}$  in (2.10).
for  $j = 1 : n - 1$ 
     $(\phi, \psi) = \text{Givens}(a, \beta_j)$ 
     $r_j = \phi a + \psi \beta_j, s_j = -\psi(\phi g + \psi \alpha_{j+1})$ 
     $a = -\psi g + \phi \alpha_{j+1}, u_{j+1} = \phi, v_j = \phi$ 
    if  $j < n - 1, t_j = \psi^2 \gamma_{j+1}, g = \phi \gamma_{j+1}$ , end
    if  $j > 1, t_{j-1} = \psi t_{j-1}$ , end
end
 $r_n = a, v_n = 1$ 
% Solve the linear system  $\hat{R}\hat{w} = \hat{v}$ .
 $w_n = v_n / r_n$ 
 $w_{n-1} = (v_{n-1} - w_n s_{n-1}) / r_{n-1}$ 
for  $j = n - 2 : -1 : 1$ 
     $w_j = (v_j - w_{j+1} s_j - w_{j+2} t_j) / r_j$ 
    if  $w_j = \text{Inf}, \tau = \text{Inf}$ , return, end
end
 $\tau = \sum_{j=1}^n u_j w_j$ 

```

The function `trace_Tinv` requires $O(n)$ operations.

3. Choosing initial approximations. The choice of the initial approximations $z_j^{(0)}, j = 1:n$, used to start the Ehrlich–Aberth iteration (1.1) crucially affects the number of steps needed by the method. For polynomials expressed in terms of their coefficients, Bini [4] derived a criterion for selecting initial guesses based on a combination of Rouché’s theorem and the use of the Newton polygon. Here we follow a divide-and-conquer strategy which better exploits the tridiagonal nature of the problem and seems to perform well in practice. More precisely, the initial approximations are chosen from the eigenvalues of two suitable tridiagonal submatrices of order roughly $n/2$.

Rewrite T as

$$(3.1) \quad T = \tilde{T} + uv^T,$$

where

$$(3.2) \quad \tilde{T} = \begin{bmatrix} T_1 & 0 \\ 0 & T_2 \end{bmatrix} = T_1 \oplus T_2, \quad u = e_m + e_{m+1}, \quad v = \beta_m e_m + \gamma_m e_{m+1}$$

and

$$T_1 = \begin{bmatrix} \alpha_1 & \gamma_1 & & 0 \\ \beta_1 & \alpha_2 & \ddots & \\ & \ddots & \ddots & \gamma_{m-1} \\ 0 & & \beta_{m-1} & \alpha_m - \beta_m \end{bmatrix}, \quad T_2 = \begin{bmatrix} \alpha_{m+1} - \gamma_m & \gamma_{m+1} & & 0 \\ \beta_{m+1} & \alpha_{m+2} & \ddots & \\ & \ddots & \ddots & \gamma_{n-1} \\ 0 & & \beta_{n-1} & \alpha_n \end{bmatrix}.$$

There are no obvious connections between the eigenvalues of T and those of $\tilde{T}$, unlike in the symmetric case, in which the eigenvalues of T interlace those of $\tilde{T}$. However, in this section, we show that the eigenvalues of $\tilde{T}$ generally provide good starting points for the Ehrlich–Aberth iteration for $p(\lambda) = \det(T - \lambda I)$.

First we consider the case where T_1 and T_2 have a common eigenvalue and then the case where either T_1 or T_2 has a multiple eigenvalue. In this regard, we observe that if a matrix $\tilde{A}$ has an eigenvalue λ of geometric multiplicity at least 2, then λ is an eigenvalue of $A = \tilde{A} + uv^T$ for any vectors u, v . Indeed, denote by x_1 and x_2 two linearly independent eigenvectors of $\tilde{A}$ corresponding to λ and set $y = \nu_1 x_1 + \nu_2 x_2$; then it holds that $Ay = \lambda y + u(v^T y)$. Therefore, choosing ν_1 and ν_2 such that $\nu_1 v^T x_1 + \nu_2 v^T x_2 = 0$ with $\nu_1 \neq 0$ or $\nu_2 \neq 0$ yields $Ay = \lambda y$, $y \neq 0$. This implies the following result.

THEOREM 3.1. *If λ is a common eigenvalue of T_1 and T_2 with geometric multiplicity at least 2, then λ is an eigenvalue of $T = (T_1 \oplus T_2) + uv^T$ with geometric multiplicity at least 2. Observe that under these assumptions λ is an eigenvalue of $T_1 \oplus T_2$ with geometric multiplicity at least 2. Therefore λ is an eigenvalue of $T_1 \oplus T_2 + uv^T$ for any vector u, v . $\square$*

Our next theorem relates the eigenvalues of T_1 and T_2 with those of $T_1 \oplus T_2 + uv^T$ and relies on the following lemma from Henrici [22].

LEMMA 3.2. *If $p(\lambda) \neq 0$ and $p'(z) \neq 0$ for $z = \lambda$, then $n|p(z)/p'(\lambda)| \leq |p(\lambda)|$.*

THEOREM 3.3. *Let $T_1 \in \mathbb{R}^{m \times m}$ and $T_2 \in \mathbb{R}^{(n-m) \times (n-m)}$ be tridiagonal matrices. Let $X_1 = X_2 = I$, $D_1 = \text{diag}(d_1, d_2, \dots, d_m)$, $D_2 = \text{diag}(d_{m+1}, d_{m+2}, \dots, d_n)$, and $\beta_m, \gamma_m \in \mathbb{R}$. Then*

$$(3.3) \quad \eta_j = \begin{cases} \beta_m(e_j^T X_1^{-1} e_m)(e_m^T X_1 e_j) & , j \leq m \\ \gamma_m(e_{j-m}^T X_2^{-1} e_1)(e_1^T X_2 e_{j-m}) & , j > m \end{cases}$$

are the eigenvalues of $T = (T_1 \oplus T_2) + uv^T$, where

$$\rho_j = \begin{cases} n|\eta_j| / \left| 1 + \sum_{\substack{k=1 \\ k \neq j}}^n \frac{\eta_j + \eta_k}{d_k - d_j} \right| & , d_j \neq d_k, \quad k \neq j, \\ 0 & , d_j = d_k, \quad k \neq j, \end{cases}$$

with $u = e_m + e_{m+1}$ and $v = \beta_m e_m + \gamma_m e_{m+1}$.

... If $d_j = d_k$ for some $k \neq j$, then the result holds from Theorem 3.1. Consider the case $d_j \neq d_k$ for $k \neq j$. Let $\tilde{T} = T_1 \oplus T_2$. We have

$$T - \lambda I = (\tilde{T} - \lambda I)(I + (\tilde{T} - \lambda I)^{-1}uv^T),$$

and taking determinants on both sides of the equation gives

$$p(\lambda) = \prod_{j=1}^n (d_j - \lambda)(1 + v^T(\tilde{T} - \lambda I)^{-1}u).$$

Using the eigendecomposition of T_1 and T_2 and the definition of u and v , the expression for $p(\lambda)$ simplifies to

$$(3.4) \quad p(\lambda) = \prod_{j=1}^n (d_j - \lambda) + \sum_{j=1}^n \eta_j \prod_{k=1, k \neq j}^n (d_k - \lambda).$$

Thus

$$p(d_j) = \eta_j \prod_{\substack{k=1 \\ k \neq j}}^n (d_k - d_j),$$

and

$$p'(\lambda) = - \sum_{k=1}^n \prod_{\substack{h=1 \\ h \neq k}}^n (d_h - \lambda) - \sum_{\ell=1}^n \eta_\ell \left(\sum_{\substack{k=1 \\ k \neq \ell}}^n \prod_{\substack{h=1 \\ h \neq \ell, k}}^n (d_h - \lambda) \right)$$

so that

$$p'(d_j) = - \prod_{\substack{k=1 \\ k \neq j}}^n (d_k - d_j) \left(1 + \sum_{\substack{k=1 \\ k \neq j}}^n \frac{\eta_j + \eta_k}{d_k - d_j} \right).$$

Since $p'(d_j) \neq 0$, applying Lemma 3.2 completes the proof. $\square$

According to Theorem 3.3, a small value for ρ_j indicates that there is an eigenvalue of $T_1 \oplus T_2$ close to an eigenvalue of T . An important question related to the effectiveness of using the eigenvalues of T_1 and T_2 as initial approximations for starting the Ehrlich–Aberth iteration is whether the number of “large” values for ρ_j is small or not. Note that if $j \leq m$, then η_j is a multiple of the product of the last components of the j th right and left eigenvectors of T_1 and, if $j > m$, then η_j is a multiple of the product of the first components of the $(j - m)$ th left and right eigenvectors of T_2 . If T is unreduced, then η_j is nonzero since the eigenvectors of unreduced tridiagonal matrices cannot have a 0 in the first and last components. We report in Table 3.1 ranges of values for ρ_j and $|\eta_j|$ obtained from 1000 randomly generated tridiagonal matrices T of size $n = 100$. The table shows that in more than 80% of the cases, $|\eta_j|$ and ρ_j are smaller than 10^{-4} . Note that the denominator $|1 + \sum_{k=1, k \neq j}^n \frac{\eta_j + \eta_k}{d_k - d_j}|$ in the definition of ρ_j does not seem to play an important role. The probability that for almost all the values of j this denominator is close to zero seems to be small. These experiments suggest that most eigenvalues of $T_1 \oplus T_2$ should be good initial values for the Ehrlich–Aberth iteration for $p(\lambda) = \det(T - \lambda I)$.

TABLE 3.1

Ranges of values for ρ_j and $|\eta_j|$ obtained from 1000 randomly generated tridiagonal matrices T of size $n = 100$.

%	$\leq 10^{-16}$	$\leq 10^{-12}$	$\leq 10^{-8}$	$\leq 10^{-4}$
ρ_j	48	60	71	82
$ \eta_j $	54	65	77	88

From the results of this section we propose the following divide-and-conquer strategy to compute initial approximations for the Ehrlich–Aberth iterations. The matrix T is recursively split according to the rank-1 tearing in (3.1)–(3.2) until 2×2 or 1×1 subproblems are reached. The Ehrlich–Aberth iteration is then used to glue back the subproblems using the previously computed eigenvalues as starting guesses for the iterations.

3.4. A similar divide-and-conquer strategy can be obtained by choosing, as initial approximations, the eigenvalues of the leading principal $m \times m$ submatrix T_1 of T and of the trailing principal $(n - m) \times (n - m)$ submatrix T_2 of T for $m = \lfloor n/2 \rfloor$. These matrices are obtained by zeroing the entries in positions $(m, m + 1)$ and $(m + 1, m)$ of T which correspond to applying a rank-2 correction to the matrix T . An analysis similar to the one performed for the rank-1 tearing can be carried out. In all the numerical experiments we performed so far no substantial difference in the performance of the two strategies was observed.

4. The algorithm. In this section we describe an implementation of the Ehrlich–Aberth iteration where the choice of the initial approximations is performed by means of a divide-and-conquer strategy. Then we provide running error bounds needed for the validation of the computed approximations and discuss the computation of the eigenvectors.

The main algorithm for eigenvalue approximation is described below in pseudocode. This implementation follows section 3. A different implementation can be based on the rank-2 tearing of Remark 3.4, where the initial approximations are the eigenvalues of the principal submatrices obtained by zeroing the entries in positions $(m, m + 1)$ and $(m + 1, m)$, where $m = \lfloor n/2 \rfloor$.

First we report on the recursive part of the algorithm and then the main refinement engine, i.e., the Ehrlich–Aberth iteration.

```

function  $z = \text{eigen}(\beta, \alpha, \gamma)$ 
% Computes the eigenvalues of the  $n \times n$  tridiagonal matrix  $T = \text{tridiag}(\beta, \alpha, \gamma)$ 
% perturb is a small scalar set to the maximum relative perturbation
% of intermediate eigenvalues
if  $n = 1$ ,  $z = \alpha_1$ , return, end
if  $n = 2$ , set  $z$  to the eigenvalues of  $\begin{bmatrix} \alpha_1 & \gamma_1 \\ \beta_1 & \alpha_2 \end{bmatrix}$ , return, end
% Recursive stage
 $i = \sqrt{-1}$ 
if  $n > 2$ 
     $m = \lfloor n/2 \rfloor$ 
     $\alpha' = \alpha(1 : m)$ ,  $\alpha'_m = \alpha'_m - \beta_m$ 
     $z(1 : m) = \text{eigen}(\beta(1 : m - 1), \alpha', \gamma(1 : m - 1))$ 
     $\alpha'' = \alpha(m + 1 : n)$ ,  $\alpha''_1 = \alpha''_1 - \gamma_m$ 
     $z(m + 1 : n) = \text{eigen}(\beta(m + 1 : n - 1), \alpha'', \gamma(m + 1 : n - 1))$ 
    Choose random  $\rho$  such that  $\text{perturb}/2 < \rho < \text{perturb}$ 

```

```

 $z(1:m) = (1 + i\rho) * z(1:m)$ 
 $z(m+1:n) = (1 - i\rho) * z(m+1:n)$ 
% Refine the current approximations
 $z = \text{Aberth}(\beta, \alpha, \gamma, z)$ 
end

```

The role of *perturb* is to avoid different approximations collapsing to a single value. In fact, if $z_j = z_k$, with $j \neq k$, then the Ehrlich–Aberth iteration cannot be applied. Moreover, since in practice the Ehrlich–Aberth iteration performs better if the intermediate approximations are in the complex field, we have chosen a pure imaginary value as perturbation. In our code we set the parameter *perturb* to 10 times the machine precision.

The pseudocode for the Ehrlich–Aberth iteration is reported below.

```

function  $z = \text{Aberth}(\beta, \alpha, \gamma, z)$ 
% Refine the  $n$  approximations  $z = (z_j)$ ,  $j = 1:n$  to the eigenvalues of
%  $T = \text{tridiag}(\beta, \alpha, \gamma)$  by means of the Ehrlich–Aberth iteration.
% maxit is the maximum number of iterations,
% tol is a small quantity used for the convergence criteria.
it = 0
 $\zeta = \text{zeros}(n, 1)$  %  $\zeta_j = 1$  if  $z_j$  has converged to an eigenvalue and 0 otherwise.
while ( $\sum_{j=1}^n \zeta_j < n$  &  $it < \text{maxit}$ )
    it = it + 1
    for  $j = 1:n$ 
        if  $\zeta_j = 0$ 
             $\tau = \text{trace\_Tinv}(\beta, \alpha - z_j, \gamma)$ 
            if  $\tau \neq \text{Inf}$ 
                 $nwtc = 0$ 
            else
                 $nwtc = -1/\tau$ 
            end
            if  $|nwtc| < \text{tol} \|T - z_j I\|_\infty$ ,  $\zeta_j = 1$ , end
             $z_j = z_j - nwtc / (1 - nwtc \sum_{k=1, k \neq j}^n \frac{1}{z_j - z_k})$ 
        end
    end
end
end

```

Observe that at the general *it*th sweep, the Ehrlich–Aberth correction is applied only to the approximations z_j which have not yet converged. This makes the cost of each sweep depend on the number of approximated eigenvalues. For this reason, in order to evaluate the complexity of the algorithm, it is convenient to introduce the *complexity index* μ given by the overall number of Ehrlich–Aberth iterations applied to each eigenvalue divided by n . For example, if $n = 4$ and the number of iterations needed for computing $\lambda_1, \lambda_2, \lambda_3, \lambda_4$ is 5, 10, 14, 21, respectively, then $\mu = 50/4 = 12.5$.

4.1. Running error bound. In this section we derive a running error bound for the error in the computed eigenvalues λ_ℓ , $\ell = 1:n$. Our bounds are based on the following result of Carstensen [9].

LEMMA 4.1. *Let $p(\lambda) = \prod_{\ell=1}^n (\lambda - \lambda_\ell)$ be a polynomial with roots $\lambda_1, \lambda_2, \dots, \lambda_n$. Let ρ_ℓ be a positive number such that $\rho_\ell > \max_{j \neq \ell} |\lambda_\ell - \lambda_j|$. Then the error in the computed eigenvalue λ_ℓ is bounded by*

$$\rho_\ell = \frac{n|p(\lambda_\ell)|}{\left| \prod_{\substack{j=1 \\ j \neq \ell}}^n (\lambda_\ell - \lambda_j) \right|}.$$

$$\mathcal{U} = \bigcup_{\ell=1}^n \mathcal{D}(\lambda_\ell, \rho_\ell), \quad p(\lambda) \in \mathcal{U}.$$

The set of disks $\{\mathcal{D}(\lambda_\ell, \rho_\ell), \ell = 1:n\}$ defined in the above lemma is called a *set of inclusion disks*. Note that because of rounding errors in the computation of $|p(\lambda_\ell)|$ and $\prod_{j=1, j \neq \ell}^n (\lambda_\ell - \lambda_j)$, the computed ρ_ℓ denoted by $\hat{\rho}_\ell$ may be inaccurate. Thus, the disks $\mathcal{D}(\lambda_\ell, \hat{\rho}_\ell)$ may not provide a set of inclusion disks.

Consider the standard model of floating point arithmetic [25, section 2.2],

$$(4.1) \quad fl(x \text{ op } y) = (x \text{ op } y)(1 + \delta), \quad |\delta| \leq u, \quad \text{op} = +, -, *, /,$$

where u is the unit roundoff. Since λ can be complex, part of the computation is carried out in complex arithmetic. Under the standard model (4.1) (see [25, Lemma 3.5]),

$$(4.2) \quad \begin{aligned} fl(x \pm y) &= (x \pm y)(1 + \delta), \quad |\delta| \leq u, \\ fl(xy) &= xy(1 + \delta), \quad |\delta| \leq 2\sqrt{2}u, \\ fl(x/y) &= (x/y)(1 + \delta), \quad |\delta| \leq 4\sqrt{2}u, \end{aligned}$$

where we ignore second order terms in u .

Suppose we can compute an upper bound Δp_ℓ for the error $||\hat{p}_\ell| - |p(\lambda_\ell)||$, where $|\hat{p}_\ell| = fl(|p(\lambda_\ell)|)$. Then from Lemma 4.1 we immediately deduce that the disk $\mathcal{D}(\lambda_\ell, \tilde{\rho}_\ell)$, where

$$(4.3) \quad \tilde{\rho}_\ell = (1 + 2nu) \frac{n(|\hat{p}_\ell| + \Delta p_\ell)}{\hat{\pi}_\ell}, \quad \hat{\pi}_\ell = fl \left(\left| \prod_{\substack{j=1 \\ j \neq \ell}}^n (\lambda_\ell - \lambda_j) \right| \right),$$

is such that $\mathcal{D}(\lambda_\ell, \rho_\ell) \subset \mathcal{D}(\lambda_\ell, \tilde{\rho}_\ell)$, $\ell = 1:n$; therefore the set $\{\mathcal{D}(\lambda_\ell, \tilde{\rho}_\ell), \ell = 1:n\}$ is a set of inclusion disks.

The main difficulty is in determining Δp_ℓ . Recall that $|p(\lambda_\ell)| = \prod_{j=1}^n |r_j|$, where r_j is the j th diagonal element of R in the QR factorization of $T - \lambda_\ell I$. Let $fl(r_j) = r_j + \delta r_j$. Then

$$fl \left(\prod_{j=1}^n r_j \right) = \prod_{j=1}^n (r_j + \delta r_j) \cdot \prod_{j=1}^{n-1} (1 + \epsilon_j), \quad |\epsilon_j| \leq 2\sqrt{2}u,$$

and, if we ignore the second order terms in u , we have $|\hat{p}_\ell| = |p(\lambda_\ell)| + \delta p_\ell$, with

$$(4.4) \quad |\delta p_\ell| \leq |\hat{p}_\ell| \left(\sum_{j=1}^n \Delta r_j / |\hat{r}_j| + (n-1)2\sqrt{2}u \right) =: \Delta p_\ell.$$

The Δr_j are computed along with the $\hat{r}_j = fl(r_j)$ thanks to a systematic running error analysis of all the quantities involved in the calculation of r_j . For that we make use of the following lemma.

LEMMA 4.2. $x = yz + \alpha v \in \mathbb{C}$, $\alpha \in \mathbb{C}$, $y = \hat{y} + \delta y$, $|\delta y| \leq \Delta y$, $z = \hat{z} + \delta z$, $|\delta z| \leq \Delta z$, $v = \hat{v} + \delta v$, $|\delta v| \leq \Delta v$, $\hat{y}, \hat{z}, \hat{v}, \Delta y, \Delta z, \Delta v$ are non-negative real numbers. Then $\hat{x} = fl(x) = x + \delta x$, $|\delta x| \leq \Delta x := |\hat{y}| \Delta z + |\hat{z}| \Delta y + 2\sqrt{2}u|\hat{y}||\hat{z}| + |\alpha| \Delta v + 2\sqrt{2}u|\alpha||\hat{v}|$.

$$|\delta x| \leq \Delta x := |\hat{y}| \Delta z + |\hat{z}| \Delta y + 2\sqrt{2}u|\hat{y}||\hat{z}| + |\alpha| \Delta v + 2\sqrt{2}u|\alpha||\hat{v}|.$$

The proof is a straightforward application of (4.1) and (4.2). $\square$

We notice that in the function `trace_Tinv` (see section 2.2) all the quantities used to compute r_i can be rewritten in the form $x = yz + \alpha v$. Hence, assuming that the function `Givens` returns $\Delta\phi$ and $\Delta\psi$, we can add the lines

$$\begin{aligned} \Delta r_j &= |\phi| \Delta a + |a| \Delta\phi + 2\sqrt{2}u|a||\phi| + |\beta_j| \Delta\psi + 2\sqrt{2}u|\beta_j||\psi|, \\ \Delta a &= |\psi| \Delta g + |g| \Delta\psi + 2\sqrt{2}u|\psi||g| + |\alpha_{j+1}| \Delta\phi + 2\sqrt{2}u|\alpha_{j+1}||\phi|, \\ \Delta g &= |\gamma_{j+1}| \Delta\phi + 2\sqrt{2}u|\gamma_{j+1}||\phi| \end{aligned}$$

to the function `trace_Tinv` after the computation of r_j , a , and g , respectively. The two quantities Δa and Δg are initially set to zero and $\Delta r_n = \Delta a$. The error bound Δp_ℓ is then obtained using (4.4).

4.2. Computing the eigenvectors. One of the most convenient methods for approximating an eigenvector of T once we are given an approximation λ of the corresponding eigenvalue is the inverse power iteration applied to the matrix $T - \lambda I$. A crucial computational issue is to determine a suitable initial guess $v^{(0)}$ for the eigenvector in order to start the iteration:

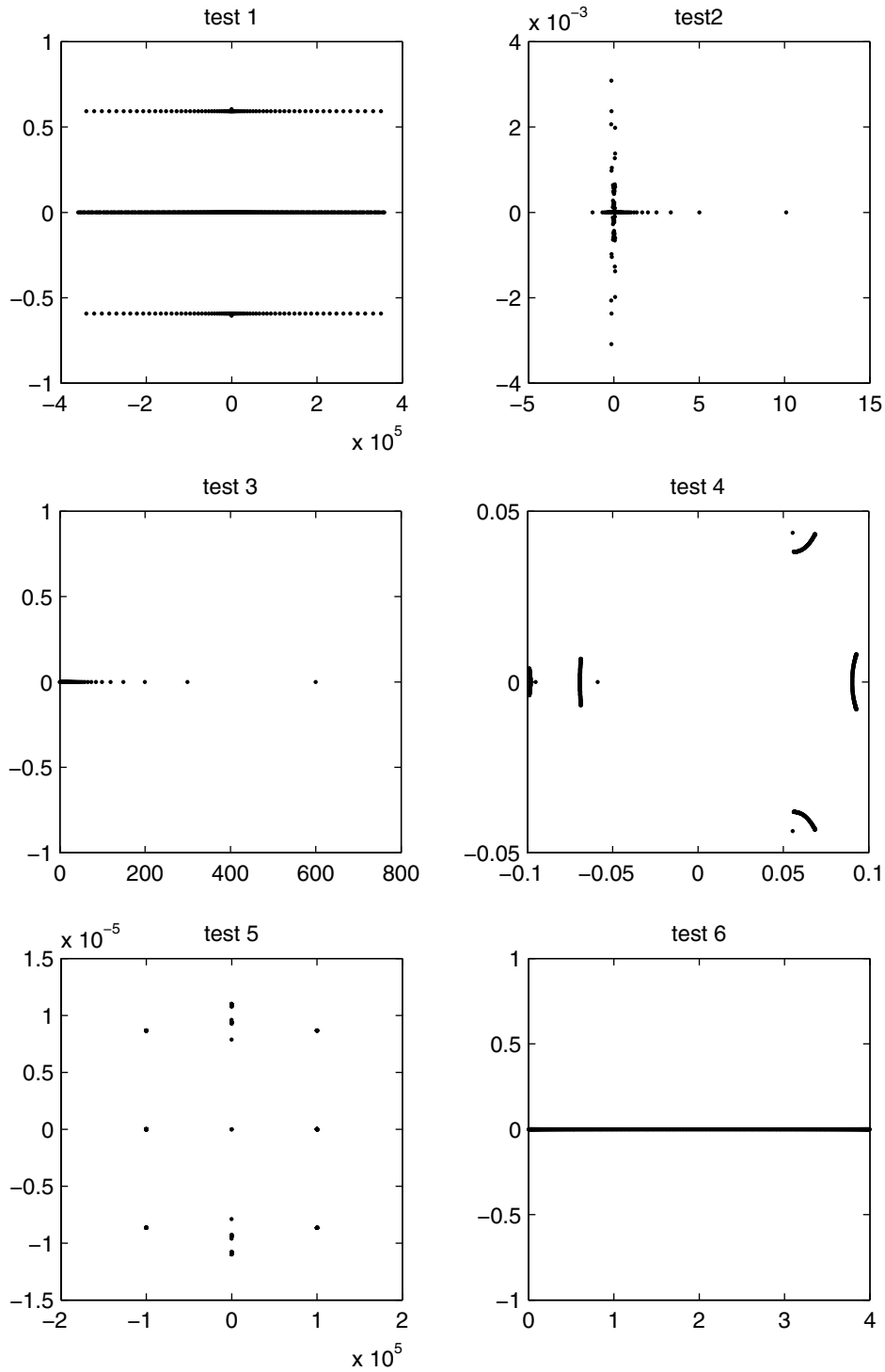
$$\begin{aligned} (T - \lambda I)w^{(j+1)} &= v^{(j)}, \\ v^{(j+1)} &= w^{(j+1)} / \|w^{(j+1)}\|, \end{aligned} \quad j = 0, 1, 2, \dots$$

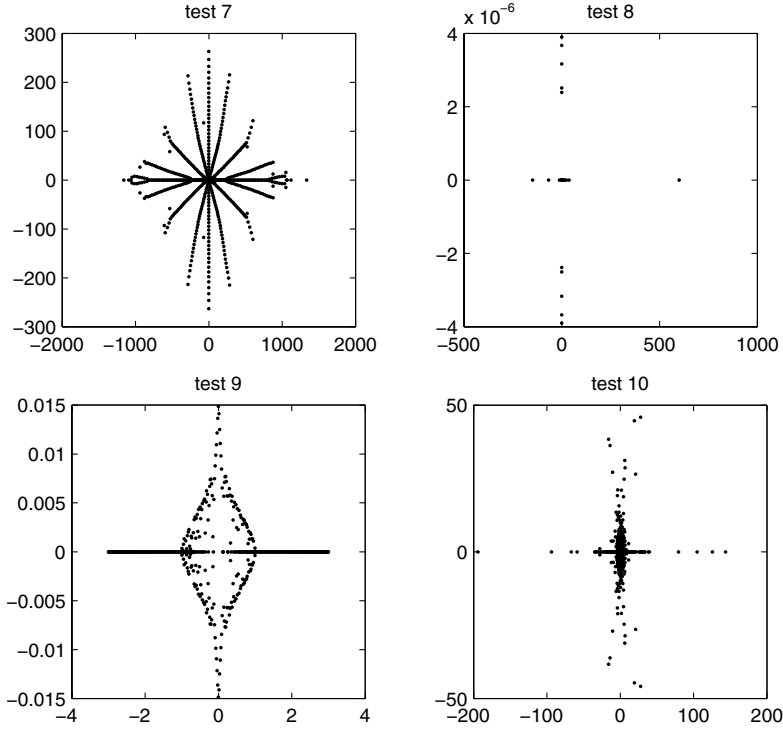
This problem has been studied in several recent papers [11], [16], [32]. In particular, in [16] a strategy is described for the choice of $v^{(0)}$ which relies on the evaluation of the index k of the entry of maximum modulus in the main diagonal of $(T - \lambda I)^{-1}$. Our algorithm for the approximation of the eigenvalues provides, as a by-product, the diagonal entries of $(T - \lambda I)^{-1}$. Therefore the value of k is determined at no cost. Moreover, the QR factorization of the matrix $T - \lambda I$ that is computed by our algorithm can be used for performing each inverse power iteration without any significant additional cost.

5. Numerical experiments. We have implemented the algorithm for the approximation of the eigenvalues of the tridiagonal matrix T in Fortran 95. The code, organized as a Fortran 95 module, can be downloaded from http://www.dm.unipi.it/~bini/eigen_v1.1.tgz. The tests were performed on an Athlon 1800 with IEEE double precision arithmetic, where the code has been compiled with the Lahey/Fujitsu compiler (release L6.20a). In what follows, **eigen** refers to our subroutine implementing the Ehrlich–Aberth iteration.

We first consider the following test matrices, which we describe in the factored form

$$(5.1) \quad T = D^{-1} \text{tridiag}(1, \alpha, 1), \quad D = \text{diag}(\delta), \quad \alpha, \delta \in \mathbb{R}^n.$$


 FIG. 5.1. Eigenvalues of the test matrices 1-6 for $n = 300$.

FIG. 5.2. *Eigenvalues of the test matrices 7–10 for $n = 300$.*

- Test 1: $\alpha_k = k(-1)^{\lfloor k/8 \rfloor}, \delta_k = (-1)^k/k, k = 1:n$.
 Test 2: $\alpha_k = 10 \cdot (-1)^{\lfloor k/8 \rfloor}, \delta_k = k \cdot (-1)^{\lfloor k/9 \rfloor}, k = 1:n$.
 Test 3: $\alpha_k = k, \delta_k = n - k + 1, k = 1:n$.
 Test 4: $\alpha_k = (-1)^k, \delta_k = 20 \cdot (-1)^{\lfloor k/5 \rfloor}, k = 1:n$.
 (5.2) Test 5: $\alpha_k = 10^{5(-1)^k} \cdot (-1)^{\lfloor k/4 \rfloor}, \delta_k = (-1)^{\lfloor k/3 \rfloor}, k = 1:n$.
 Test 6: $\alpha_k = 2, \delta_k = 1, k = 1:n$.
 Test 7: $\alpha_k = \frac{1}{k} + \frac{1}{n - k + 1}, \delta_k = \frac{1}{k}(-1)^{\lfloor k/9 \rfloor}, k = 1:n$.
 Test 8: $\alpha_k = k \cdot (-1)^{\lfloor k/13 \rfloor + \lfloor k/5 \rfloor}, \delta_k = (n - k + 1)^2 \cdot (-1)^{\lfloor k/11 \rfloor}, k = 1:n$.
 Test 9: $\alpha_k = 1, k = 1:n; \delta_k = 1$ if $k < n/2$, $\delta_k = -1$ if $k \geq n/2$.
 Test 10: α_k and δ_k take random values uniformly distributed in $[-0.5, 0.5]$.

The eigenvalues of these test matrices have a variety of distributions, as illustrated in Figures 5.1 and 5.2 for $n = 300$. In particular, the eigenvalues in Test 4 are distributed along curves and, in Test 7, along lines from the origin; in Test 5 the eigenvalues are distributed in tight clusters with very different moduli.

Let

$$(5.3) \quad \kappa(\lambda) = \frac{\|x\|_2 \|y\|_2 \|T\|_2}{|\lambda| |y^T x|}$$

TABLE 5.1

Largest eigenvalue condition number $\kappa_{\max}$ and largest relative error for the eigenvalues of the test matrices 1–9 computed with **dhseqr** and **eigen** ($n = 100$).

Test matrices	1	2	3	4	5	6	7	8	9
$\kappa_{\max} = \max_i \kappa(\lambda_i)$	3e4	240	4e4	2.2	2e10	4e3	2e3	4e6	213
$\max_i \frac{ \lambda_i - \hat{\lambda}_i }{ \lambda_i }$	dhseqr	2e-13	6e-15	5e-15	4e-15	3e-6	3e-13	3e-14	7e-15
	eigen	3e-16	2e-16	2e-16	2e-16	1e-10	2e-14	6e-16	5e-16

TABLE 5.2

Condition numbers and relative errors for the three clusters of eigenvalues in Test 5, with $n = 20$.

	$\kappa(\lambda)$	$\max_i \lambda_i - \hat{\lambda}_i / \lambda_i $	
		dhseqr	eigen
$\lambda \approx -10^5$	1.2	3e-16	8e-18
$\lambda \approx 10^5$	2e2	1e-13	1e-14
$ \lambda \approx 10^{-5}$	1e10	4e-7	1e-16

be the 2-norm condition number of λ with corresponding right eigenvector x and left eigenvector y . Note that $\kappa(\lambda) \geq 1$ for symmetric matrices. As shown in Table 5.1 the test matrices 1–9 have eigenvalue condition numbers varying from small to large.

To validate our code, we report in Table 5.1 the size of the largest relative error

$$(5.4) \quad \text{err}(\lambda_i) = |\lambda_i - \hat{\lambda}_i|/|\lambda_i|.$$

Here λ_i is computed in quadruple precision and $\hat{\lambda}_i$ denotes the computed eigenvalue with either our subroutine **eigen** or with the LAPACK subroutine **dhseqr** that implements the QR algorithm for computing the eigenvalues of an upper Hessenberg matrix. For these test matrices the eigenvalues computed by **eigen** are, in general, more accurate than those computed by **dhseqr** and do not seem to be affected as much by large eigenvalue condition numbers. Also, **eigen** tends to compute eigenvalues with small moduli more accurately than **dhseqr**. This is better illustrated in Table 5.2 for the matrix in Test 5 with $n = 20$. This matrix has three groups of clustered eigenvalues: two around $\pm 10^5$ and one around 0. Both **dhseqr** and **eigen** compute the eigenvalues around -10^5 to high accuracy. For the eigenvalues with small modulus, the LAPACK routine yields relative errors as large as 10^{-7} , whereas **eigen** computes eigenvalues with relative errors of the order of the machine precision.

The Clement matrix $T = \text{tridiag}(\beta, 0, \gamma)$ with $\beta_j = \gamma_{n-j}$ and $\gamma_j = j$, $j = 1:n-1$ [10], has eigenvalues plus and minus the numbers $n-1, n-3, \dots, 1$ for n even and $n-1, n-3, \dots, 0$ for n odd. We take $n = 50$. The condition number $\kappa(\lambda)$ in (5.3) is small at the ends of the spectrum and large in the middle, as illustrated in the left-hand plot in Figure 5.3. This explains the relative errors in the right-hand plot. On these plots, the dotted lines mark the machine precision. Note that all the eigenvalues computed by **eigen** have a relative error around 10^{-16} or below, whereas those computed by **dhseqr** all have a relative error significantly above 10^{-16} . In this case, our algorithm computes exactly the real part of the eigenvalues, which are integers. The imaginary part is approximated with an absolute error which is less than 10^{-25} for almost all the eigenvalues. Note that the Clement matrices are essentially symmetric; i.e., they can be transformed by a similarity to a symmetric form by means of diagonal scalings

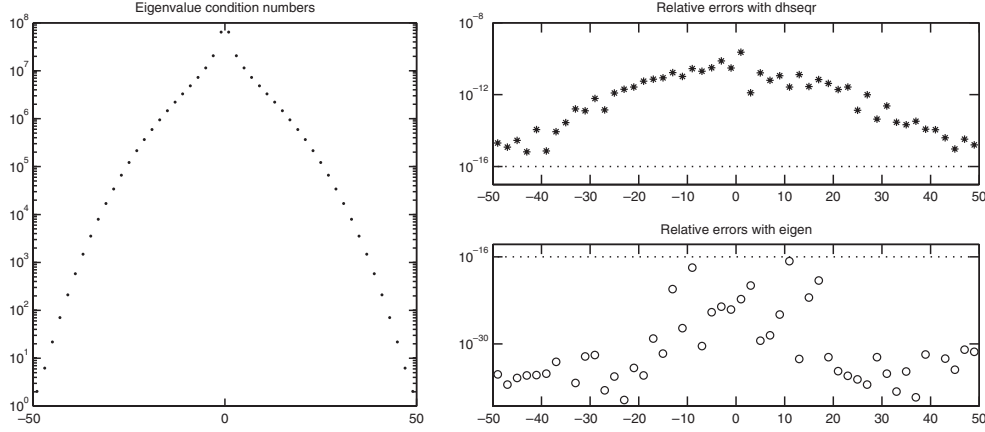


FIG. 5.3. *Clement matrix of size $n = 50$; on the x-abcissa are the n eigenvalues $\{-49, -47, \dots, -1, 1, \dots, 47, 49\}$.*

since $\beta_i \gamma_i \geq 0$. The eigenvalues of the symmetric matrices obtained in this way can be determined to high relative accuracy [28].

The Bessel matrices associated with generalized Bessel polynomials [33] are non-symmetric tridiagonal matrices defined by $T_{(a,b)} = \text{tridiag}(\beta, \alpha, \gamma)$ with

$$\begin{aligned} \alpha_1 &= -\frac{b}{a}, \quad \beta_1 = \frac{\gamma_1}{a+1}, \quad \gamma_1 = -\beta_1, \\ \alpha_j &= -b \frac{a-2}{(2j+a-2)(2j+a-4)}, \quad j = 2:n, \\ \beta_j &= -b \frac{j}{(2j+a-1)(2j+a-2)}, \quad \gamma_j = b \frac{j+a-2}{(2j+a-2)(2j+a-3)}, \quad j = 2:n-1. \end{aligned}$$

The spectrum is given in Figure 5.4 for $a = -8.5$, $b = 2$, and $n = 18$. These matrices are well known to have ill-conditioned eigenvalues. For this particular example, $\kappa(\lambda)$ ranges from 10^{10} to 10^{15} . The approximations provided by **eigen** are more accurate than those provided by **dhseqr** by a factor of 10 on average.

We tested Liu's tridiagonal pseudosymmetric matrices $T_n = \text{tridiag}(1, \alpha^n, \gamma^n)$ for $n = 14$ and $n = 28$ defined by [29]

$$\begin{aligned} \alpha^{14} &= [0, 0, 0, 0, 0, 0, -1, -1, 0, 0, 0, 0, 0, 0]^T, \\ \gamma^{14} &= [-1, 1, 1, -1, 1, -1, -1, -1, 1, -1, 1, 1, -1]^T \end{aligned}$$

and

$$\begin{aligned} \alpha^{28} &= [0, 0, 0, 0, 0, 0, -1, 1, 0, 0, 0, 0, 0, -1, 1, 0, 0, 0, 0, 0, 1, -1, 0, 0, 0, 0, 0, 0]^T, \\ \gamma^{28} &= [-1, 1, 1, -1, 1, -1, -1, -1, 1, -1, 1, 1, -1, -1, -1, \\ &\quad 1, 1, -1, 1, -1, -1, -1, 1, -1, 1, 1, -1]^T. \end{aligned}$$

Both matrices have only one zero eigenvalue of multiplicity 14 and 28, respectively. As shown in Figure 5.5, the accuracy of the approximations delivered by **eigen** are as good as those provided by **dhseqr**.

Summarizing the results concerning the approximation errors, we may say that in all our tests **eigen** generally yields errors no larger, and sometimes much smaller,

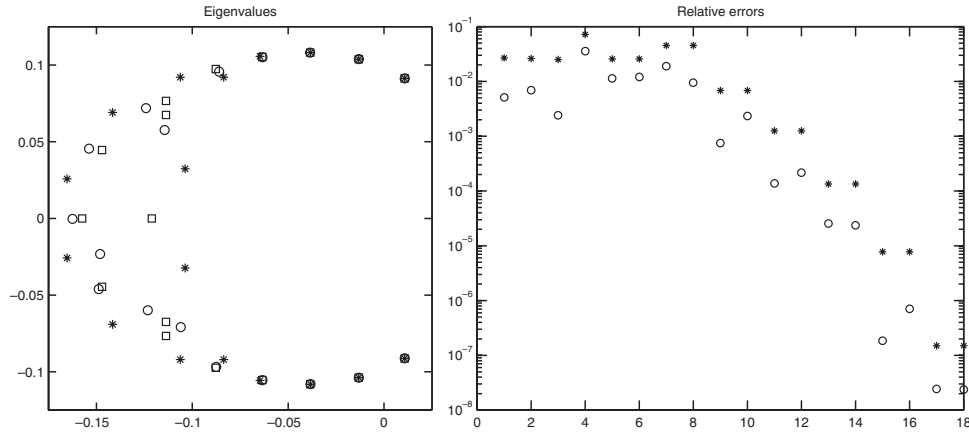


FIG. 5.4. *Bessel matrix*, $n = 18$, $a = -8.5$, $b = 2$. Eigenvalues computed in quadruple precision ($\square$), and approximations to zero eigenvalues provided by **eigen** ($\circ$) and by the LAPACK subroutine **dhseqr** (*).

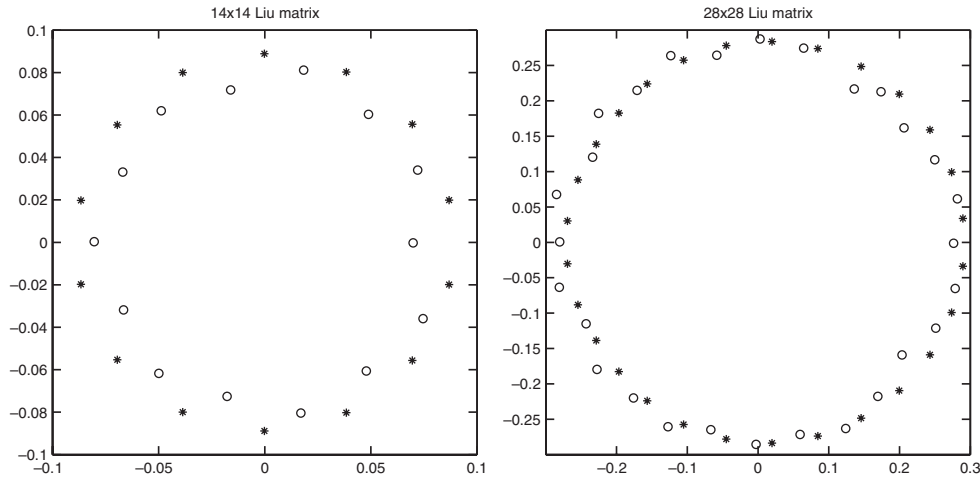


FIG. 5.5. *Liu's matrices*. Approximations to zero eigenvalues provided by **eigen** ($\circ$) and by the LAPACK subroutine **dhseqr** (*).

than the LAPACK subroutine. The difference of the errors is more evident for the eigenvalues of small modulus.

To illustrate our running error bounds $\tilde{\rho}$ in (4.3) we consider two matrices:

1. The tridiagonal matrix defined in factored form (5.1) by

$$\alpha_j = 10^{6(-1)^{j+1}}, \quad \delta_j = 1, \quad j \leq n/2, \quad \delta_j = -1, \quad j > n/2$$

and for which the eigenvalues are grouped into five clusters.

2. Liu's matrix T_{14} modified with $\alpha_6^{14} = 2^{-5}$ and $\alpha_{14}^{14} = 2^{-20}$ so that all its eigenvalues are distinct.

Note that these two matrices are exactly represented in base 2. In Tables 5.3 and 5.4 we report, for each eigenvalue λ , the relative error for its approximation by **eigen** and our running error bound $\tilde{\rho}$ (4.3). We also report the structured condition number

TABLE 5.3

Bounds $\tilde{\rho}$ on the relative error for the eigenvalues of T : $\alpha_j = 10^{6(-1)^{j+1}}$, $\delta_j = 1$, $j \leq n/2$, $\delta_j = -1$, $j > n/2$, $n = 10$.

λ	$\frac{ \lambda - \hat{\lambda} }{ \lambda }$	$\tilde{\rho}$	$\kappa_s(\lambda)$	$\eta_s(\lambda)$
-1.0e+6	0	5e-23	0.7	2e-22
-1.0e+6	0	2e-24	0.7	2e-22
-1.7e-6	7e-9	7e-8	5e11	2e-20
-7.4e-7	1e-4	1e-3	1e12	2e-16
2.2e-7 + i5.0e-7	4e-6	4e-5	1e12	5e-18
2.2e-7 - i5.0e-7	4e-10	4e-9	1e12	8e-22
2.0e-6	4e-6	4e-5	4e11	1e-17
1.0e+6	0	1e-16	0.7	3e-16
1.0e+6	0	1e-1	0.7	1e-16
1.0e+6	0	8e-17	0.7	2e-16

TABLE 5.4

Bounds $\tilde{\rho}$ on the relative error: Liu's matrix T_{14} , modified with $\alpha_6 = 2^{-5}$ and $\alpha_{14} = 2^{-20}$.

λ	$\frac{ \lambda - \hat{\lambda} }{ \lambda }$	$\tilde{\rho}$	$\kappa_s(\lambda)$	$\eta_s(\lambda)$
-7.2e-1	8e-16	5e-12	1e2	2e-15
-6.8e-1 ± i 3.5e-1	2e-15	4e-12	7e1	2e-15
-3.4e-1 ± i 6.9e-1	3e-15	4e-12	6e1	2e-14
-6.1e-3 ± i 7.1e-1	1e-15	5e-12	5e1	1e-14
-5.5e-3	2e-11	1e-8	4e6	1e-16
5.5e-3	9e-11	2e-8	4e6	1e-16
3.6e-1 ± i 6.8e-1	3e-15	4e-12	6e1	9e-15
6.9e-1 ± i 3.6e-1	1e-15	4e-12	6e1	2e-15
7.1e-1	3e-15	6e-12	1e2	4e-16

$\kappa_s(\lambda)$ and the structured backward error $\eta_s(\lambda)$ for which we impose the perturbations to be real tridiagonal [23]. The product $\kappa_s(\lambda)\eta_s(\lambda)$ provides a first order bound on the forward error $|\lambda - \hat{\lambda}|/|\lambda|$. The results in Tables 5.3 and 5.4 confirm that the running error bound is a strict upper bound and show that for these two matrices it exceeds the error by 1 to 3 orders of magnitude. It is important to notice that computing $\tilde{\rho}$ is inexpensive, requiring just $O(n)$ flops per eigenvalue. On the other hand, the upper bound $\kappa_s(\lambda)\eta_s(\lambda)$ requires the knowledge of the right and left eigenvectors and is usually expensive to compute when structure in the perturbations is imposed. The small values for $\eta_s(\lambda)$ show that on these examples **eigen** is structured backward stable.

Some performance tests have been carried out with n ranging from 200 to 6400 on a PC with an Athlon 1800 CPU. Table 5.5 reports the CPU time in seconds required by **eigen** versus the time required by the LAPACK subroutine **dhseqr**. These timings show that in all the cases the growth of the CPU time required by our algorithm is a quadratic function of n , whereas the cost of **dhseqr** grows cubically with n . The threshold value for which our algorithm is faster than the LAPACK subroutine is about $n = 400$ for Tests 1, 2, 3, 8, 10, $n = 800$ for Tests 5, 7, 9, and $n = 1600$ for Tests 4 and 6. The speed-up reached for $n = 6400$ is in the range 6.5–189.9.

For all the test matrices with the exception of Tests 4 and 6 the algorithm converges quickly: the average number of iterations per eigenvalue required in the last

TABLE 5.5
CPU time in seconds for **eigen** (in boldface) versus LAPACK's **dhseqr**.

n	200	400	800	1600	3200	6400
Test 1	0.1 /0.06	0.4 /0.6	1.4 /4.8	5.6 /76	24 /661	103 /6209
Test 2	0.1 /0.10	0.4 /0.8	1.2 /7.5	4.5 /56	19 /315	82 /2378
Test 3	0.1 /0.07	0.3 /0.7	1.1 /6.7	4.3 /152	18 /1510	80 /15188
Test 4	0.9 /0.07	3.1 /0.6	13.2 /4.3	54 /75.6	249 /896	1100 /6092
Test 5	0.2 /0.10	1.1 /0.9	4.9 /6.8	21 /135	84 /1239	360 /11088
Test 6	0.8 /0.08	2.9 /0.6	13.2 /5.4	54 /93	251 /744	1128 /7351
Test 7	0.3 /0.10	0.9 /0.8	3.1 /6.7	12.0 /124	49 /1460	198 /9155
Test 8	0.08 /0.08	0.3 /0.6	0.9 /4.0	4.0 /49.9	17 /382	76 /2838
Test 9	0.4 /0.11	1.6 /0.9	6.1 /8.2	27 /175	114 /1450	543 /13250
Test 10	0.2 /0.08	0.6 /0.7	2.1 /6.0	7.7 /130	32 /975	137 /8990

TABLE 5.6
Average number of iterations per eigenvalue and maximum number of iterations (in parentheses) of the Ehrlich–Aberth method.

n	200	400	800	1600	3200	6400
Test 1	1.9 (3)	1.9 (26)	1.9 (21)	1.8 (20)	1.8 (21)	1.8 (19)
Test 2	1.9 (5)	1.8 (14)	1.5 (4)	1.5 (3)	1.5 (6)	1.5 (6)
Test 3	1.9 (4)	1.6 (4)	1.5 (4)	1.5 (3)	1.5 (6)	1.5 (6)
Test 4	21.7 (26)	16.8 (26)	19.5 (26)	18.0 (26)	19.0 (50)	18.1 (29)
Test 5	4.8 (17)	7.1 (27)	7.8 (28)	7.4 (25)	5.8 (27)	5.8 (36)
Test 6	22.6 (26)	16.2 (24)	21.5 (27)	18.7 (27)	19.7 (26)	18.9 (25)
Test 7	4.6 (10)	3.9 (10)	3.5 (11)	3.3 (14)	3.2 (17)	2.6 (19)
Test 8	1.4 (3)	1.4 (3)	1.4 (3)	1.4 (3)	1.4 (3)	1.4 (2)
Test 9	5.9 (14)	5.9 (13)	5.8 (15)	5.9 (22)	5.8 (17)	5.7 (21)
Test 10	2.7 (7)	2.4 (7)	2.3 (8)	2.1 (12)	2.1 (9)	2.0 (9)

recursive step is between 1.9 and 5.9 (see Table 5.6). In Test 6, despite the eigenvalues being real since the matrix is real symmetric, the average number of iterations per eigenvalue is larger than in the other tests for which the eigenvalues are complex. In general, the Ehrlich–Aberth iteration does not take any advantage of the reality of the eigenvalues; moreover the convergence speed is not improved by choosing initial real approximations, and the best choice remains to select initial approximations along a suitable circle in the complex plane [21]. The average number of Ehrlich–Aberth iterations per eigenvalue seems to be almost independent of n and varies according to the specific problem.

6. Conclusion. We have introduced an algorithm for the computation of the Newton quotient $p(\lambda)/p'(\lambda)$ for $p(\lambda) = \det(T - \lambda I)$, and T a tridiagonal matrix, based on the QR factorization of $T - \lambda I$ and on the semiseparable structure of $(T - \lambda I)^{-1}$. The algorithm, whose arithmetic cost is linear in the size n of the matrix T , is robust.

This algorithm was used for implementing the Ehrlich–Aberth iteration, which approximates all eigenvalues of T . Besides the straightforward way of choosing the initial approximations in the unit circle, a more elaborate strategy for the choice of the initial approximations was proposed. This strategy is based on a divide-and-conquer technique, which, even though heuristic, is motivated by some inclusion results that we have proved in section 3.2. Running error bounds for the errors in the computed eigenvalues were also provided.

The numerical experiments, performed with a large set of test matrices, confirmed the effectiveness of the algorithm. Comparisons with the LAPACK subroutine `dhseqr` showed that for moderately large values of n our algorithm is faster. In particular, the Ehrlich–Aberth iteration has a cost which is $O(n^2)$, whereas the LAPACK subroutine costs $O(n^3)$ operations. Moreover, the Ehrlich–Aberth iteration generally yields more accurate computed eigenvalues than the LAPACK subroutine `dhseqr`.

Acknowledgments. We thank Inderjit Dhillon and the referees for valuable suggestions that improved the paper, and Sven Hammarling for pointing out that Theorem 2.1, which we had obtained independently, appears in [19].

REFERENCES

- [1] O. ABERTH, *Iteration methods for finding all zeros of a polynomial simultaneously*, Math. Comp., 27 (1973), pp. 339–334.
- [2] L. ADAMS AND P. ARBENZ, *Towards a divide and conquer algorithm for the real nonsymmetric eigenvalue problem*, SIAM J. Matrix Anal. Appl., 15 (1994), pp. 1333–1353.
- [3] D. BINDEL, J. DEMMEL, W. KAHAN, AND O. MARQUES, *On computing Givens rotations reliably and efficiently*, ACM Trans. Math. Software, 28 (2002), pp. 206–238.
- [4] D. A. BINI, *Numerical computation of polynomial zeros by means of Aberth’s method*, Numer. Algorithms, 13 (1996), pp. 179–200.
- [5] D. A. BINI AND G. FIORENTINO, *MPSSolve: Numerical computation of polynomial roots*, v. 2.2, FRISCO report, 1999. Software and papers available online from <ftp://ftp.dm.unipi.it/pub/mpsolve>.
- [6] D. A. BINI AND G. FIORENTINO, *Design, analysis, and implementation of a multiprecision polynomial rootfinder*, Numer. Algorithms, 23 (2000), pp. 127–173.
- [7] M. A. BREBNER AND J. GRAD, *Eigenvalues of $Ax = \lambda Bx$ for real symmetric matrices A and B computed by reduction to a pseudosymmetric form and the HR process*, Linear Algebra Appl., 43 (1982), pp. 99–118.
- [8] A. BUNSE-GERSTNER, *An analysis of the HR algorithm for computing the eigenvalues of a matrix*, Linear Algebra Appl., 35 (1981), pp. 155–173.
- [9] C. CARSTENSEN, *Inclusion of the roots of a polynomial based on Gerschgorin’s theorem*, Numer. Math., 59 (1991), pp. 349–360.
- [10] P. A. CLEMENT, *A class of triple-diagonal matrices for test purposes*, SIAM Rev., 1 (1959), pp. 50–52.
- [11] I. S. DHILLON, *Reliable computation of the condition number of a tridiagonal matrix in $O(n)$ time*, SIAM J. Matrix Anal. Appl., 19 (1998), pp. 776–796.
- [12] I. DHILLON, *A New $O(n^2)$ Algorithm for the Symmetric Tridiagonal Eigenvalue/Eigenvector Problem*, Ph.D. thesis, UCB Tech. Report UCB//CSD-97-971, University of California, Berkeley, CA, 1997.
- [13] J. J. DONGARRA, G. A. GEIST, AND C. H. ROMINE, *Algorithm 710: FORTRAN subroutines for computing the eigenvalues and eigenvectors of a general matrix by reduction to general tridiagonal form*, ACM Trans. Math. Software, 18 (1992), pp. 392–400.
- [14] Q. DU, M. JIN, T. Y. LI, AND Z. ZENG, *The quasi-Laguerre iteration*, Math. Comp., 66 (1997), pp. 345–361.
- [15] L. W. EHRLICH, *A modified Newton method for polynomials*, Comm. ACM, 10 (1967), pp. 107–108.
- [16] K. V. FERNANDO, *On computing an eigenvector of a tridiagonal matrix. Part I: Basic results*, SIAM J. Matrix Anal. Appl., 18 (1997), pp. 1013–1034.
- [17] I. GARGANTINI, *Parallel Laguerre iterations: The complex case*, Numer. Math., 26 (1976), pp. 317–323.
- [18] G. A. GEIST, *Reduction of a general matrix to tridiagonal form*, SIAM J. Matrix Anal. Appl., 12 (1991), pp. 362–373.
- [19] P. E. GILL, G. H. GOLUB, W. MURRAY, AND M. A. SAUNDERS, *Methods for modifying matrix factorizations*, Math. Comp., 28 (1974), pp. 505–535.
- [20] G. H. GOLUB AND C. F. VAN LOAN, *Matrix Computations*, 3rd ed., Johns Hopkins University Press, Baltimore, MD, 1996.
- [21] H. W. GUGGENHEIMER, *Initial approximations in Durand-Kerner’s root finding method*, BIT, 26 (1986), pp. 537–539.
- [22] P. HENRICI, *Applied and Computational Complex Analysis*, Vol. 1, Wiley, New York, 1974.

- [23] D. J. HIGHAM AND N. J. HIGHAM, *Structured backward error and condition of generalized eigenvalue problems*, SIAM J. Matrix Anal. Appl., 20 (1998), pp. 493–512.
- [24] N. J. HIGHAM, *Efficient algorithms for computing the condition number of a tridiagonal matrix*, SIAM J. Sci. Statist. Comput., 7 (1986), pp. 150–165.
- [25] N. J. HIGHAM, *Accuracy and Stability of Numerical Algorithms*, 2nd ed., SIAM, Philadelphia, 2002.
- [26] Y. IKEBE, *On inverses of Hessenberg matrices*, Linear Algebra Appl., 24 (1979), pp. 93–97.
- [27] E. R. JESSUP, *A case against a divide and conquer approach to the nonsymmetric eigenvalue problem*, Appl. Numer. Math., 12 (1993), pp. 403–420.
- [28] W. KAHAN, *Accurate Eigenvalues of a Symmetric Tri-diagonal Matrix*, Tech. Report CS-TR-66-41, Stanford University, Stanford, CA, 1966.
- [29] Z. S. LIU, *On the Extended HR Algorithm*, Tech. Report PAM-564, Center for Pure and Applied Mathematics, University of California, Berkeley, CA, 1992.
- [30] B. N. PARLETT, *The Symmetric Eigenvalue Problem*, SIAM, Philadelphia, 1998.
- [31] B. N. PARLETT AND H. C. CHEN, *Use of indefinite pencils for computing damped natural modes*, Linear Algebra Appl., 140 (1990), pp. 53–88.
- [32] B. N. PARLETT AND I. S. DHILLON, *Fernando's solution to Wilkinson's problem: An application of double factorization*, Linear Algebra Appl., 267 (1997), pp. 247–279.
- [33] L. PASQUINI, *Accurate computation of the zeros of the generalized Bessel polynomials*, Numer. Math., 86 (2000), pp. 507–538.
- [34] M. S. PETKOVIĆ, *Iterative Methods for Simultaneous Inclusion of Polynomial Zeros*, Springer-Verlag, Berlin, Heidelberg, New York, 1989.
- [35] M. S. PETKOVIĆ, L. D. PETKOVIĆ, AND L. RANČIĆ, *Higher-order simultaneous methods for the determination of polynomial multiple zeros*, Int. J. Comput. Math., 80 (2000), pp. 1407–1427.
- [36] M. S. PETKOVIĆ, L. D. PETKOVIĆ, AND D. ŽIVKOVIĆ, *Laguerre-like methods for the simultaneous approximation of polynomial zeros*, in Topics in Numerical Analysis, Comput. Suppl. 15, Springer-Verlag, Vienna, 2001, pp. 189–209.
- [37] M. S. PETKOVIĆ AND D. V. VRANIĆ, *The convergence of Euler-like method for the simultaneous inclusion of polynomial zeros*, Comput. Math. Appl., 39 (2000), pp. 95–105.
- [38] H. RUTISHAUSER, *Solution of eigenvalue problems with the LR-transformation*, Nat. Bur. Standards Appl. Math. Ser., 1958 (1958), pp. 47–81.
- [39] F. TISSEUR, *Tridiagonal-diagonal reduction of symmetric indefinite pairs*, SIAM J. Matrix Anal. Appl., 26 (2004), pp. 215–232.
- [40] F. TISSEUR AND K. MEERBERGEN, *The quadratic eigenvalue problem*, SIAM Rev., 43 (2001), pp. 235–286.

AN ORTHOGONAL SIMILARITY REDUCTION OF A MATRIX INTO SEMISEPARABLE FORM*

M. VAN BAREL[†], R. VANDEBRIL[†], AND N. MASTRONARDI[‡]

Abstract. An algorithm to reduce a symmetric matrix to a similar semiseparable one of semiseparability rank 1, using orthogonal similarity transformations, is proposed in this paper.

It is shown that, while running to completion, the proposed algorithm gives information on the spectrum of the similar initial matrix. In fact, the proposed algorithm shares the same properties of the Lanczos method and the Householder reduction to tridiagonal form.

Furthermore, at each iteration, the proposed algorithm performs a step of the QR method without shift to a principal submatrix to retrieve the semiseparable structure. The latter step can be considered a kind of subspace-like iteration method, where the size of the subspace increases by one dimension at each step of the algorithm. Hence, when during the execution of the algorithm the Ritz values approximate the dominant eigenvalues closely enough, diagonal blocks will appear in the semiseparable part where the norm of the corresponding subdiagonal blocks goes to zero in the subsequent iteration steps, depending on the corresponding gap between the eigenvalues.

A numerical experiment is included, illustrating the properties of the new algorithm.

Key words. similarity transformation, semiseparable matrix, Lanczos algorithm, Ritz values

AMS subject classifications. 15A18, 15A21, 65F15

DOI. 10.1137/S0895479803428229

1. Introduction. Numerical linear algebra problems involving semiseparable matrices have raised much interest in the last decade. In particular, some fast and stable algorithms to compute the eigendecomposition of symmetric diagonal plus semiseparable matrices have been developed [2, 3, 6, 11].

In this paper an algorithm that transforms a symmetric matrix $A \in \mathbb{R}^{n \times n}$ into a similar symmetric semiseparable one by means of orthogonal transformations with $\frac{4}{3}n^3 + \mathcal{O}(n^2)$ flops¹ is proposed. Hence, the latter algorithm, combined with one of those for computing the eigendecomposition of semiseparable matrices, could give an alternative way to compute the eigenvalue decomposition of symmetric matrices.

There are excellent algorithms that compute the complete eigendecomposition of a dense symmetric matrix in the literature. The standard procedure to compute

*Received by the editors May 22, 2003; accepted for publication (in revised form) by D. A. Bini October 8, 2004; published electronically July 13, 2005.

<http://www.siam.org/journals/simax/27-1/42822.html>

[†]Department of Computer Science, Katholieke Universiteit Leuven, Celestijnenlaan 200A, B-3001 Leuven (Heverlee), Belgium (Raf.Vandebril@cs.kuleuven.ac.be, Marc.VanBarel@cs.kuleuven.ac.be). The research of the first two authors was supported by the Research Council K. U. Leuven, project OT/00/16 (SLAP: Structured Linear Algebra Package); by the Fund for Scientific Research–Flanders (Belgium), projects G.0078.01 (SMA: Structured Matrices and Their Applications), G.0176.02 (AN-CILA: Asymptotic Analysis of the Convergence Behavior of Iterative Methods in Numerical Linear Algebra), and G.0184.02 (CORFU: Constructive Study of Orthogonal Rational Functions); by the K.U. Leuven (Bijzonder Onderzoeksfonds); and by the Belgian Programme on Interuniversity Poles of Attraction, initiated by the Belgian State, Prime Minister’s Office for Science, Technology and Culture, project IUAP V-22 (Dynamical Systems and Control: Computation, Identification & Modelling). The scientific responsibility rests with the authors.

[‡]Istituto per le Applicazioni del Calcolo “M. Picone,” sez. Bari, Consiglio Nazionale delle Ricerche, Via G. Amendola, 122/I, I-70126 Bari, Italy (irmanm21@area.ba.cnr.it). The research of this author was partially supported by MIUR, grant 2002014121.

¹In this paper the definition given in [7] is adopted: “A flop is a floating point operation.” The square root is considered a flop, too.

the complete eigendecomposition of a dense symmetric matrix first reduces it to a similar symmetric tridiagonal one by means of orthogonal transformations. This step is accomplished in $\frac{4}{3}n^3 + \mathcal{O}(n^2)$ flops. Once the similar symmetric tridiagonal matrix has been computed, many fast and stable algorithms can be considered to compute the spectral decomposition of the latter matrix (see, for instance, [7, 15, 13, 5, 9, 12] and the references therein).

The computation of the spectral decomposition of dense symmetric matrices is beyond the scope of this paper. However, the proposed algorithm for reducing a symmetric matrix into a semiseparable one, while running to completion, gives information on the spectrum of the similar initial matrix. In fact, similarly to the Lanczos method [5, 7] and the Householder reduction to tridiagonal form [10], [7, pp. 472–473], after k steps of the algorithm the eigenvalues of the partially reduced matrix are the Ritz values of the original matrix with respect to the Krylov subspace $\{e_n, Ae_n, \dots, A^{k-1}e_n\}$, where e_n is the last vector of the canonical basis of $\mathbb{R}^n$.

Furthermore, at the k th iteration, the proposed algorithm performs a step of the QR method without shift to the principal submatrix of order $k+1$ to retrieve the semiseparable structure. The latter step can be considered a kind of subspace-like iteration method, where the size of the subspace increases by one dimension at each step of the algorithm. Hence, when, during the execution of the algorithm, the dominant eigenvalues are approximated sufficiently well by the Lanczos–Ritz values, diagonal blocks will appear in the semiseparable part. The norm of the corresponding subdiagonal blocks will go to zero depending on the corresponding gap between the eigenvalues. Therefore it turns out that the proposed algorithm could be chosen when an approximation of the subspace associated to the largest eigenvalues of large dense matrices is required. Such problems arise in many different areas, such as principal component analysis, data mining, magnetic resonance spectroscopy, microarray data analysis, gene expression data, and computation of the eigensolutions of the Schrödinger equation on a grid (see, for instance, [1, 8, 20, 21, 24, 25] and the references therein).

The paper is organized as follows. In section 2 the definition and possible stable representations of semiseparable matrices are discussed. The algorithm for reducing symmetric matrices to similar symmetric semiseparable ones is described in section 3. The unsymmetric case is considered briefly in section 4. In section 5 the computational complexity is discussed. It also contains an algorithm for reducing symmetric tridiagonal matrices into similar symmetric semiseparable ones. The strong connection with the Lanczos–Ritz values is investigated in section 6. Convergence properties of the subspace iteration are studied in section 7. A numerical experiment is shown in section 8, followed by the conclusions.

2. Definition and representation of semiseparable matrices. In the literature the class of semiseparable matrices is defined in slightly different ways. For an overview of the various definitions, representations, and properties, we refer the interested reader to [19, 14]. In this paper, semiseparable matrices are defined in the following way.

DEFINITION 2.1. A matrix $S \in \mathbb{R}^{n \times n}$ is called *semiseparable* if there exist matrices $R \in \mathbb{R}^{n \times r}$ and $S \in \mathbb{R}^{r \times n}$ such that $S = RS^T$ and $\|R\|_F \leq r$ and $\|S\|_F \leq r$.

In the remainder of the paper, we will consider only symmetric semiseparable matrices of (lower- and upper-)semiseparability rank 1, referring to them as semiseparable matrices.

In [19], it is shown that under weak assumptions, the matrix S can be represented by two (column-)vectors u and v , called generators of the matrix S :

$$S = \text{triu}(uv^T) + \text{tril}(vu^T),$$

where $\text{triu}(A)$ and $\text{tril}(A)$ denote, respectively, the strictly upper triangular part and the lower triangular part of the matrix A .

THEOREM 2.2. Let S be a symmetric matrix of dimension n with the following properties:

$$(2.1) \quad \begin{aligned} & S_{i,j} = 0 \text{ for } i < j \text{ and } i > n-j+1, \\ & S_{i,j} = 0 \text{ for } 1 \leq j \leq i \leq n, \text{ if } S(i,j) = 0 \Rightarrow S(i,1:i) = 0, \text{ } S(j:n,j) = 0 \end{aligned}$$

The proof can be found in [19]. $\square$

Because the representation with the generators is cheap in terms of memory usage and it is easy to reconstruct arbitrary elements within the matrix, several algorithms for semiseparable matrices are based on this representation. For several of these algorithms, this representation can lead to a big loss of accuracy when performing the computations in finite precision. This can be seen from the following example.

1. Suppose we have a symmetric 5×5 matrix with the following eigenvalues: $(1, 2, 3, 100, 10^4)$. Constructing a semiseparable matrix from it (the algorithm used to do this will be explained later on) generates the following matrix:

$$S = \begin{pmatrix} 1.6254 & -5.0812 \cdot 10^{-1} & -8.0163 \cdot 10^{-2} & -6.3532 \cdot 10^{-5} & 5.2384 \cdot 10^{-11} \\ -5.0812 \cdot 10^{-1} & 1.4259 & 2.2496 \cdot 10^{-1} & 1.7829 \cdot 10^{-4} & -1.4597 \cdot 10^{-10} \\ -8.0163 \cdot 10^{-2} & 2.2496 \cdot 10^{-1} & 2.9485 & 2.3368 \cdot 10^{-3} & -1.9152 \cdot 10^{-9} \\ -6.3532 \cdot 10^{-5} & 1.7829 \cdot 10^{-4} & 2.3368 \cdot 10^{-3} & 9.9999 \cdot 10^1 & -8.1957 \cdot 10^{-5} \\ 5.2384 \cdot 10^{-11} & -1.4597 \cdot 10^{-10} & -1.9152 \cdot 10^{-9} & -8.1957 \cdot 10^{-5} & 1.0000 \cdot 10^4 \end{pmatrix}.$$

Although this matrix is semiseparable, it can clearly be seen that the lower right element has already converged to the largest eigenvalue. Representing this matrix now with the generators u and v gives us the following vectors: Even though this matrix is semiseparable, it can clearly be seen that the last entry of the diagonal already approximates the largest eigenvalue. Representing this matrix with the traditional generators u and v gives us the following vectors:

$$u = (3.1029 \cdot 10^{10} \quad -9.6998 \cdot 10^9 \quad -1.5302 \cdot 10^9 \quad -1.2128 \cdot 10^6 \quad 1.0000)^T$$

and

$$v = (5.2384 \cdot 10^{-11} \quad -1.4597 \cdot 10^{-10} \quad -1.9152 \cdot 10^{-9} \quad -8.1957 \cdot 10^{-5} \quad 1.0000 \cdot 10^4)^T.$$

Because the first element of v is of the order 10^{-11} and is constructed by summations of elements of order 1, we can expect that this element has a precision of only 5 significant decimal digits left. Using this number to reconstruct the elements within the matrix will give these elements with only a limited number of exact digits.

To overcome this problem we consider another representation. For a semiseparable matrix of dimension n , this representation consists of $n-1$ Givens transformations and a vector of length n . The Givens transformations are denoted as $G = [G_1, \dots, G_{n-1}]$ and the vector as $d = [d_1, \dots, d_n]$.

The following figures denote how the semiseparable matrix can be reconstructed using this information. The elements denoted by $\boxtimes$ make up the semiseparable part of the matrix. The construction of the semiseparable matrix is for a matrix of size 5. Initially one starts on the first two rows of the matrix. The element d_1 is placed in the upper left position, then a Givens transformation is applied, and finally, to complete the first step, element d_2 is added in position $(2, 1)$. Only the first two columns and rows are shown here.

$$\begin{pmatrix} d_1 & 0 \\ 0 & 0 \end{pmatrix} \rightarrow G_1 \begin{pmatrix} d_1 & 0 \\ 0 & 0 \end{pmatrix} + \begin{pmatrix} 0 & 0 \\ 0 & d_2 \end{pmatrix} \rightarrow \begin{pmatrix} \boxtimes & 0 \\ \boxtimes & d_2 \end{pmatrix}.$$

The second step consists of applying the Givens transformation G_2 on the second and the third rows; furthermore d_3 is added in position $(3, 3)$. Here only the first three columns and the second and third rows are shown. This leads to

$$\begin{pmatrix} \boxtimes & d_2 & 0 \\ 0 & 0 & 0 \end{pmatrix} \rightarrow G_2 \begin{pmatrix} \boxtimes & d_2 & 0 \\ 0 & 0 & 0 \end{pmatrix} + \begin{pmatrix} 0 & 0 & 0 \\ 0 & 0 & d_3 \end{pmatrix} \rightarrow \begin{pmatrix} \boxtimes & \boxtimes & 0 \\ \boxtimes & \boxtimes & d_3 \end{pmatrix}.$$

This process can be repeated by applying the Givens transformation G_3 on the third and the fourth rows of the matrix, and afterwards adding the diagonal element d_4 . After applying all the Givens transformations and adding all the diagonal elements, the lower triangular part of a symmetric semiseparable matrix is constructed. Because of the symmetry, the upper triangular part is also known.

Suppose the Givens and vector representation of a semiseparable matrix S is known. Let us denote the Givens transformations by

$$(2.2) \quad G_l = \begin{pmatrix} c_l & -s_l \\ s_l & c_l \end{pmatrix}.$$

The elements $S(i, j)$ with $n > i \geq j$ are calculated in the following way: $S(i, j) = c_i s_{i-1} s_{i-2} \cdots s_j d_j$. When $n = i$ we have $S(i, j) = s_{n-1} s_{n-2} \cdots s_j d_j$. When $n \geq j > i$, $S(i, j)$ can be calculated in a similar way, due to the symmetry. The elements of the semiseparable matrix can therefore be calculated in a stable way.

Example 2 (Example 1 continued). The Givens-vector representation of the matrix in Example 1 is the following (in the first row of G the elements $c_1, \dots, c_4$ are stored and in the second row the elements $s_1, \dots, s_4$):

$$G = \begin{pmatrix} 0.9534 & 0.9878 & 1.0000 & 1.0000 \\ -0.3017 & 0.1558 & 0.0008 & 0.0000 \end{pmatrix}$$

and

$$d = (1.7049 \quad 1.4435 \quad 2.9485 \quad 9.9999 \cdot 10 \quad 1.0000 \cdot 10^4).$$

Moreover this representation has another advantage connected to the QR factorization of this semiseparable matrix. Let $\hat{G}_j$ represent the Givens transformation G_j embedded in an identity matrix of size $n \times n$ such that the Givens transformation G_j is executed on rows j and $j + 1$. Then, as was proved in [17], the QR factorization of S can be written as

$$\begin{aligned} S &= \hat{G}_{n-1} \cdots \hat{G}_2 \hat{G}_1 R \\ &= QR \end{aligned}$$

with R upper triangular.

3. Orthogonal similarity transformation to reduce a symmetric matrix to a semiseparable one.

3.1. Theory. An algorithm to transform a symmetric matrix into a semiseparable one using an orthogonal similarity transformation is presented in this section. The constructive proof of the next theorem provides the algorithm.

THEOREM 3.1. *Let A be a symmetric matrix. Then there exists an orthogonal matrix U such that*

$$(3.1) \quad U^T A U = S,$$

where S is a symmetric semiseparable matrix.

The proof is a constructive one and is made by induction on the rows of matrix A . We will construct a similar symmetric semiseparable matrix from a symmetric one. Let $A_0^{(0)} \equiv A$. Let $G_i^{(k)}$ be the Givens transformation, such that the product $A_{i-1}^{(k)} G_i^{(k)}$ has the entry $(n-k, i)$ annihilated, modifying the i th and the $(i+1)$ th columns of $A_{i-1}^{(k)}$.

Step 1 We will start by constructing a similarity transformation which will make the last two rows (columns) linearly dependent in the lower (upper) triangular part. To this end, we multiply $A_0^{(0)}$ to the left by $G_1^{(0)T}$ and to the right by $G_1^{(0)}$ to annihilate the elements in positions $(1, n)$ and $(n, 1)$ in $A_0^{(0)}$, respectively:

$$\begin{aligned} & \begin{array}{c} \downarrow \quad \downarrow \\ \rightarrow \left(\begin{array}{ccccc} \times & \times & \cdots & \times & \otimes \\ \times & \ddots & \vdots & \vdots & \times \\ \cdots & \cdots & \ddots & \vdots & \vdots \\ \times & \cdots & \cdots & \ddots & \times \\ \otimes & \times & \cdots & \times & \times \end{array} \right) \end{array} \xrightarrow{G_1^{(0)T} A_0^{(0)} G_1^{(0)}} \begin{array}{c} \left(\begin{array}{ccccc} \times & \times & \cdots & \times & 0 \\ \times & \ddots & \vdots & \vdots & \times \\ \cdots & \cdots & \ddots & \vdots & \vdots \\ \times & \cdots & \cdots & \ddots & \times \\ 0 & \times & \cdots & \times & \times \end{array} \right) \end{array} \\ & \quad \quad \quad \Updownarrow \\ & A_0^{(0)} \xrightarrow{G_1^{(0)T} A_0^{(0)} G_1^{(0)}} A_1^{(0)}. \end{aligned}$$

Note that $\times$ denotes arbitrary elements of the matrix, while $\otimes$ denotes the elements which are annihilated by the orthogonal similarity transformation. Summarizing, we obtain that $A_1^{(0)} \equiv G_1^{(0)T} A_0^{(0)} G_1^{(0)}$.

Continuing this process of annihilating all the elements in the last row (column), except for the element in position $(n, n-1)$ ($(n-1, n)$), we get

$$A_{n-2}^{(0)} = \begin{pmatrix} \times & \times & \cdots & \times & 0 \\ \times & \ddots & \vdots & \vdots & \vdots \\ \cdots & \cdots & \ddots & \times & 0 \\ \times & \cdots & \times & \times & \times \\ 0 & \cdots & 0 & \times & \times \end{pmatrix}.$$

Multiplying $A_{n-2}^{(0)}$ to the left by $G_{n-1}^{(0)T}$, we have the following situation:

$$\begin{aligned} & \begin{matrix} \rightarrow \\ \rightarrow \end{matrix} \begin{pmatrix} \times & \times & \cdots & \times & 0 \\ \times & \ddots & \vdots & \vdots & \vdots \\ \cdots & \cdots & \ddots & \vdots & 0 \\ \times & \cdots & \cdots & \times & \otimes \\ 0 & \cdots & 0 & \times & \times \end{pmatrix} \xrightarrow{G_{n-1}^{(0)T} A_{n-2}^{(0)}} \begin{pmatrix} \times & \times & \cdots & \times & 0 \\ \times & \ddots & \vdots & \vdots & \vdots \\ \cdots & \cdots & \ddots & \times & 0 \\ \boxtimes & \cdots & \boxtimes & \times & 0 \\ \boxtimes & \cdots & \boxtimes & \times & \times \end{pmatrix} \\ & \quad \quad \quad \Updownarrow \\ & A_{n-2}^{(0)} \xrightarrow{G_{n-1}^{(0)T} A_{n-2}^{(0)}} \tilde{A}_{n-2}^{(0)}; \end{aligned}$$

i.e., the last two rows are proportional with the exception of the entries in the last two columns. (To emphasize the semiseparable structure among the rows (columns) we denote by $\boxtimes$ the elements of the matrix belonging to these rows (columns).) Let $\tilde{A}_{n-2}^{(0)} \equiv G_{n-1}^{(0)T} A_{n-2}^{(0)}$. Moreover, multiplying $\tilde{A}_{n-2}^{(0)}$ to the right by $G_{n-1}^{(0)}$, the last two columns become linearly dependent above and on the main diagonal; i.e., they form an uppersemiseparable part. Because of symmetry, the last two rows become linearly dependent below and on the main diagonal and form a lowersemiseparable part:

$$\begin{aligned} & \begin{matrix} \downarrow & \downarrow \\ \begin{pmatrix} \times & \times & \cdots & \times & 0 \\ \times & \ddots & \vdots & \vdots & \vdots \\ \cdots & \cdots & \ddots & \vdots & 0 \\ \boxtimes & \cdots & \boxtimes & \times & 0 \\ \boxtimes & \cdots & \boxtimes & \times & \times \end{pmatrix} \end{matrix} \xrightarrow{\tilde{A}_{n-2}^{(0)} G_{n-1}^{(0)}} \begin{pmatrix} \times & \times & \cdots & \boxtimes & \boxtimes \\ \times & \ddots & \vdots & \vdots & \vdots \\ \cdots & \cdots & \ddots & \boxtimes & \boxtimes \\ \boxtimes & \cdots & \boxtimes & \boxtimes & \boxtimes \\ \boxtimes & \cdots & \boxtimes & \boxtimes & \boxtimes \end{pmatrix} \\ & \quad \quad \quad \Updownarrow \\ & \tilde{A}_{n-2}^{(0)} \xrightarrow{\tilde{A}_{n-2}^{(0)} G_{n-1}^{(0)}} A_{n-1}^{(0)}. \end{aligned}$$

To start the next step, $A_0^{(1)}$ is defined as $A_0^{(1)} \equiv A_{n-1}^{(0)}$.

2 Let $m = n - j$, $1 \leq j < n$. Assume by induction that $A_0^{(m)}$ has the lower (upper) triangular part already semiseparable from row n up to row j (column j to n). We will now prove that we can make the lower (upper) triangular part semiseparable up to row $j - 1$ (column $j - 1$). Let us denote the lower (and upper) triangular elements which form a semiseparable part with $\boxtimes$. So matrix $A_0^{(m)}$ looks like

$$A_0^{(m)} = \begin{pmatrix} \times & \cdots & \times & \boxtimes & \cdots & \boxtimes \\ \vdots & \ddots & \cdots & \vdots & \vdots & \vdots \\ \times & \cdots & \times & \boxtimes & \cdots & \boxtimes \\ \boxtimes & \cdots & \boxtimes & \boxtimes & \cdots & \boxtimes \\ \vdots & \cdots & \cdots & \cdots & \ddots & \vdots \\ \boxtimes & \cdots & \boxtimes & \boxtimes & \cdots & \boxtimes \end{pmatrix} \begin{matrix} \leftarrow 1 \\ \vdots \\ \leftarrow j-1 \\ \leftarrow j \\ \vdots \\ \leftarrow n \end{matrix}.$$

Because the submatrix of $A_0^{(m)}$ consisting of the first $j - 1$ rows and the last $n - j + 1$ columns is already semiseparable, it has rank ≤ 1 . Hence, we can construct

$$\begin{array}{ccc}
\begin{array}{c} j-1 \rightarrow \\ j \rightarrow \end{array} \begin{pmatrix} \times & \times & \cdots & \times & \times & 0 & 0 & \cdots & 0 \\ \times & \times & \cdots & \times & \times & 0 & 0 & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots & \vdots & \vdots & \vdots & \ddots & \vdots \\ \times & \times & \cdots & \times & \times & 0 & 0 & \cdots & 0 \\ \times & \times & \cdots & \times & \times & \boxtimes & \boxtimes & \cdots & \boxtimes \\ 0 & \cdots & 0 & 0 & \times & \boxtimes & \boxtimes & \cdots & \boxtimes \\ 0 & \cdots & 0 & 0 & \times & \boxtimes & \boxtimes & \cdots & \boxtimes \\ \vdots & \vdots & \ddots & \vdots & \vdots & \vdots & \vdots & \ddots & \vdots \\ \vdots & \vdots & \ddots & \vdots & \vdots & \vdots & \vdots & \ddots & \vdots \\ 0 & \cdots & 0 & 0 & \times & \boxtimes & \boxtimes & \cdots & \boxtimes \end{pmatrix} & \xrightarrow{G_{j-1}^{(m)T} A_{j-2}^{(m)}} & \begin{array}{c} j-1 \quad j \\ \downarrow \quad \downarrow \end{array} \begin{pmatrix} \times & \times & \cdots & \times & \times & 0 & 0 & \cdots & 0 \\ \times & \times & \cdots & \times & \times & 0 & 0 & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots & \vdots & \vdots & \vdots & \ddots & \vdots \\ \times & \times & \cdots & \times & \times & 0 & 0 & \cdots & 0 \\ \boxtimes & \cdots & \boxtimes & \times & \times & 0 & 0 & \cdots & 0 \\ \boxtimes & \cdots & \boxtimes & \times & \times & 0 & 0 & \cdots & 0 \\ \boxtimes & \cdots & \boxtimes & \times & \times & 0 & 0 & \cdots & 0 \\ 0 & \cdots & 0 & 0 & \boxtimes & \boxtimes & \cdots & \boxtimes \\ \vdots & \vdots & \ddots & \vdots & \vdots & \vdots & \vdots & \ddots & \vdots \\ \vdots & \vdots & \ddots & \vdots & \vdots & \vdots & \vdots & \ddots & \vdots \\ 0 & \cdots & 0 & 0 & \boxtimes & \boxtimes & \cdots & \boxtimes \end{pmatrix} \\
\\
& \xrightarrow{G_{j-1}^{(m)T} A_{j-2}^{(m)} G_{j-1}^{(m)}} & \begin{pmatrix} \times & \times & \cdots & \times & \boxtimes & \boxtimes & 0 & \cdots & 0 \\ \times & \times & \cdots & \times & \boxtimes & \boxtimes & 0 & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots & \vdots & \vdots & \vdots & \ddots & \vdots \\ \times & \times & \cdots & \times & \boxtimes & \boxtimes & 0 & \cdots & 0 \\ \boxtimes & \cdots & \boxtimes & \times & \boxtimes & \boxtimes & 0 & \cdots & 0 \\ \boxtimes & \cdots & \boxtimes & \times & \boxtimes & \boxtimes & 0 & \cdots & 0 \\ \boxtimes & \cdots & \boxtimes & \times & \boxtimes & \boxtimes & 0 & \cdots & 0 \\ 0 & \cdots & 0 & 0 & 0 & \boxtimes & \boxtimes & \cdots & \boxtimes \\ \vdots & \vdots & \ddots & \vdots & \vdots & \vdots & \vdots & \ddots & \vdots \\ \vdots & \vdots & \ddots & \vdots & \vdots & \vdots & \vdots & \ddots & \vdots \\ 0 & \cdots & 0 & 0 & 0 & \boxtimes & \boxtimes & \cdots & \boxtimes \end{pmatrix}.
\end{array}$$

FIG. 3.1. The transformation $A_{j-1}^{(m)} \equiv G_{j-1}^{(m)T} A_{j-2}^{(m)} G_{j-1}^{(m)}$ makes the first $j-1$ entries of rows (columns) $j-1$ and j proportional.

a Givens transformation $G_1^{(m)}$ such that when we multiply $A_0^{(m)}$ to the left by $G_1^{(m)T}$ all elements $(1, j)$, $(1, j+1)$ up to $(1, n)$ of the first row are annihilated. In a similar way, we can construct the Givens transformations $G_2^{(m)} \cdots G_{j-2}^{(m)}$ which make zero elements in columns $j, j+1, \dots, n$ in rows 2 up to $j-2$. Applying these similarity transformations, we obtain the following matrix:

$$A_{j-2}^{(m)} \equiv G_{j-2}^{(m)T} \cdots G_1^{(m)T} A_0^{(m)} G_1^{(m)} \cdots G_{j-2}^{(m)} = \begin{pmatrix} \times & \cdots & \times & \times & 0 & \cdots & 0 \\ \vdots & \ddots & \vdots & \vdots & \vdots & \vdots & \vdots \\ \times & \cdots & \times & \times & 0 & \vdots & 0 \\ \times & \cdots & \times & \times & \boxtimes & \cdots & \boxtimes \\ 0 & \cdots & 0 & \boxtimes & \boxtimes & \cdots & \boxtimes \\ \vdots & \cdots & \vdots & \cdots & \cdots & \ddots & \vdots \\ 0 & \cdots & 0 & \boxtimes & \boxtimes & \cdots & \boxtimes \end{pmatrix} \begin{array}{l} \leftarrow 1 \\ \vdots \\ \vdots \\ \leftarrow j-1 \\ \leftarrow j \\ \vdots \\ \leftarrow n \end{array}.$$

Since rows $j-1$ and j are proportional for the indexes of columns greater than $j-1$, the Givens transformation $G_{j-1}^{(m)T}$ can also be constructed annihilating all the entries in row $j-1$ with column index greater than $j-1$. Furthermore, the product of $G_{j-1}^{(m)T} A_{j-2}^{(m)}$ to the right by $G_{j-1}^{(m)}$ makes columns $j-1$ and j proportional in the first $j-1$ entries. The latter similarity transformation is depicted in Figure 3.1.

To retrieve the semiseparable structure, below the j th row and to the right of the j th column, $n-j$ more Givens similarity transformations $G_k^{(m)}$, $k = j, j+1, \dots, n-1$, are considered. Each of these is chosen such that when the Givens transformation $G_k^{(m)T}$ is first applied to the left, the elements $k+1, k+2, \dots, n$ are annihilated in row k , using the corresponding elements of row $k+1$. To obtain a similar matrix, we apply $G_k^{(m)}$ also on the right, adding one more row (column) to the semiseparable part. The whole process is summarized in Figure 3.2. $\square$

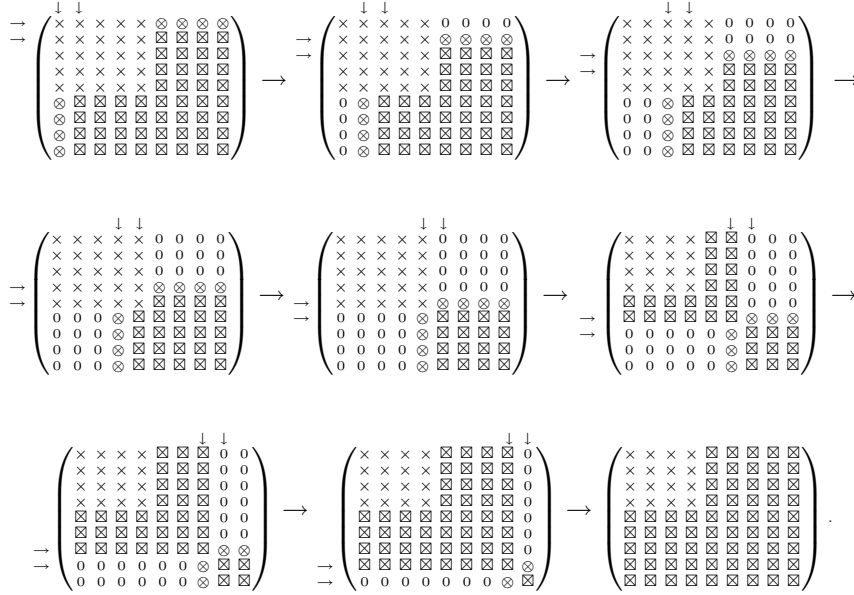


FIG. 3.2. *Description of the similarity transformations used to retrieve the semiseparable structure.*

In the next subsection, details are given on how to implement this algorithm using the Givens-vector representation for the semiseparable part of the matrices involved. This will result in an $O(n^3)$ algorithm to compute the Givens-vector representation of a semiseparable matrix similar to the original symmetric matrix.

Once the semiseparable matrix has been computed, several algorithms can be considered to compute its eigenvalues (see, for instance, [2, 3, 6, 11]).

The proposed algorithm itself, however, while running to completion, already gives information on the spectrum of the initial matrix. In section 6 we will show that the eigenvalues of the subsequent bottom-right semiseparable matrices computed by the proposed algorithm are nothing else but the Ritz values obtained by the Lanczos method.

Moreover, the algorithm can be considered as a nested subspace iteration method, as shown in section 7, where in each step a QL step without shift is performed on the semiseparable part of the reduced matrix.

First, we look at a more detailed implementation of the algorithm in the next subsection. The computational complexity is discussed in section 5.

3.2. Effective implementation of the reduction. Now we describe how to implement the method above using the Givens-vector representation of a semiseparable matrix.

This representation gives information which can be used directly in the QR algorithm applied to the resulting semiseparable matrix [18].

The implementation involves detailed shuffling of indices; therefore only the mathematics behind the implementation is given. The MATLAB files can be downloaded from <http://www.cs.kuleuven.ac.be/~marc>. Following the mathematical details, one should be able to understand and/or reconstruct the algorithm.

Suppose rows $j, j+1, \dots, n$ (columns $j, j+1, \dots, n$) are already semiseparable. We can represent this semiseparable part by a row-vector $R^{(j)} \in \mathbb{R}^{1 \times (j-1)}$, Givens transformations

$$\begin{pmatrix} c_j & -s_j \\ s_j & c_j \end{pmatrix}, \quad k = j, j+1, \dots, n-1,$$

and a vector $[d_j, d_{j+1}, \dots, d_n]$. At this point, the matrix similar to the original symmetric matrix can be divided into four blocks:

$$(3.2) \quad A_0^{(m)} = \left(\begin{array}{cc|ccc} A^{(j)} & c_j R_j^T & c_{j+1} s_j R_j & \cdots & s_{n-1} \dots s_j R_j^T \\ c_j R_j & c_j d_j & c_{j+1} s_j d_j & \cdots & s_{n-1} \dots s_j d_j \\ \hline c_{j+1} s_j R_j & c_{j+1} s_j d_j & c_{j+1} d_{j+1} & & \\ \vdots & & & \ddots & \vdots \\ s_{n-1} \dots s_j R_j & & & \cdots & d_n \end{array} \right),$$

with $m = n - j$ and $A^{(j)} \in \mathbb{R}^{(j-1) \times (j-1)}$.

The first substeps in step m of the method described in the proof of Theorem 3.1 eliminate the elements $1, 2, \dots, j-2$ in row j by multiplying $A_0^{(m)}$ to the right by the Givens transformations $G_1^{(m)}, G_2^{(m)}, \dots, G_{j-2}^{(m)}$, i.e.,

$$R_j G_1^{(m)} G_2^{(m)} \dots G_{j-2}^{(m)} = [0, \dots, 0, \tilde{\alpha}_j] = \tilde{R}_j.$$

It is clear that we can obtain a similar result by applying a Householder transformation $H^{(m)}$ on the row R_j such that the following equation is obtained:

$$R_j H^{(m)} = [0, \dots, 0, \alpha_j] = \hat{R}_j,$$

with $|\alpha_j| = |\tilde{\alpha}_j|$. Performing the similarity Householder transformation on the matrix (3.2) transforms this matrix into the following one:

$$(3.3) \quad \left(\begin{array}{cc|ccc} H^{(m)T} A^{(j)} H^{(m)} & c_j \hat{R}_j^T & c_{j+1} s_j \hat{R}_j^T & \cdots & s_{n-1} \dots s_j \hat{R}_j^T \\ c_j \hat{R}_j & c_j d_j & c_{j+1} s_j d_j & \cdots & s_{n-1} \dots s_j d_j \\ \hline c_{j+1} s_j \hat{R}_j & c_{j+1} s_j d_j & c_{j+1} d_{j+1} & & \vdots \\ \vdots & & & \ddots & \\ s_{n-1} \dots s_j \hat{R}_j & \cdots & & & d_n \end{array} \right).$$

From this point the Givens transformation $G_{j-1}^{(m)}$ can be calculated such that the following equation is satisfied:

$$[\alpha_j, d_j] G_{j-1}^{(m)} = [0, \alpha_{j+1}] \quad \text{with} \quad G_{j-1}^{(m)} = \begin{pmatrix} c_{j-1} & s_{j-1} \\ -s_{j-1} & c_{j-1} \end{pmatrix}.$$

The upper left part $H^{(m)T} A^{(j)} H^{(m)}$ can be written as

$$H^{(m)T} A^{(j)} H^{(m)} = \begin{pmatrix} A^{(j-1)} & R_{j-1}^T \\ R_{j-1} & d_{j-1} \end{pmatrix}.$$

After applying the similarity Givens transformation $G_{j-1}^{(m)}$ on the matrix (3.3), we get the following matrix:

$$(3.4) \quad \begin{pmatrix} A^{(j-1)} & c_{j-1}R_{j-1}^T & s_{j-1}R_{j-1}^T & 0 & \dots & 0 \\ c_{j-1}R_{j-1} & c_{j-1}\hat{d}_{j-1} & s_{j-1}\hat{d}_{j-1} & 0 & \dots & 0 \\ s_{j-1}R_{j-1} & s_{j-1}\hat{d}_{j-1} & c_j\alpha_{j+1} & c_{j+1}s_j\alpha_{j+1} & \dots & s_{n-1}\dots s_j\alpha_{j+1} \\ 0 & 0 & c_{j+1}s_j\alpha_{j+1} & c_{j+1}d_{j+1} & & \\ \vdots & \vdots & \vdots & & \ddots & \vdots \\ 0 & 0 & s_{n-1}\dots s_j\alpha_{j+1} & & \dots & d_n \end{pmatrix},$$

with $\hat{d}_{j-1} = \tilde{d}_{j-1}c_{j-1} + s_{j-1}c_j\alpha_j$. The Givens transformation $G_{j-1}^{(m)T}$ and the element $\hat{d}_{j-1}$ can already be stored as part of the representation of the new semiseparable part, which has to be formed at the bottom of the matrix.

The process explained here can now be repeated to make all rows $j-1, j$ up to n semiseparable, because the matrix (3.4) has essentially the same structure as the matrix (3.3).

The latter step, i.e., the reduction of the lower right part towards the semiseparable structure, can also be applied directly on a semiseparable matrix. This corresponds to performing a QR step (in fact, a QL step) without shift on this semiseparable matrix [18], as will be shown in section 7.

4. The reduction of an arbitrary matrix. It is well known that any matrix can be transformed into an upper Hessenberg one by using an orthogonal similarity transformation. To find all the eigenvalues of the original matrix, the QR algorithm can then be applied to this upper Hessenberg matrix. By using an algorithm similar to the one described in the previous section, it is possible to transform every matrix into the sum of a semiseparable and a strictly upper triangular part. This is proved in the next theorem.

THEOREM 4.1. *Let A be an $n \times n$ matrix. Then there exists a semiseparable matrix S and a strictly upper triangular matrix R such that*

$$A = U(S + R)U^T;$$

where U is an orthogonal matrix and R is a strictly upper triangular matrix.

Following the same reduction as in the proof of Theorem 3.1, we can find an orthogonal matrix U such that the lower triangular part of $U^T A U$ is semiseparable.

Hence, the lower triangular part of A can be written as the lower triangular part of a symmetric semiseparable matrix S , i.e.,

$$\begin{aligned} U^T A U &= \text{tril}(S) + \bar{R} \\ &= S + (\bar{R} - \text{triu}(S)) \\ &= S + R. \quad \square \end{aligned}$$

Note that the algorithm to reduce an arbitrary matrix by means of orthogonal transformations into a similar matrix whose lower triangular part is semiseparable also requires $\mathcal{O}(n^3)$ operations.

5. The computational complexity of each of the algorithms.

5.1. Reduction of a symmetric matrix into a similar semiseparable one.

When reducing a symmetric matrix to a similar semiseparable one, the implementation can be done using only Givens transformations or using Householder and Givens transformations. One can easily see that the complexity can be divided into two parts: first, there is the application of a Householder transformation or a number of Givens transformations on the upper left block of the matrix, and second, there is the updating of the semiseparable part. Executing the first part of all the steps requires the same number of operations as reducing a symmetric matrix into a similar tridiagonal one either by Householders or by Givens transformations. Hence, the computational complexity of this part is

$$\begin{aligned} &2n^3 + O(n^2) \text{ when using Givens transformations,} \\ &\frac{4}{3}n^3 + O(n^2) \text{ when using Householder transformations.} \end{aligned}$$

Executing the second part of all the steps requires $O(n^2)$ operations. More details can be found in the next subsection.

5.2. Reduction to a semiseparable matrix via a tridiagonal matrix.

The third approach to reduce a symmetric matrix to semiseparable form is the following. First, we reduce the matrix to a similar tridiagonal one by standard algorithms. (This can be achieved with Householder or Givens transformations; see, e.g., [7].) The latter tridiagonal matrix can then be reduced into semiseparable form. In this case each iteration step starts with a matrix already of the form (3.3). Hence, the first Householder or Givens transformations do not have to be performed. Note that this algorithm involves essentially the same orthogonal similarity transformations as the algorithms described before. Only the order in which they are applied is different. The results of sections 6 and 7 state, however, that it is worth intertwining the two reductions, instead of first reducing towards tridiagonal form and thereafter to a semiseparable matrix. Hence, the number of operations needed to reduce a tridiagonal into a semiseparable matrix is the same as for the second step in the reduction from a symmetric matrix to semiseparable form.

The calculation of the Givens requires 7 flops, as described in [7]. The multiplication on the left- and right-hand sides takes 11 operations, when everything is optimized. Therefore, the computational complexity is

$$\sum_{j=n:-1:2} \left(\sum_{i=j:1:n} 18 \right) = 9n^2 + O(n).$$

5.3. Reduction of a nonsymmetric matrix. As already mentioned, the algorithm applied to nonsymmetric matrices reduces the matrix to the sum of a semiseparable and a strictly upper triangular matrix. To implement this, similar algorithms to those mentioned above can be used. We now take a look at the complexity of this reduction. Let us assume that we first reduce the matrix to an upper Hessenberg one, and then apply the algorithm to make the lower triangular part semiseparable. (This is a similar algorithm to those applied to tridiagonal matrices.) Then we first have the reduction to Hessenberg form, which costs $5/3n^3 + O(n^2)$ flops. The reduction from Hessenberg to lower semiseparable form costs an extra $3n^3 + O(n^2)$ flops.

5.4. Comparison with the classical reduction algorithms reducing a symmetric matrix to tridiagonal form. When we compare the algorithm which tridiagonalizes a symmetric matrix to the algorithm which constructs a similar semiseparable matrix, we find that the latter algorithm performs only $9n^2 + O(n)$ more flops than the former algorithm, while both algorithms have an overall complexity of $O(n^3)$. It is worthwhile to perform these additional operations as is shown by the convergence properties as studied in the next sections.

6. Connection with the Lanczos–Ritz values. In this section, we show that at each step of the algorithm the eigenvalues of the semiseparable matrix at the bottom right are the Ritz values of the matrix A with respect to the Krylov subspace $\mathcal{K}_m$ (by $\langle x, y, z, \dots \rangle$ we denote the subspace spanned by the vectors $x, y, z, \dots$):

$$\mathcal{K}_m = \langle e_n, Ae_n, \dots, A^m e_n \rangle,$$

with e_n the n th vector of the standard basis in $\mathbb{R}^n$. Hence the convergence behavior of these eigenvalues is the same as the convergence behavior of the Lanczos algorithm with initial vector e_n .

As indicated in subsection 5.2, for theoretical purposes one can consider the similarity reduction to semiseparable form as consisting of two parts. First, the orthogonal similarity reduction into tridiagonal form is performed, which requires $O(n^3)$ flops. Then the tridiagonal matrix is transformed into a semiseparable one, requiring $O(n^2)$ flops. More specifically, at the bottom right of the matrix, semiseparable submatrices are formed which are orthogonally similar to the original tridiagonal submatrix at that position. Hence, because we know that the eigenvalues of these tridiagonal submatrices are the Ritz values as described above, we also have that the eigenvalues of the corresponding semiseparable submatrices are equal to these Ritz values. However, besides this property, the part of the method reducing the tridiagonal to a similar semiseparable matrix has an additional advantage, as is shown in the following section.

7. Subspace iteration while reducing the matrix. The intermediate matrices do not only contain the Lanczos–Ritz values in the already semiseparable part. During the construction, also a kind of nested subspace iteration is performed on the matrix which corresponds to executing a QL step on the already semiseparable part. Depending on the gaps and the approximation of the Ritz values, the subspace iteration will transform the semiseparable part into a block-diagonal matrix.

7.1. Theoretical proof. At each step of the algorithm introduced in Theorem 3.1, one more row (column) is added by means of orthogonal transformations to the set of the rows (columns) of the matrix already forming a semiseparable part. In this section, using arguments similar to those considered in [22, 23], we show that this algorithm can be interpreted as a kind of nested subspace iteration method [7], where the size of the vector subspace is increased by one and a change of coordinate system is made at each step of the algorithm. As a consequence, depending on the gap between the eigenvalues, the semiseparable part of the matrix will converge to a block-diagonal matrix, and the eigenvalues of these blocks converge to the largest eigenvalues in absolute value of the original symmetric matrix.

Given a matrix A and an initial subspace $\mathcal{S}^{(0)}$, the subspace iteration method [7] can be described as follows:

$$\mathcal{S}^{(i)} = A\mathcal{S}^{(i-1)}, \quad i = 1, 2, 3, \dots$$

Under weak assumptions on A and $\mathcal{S}^{(0)}$, the $\mathcal{S}^{(i)}$ converge to an invariant subspace. We will see that the reduction algorithm from symmetric to semiseparable matrix can be seen as such a subspace iteration, where the size of the subspace grows by one dimension at each step of the algorithm. Let $A_0^{(0)} = A$. Suppose we have only performed the first orthogonal similarity transformations such that the rows (columns) n and $n - 1$ are already proportional:

$$(7.1) \quad A_0^{(1)} = Q_1^T A_0^{(0)} Q_1,$$

where $A_0^{(1)}$ has the semiseparable structure in the rows (columns) n and $n - 1$ and $Q_1 = [q_1^{(1)}, \dots, q_n^{(1)}]$. From (7.1), we can write

$$(7.2) \quad A_0^{(0)} = Q_1 \begin{pmatrix} \times & \cdots & \times & 0 \\ \vdots & & \vdots & \vdots \\ \times & \cdots & \times & 0 \\ \times & \cdots & \times & \times \end{pmatrix} \equiv Q_1 L_1.$$

Let $e_1, \dots, e_n$ be the standard basis vectors of $\mathbb{R}^n$. From (7.2), because of the structure of L_1 , it can clearly be seen that

$$A_0^{(0)} \langle e_n \rangle = \langle q_n^{(1)} \rangle.$$

This means that the last column of $A_0^{(0)}$ and $q_n^{(1)}$ span the same one-dimensional space. In fact one subspace iteration step is performed on column e_n . The first step of the algorithm is completed when the following orthogonal transformation is performed:

$$A_0^{(1)} = Q_1^T A_0^{(0)} Q_1.$$

The latter transformation can be interpreted as a change of coordinate system: $A_0^{(0)}$ and $A_0^{(1)}$ represent the same linear transformation with respect to different coordinate systems. Let $y \in \mathbb{R}^n$. Then y is represented in the new system by $Q_1^T y$. This means that for the vector $q_n^{(1)}$ we get $Q_1^T q_n^{(1)} = e_n$. Summarizing, this means that one step of subspace iteration on the subspace $\langle e_n \rangle$ is performed, resulting in a new subspace $\langle q_n^{(1)} \rangle$, and then, by means of a coordinate transformation, it is transformed back into the subspace $\langle e_n \rangle$. So, instead of working with a fixed matrix and changing subspaces, we work with fixed subspaces and changing matrices.

The second step can be interpreted in a completely analogous way. Suppose we already have the semiseparable structure in the last two rows (columns). Then, in order to make the rows (columns) n up to $n - 2$ dependent, we perform the following similarity transformation on $A_0^{(1)}$:

$$(7.3) \quad A_0^{(2)} = Q_2^T A_0^{(1)} Q_2.$$

Using (7.3), $A_0^{(1)}$ can be written as follows:

$$A_0^{(1)} = Q_2 \begin{pmatrix} \times & \cdots & \times & 0 & 0 \\ \vdots & & & \vdots & \vdots \\ \times & \cdots & \times & 0 & 0 \\ \times & \cdots & \times & \times & 0 \\ \times & \cdots & \times & \times & \times \end{pmatrix} = Q_2 L_2.$$

Considering the subspace $\langle e_{n-1}, e_n \rangle$ and using the same notation as above, we have

$$A_0^{(1)} \langle e_{n-1}, e_n \rangle = \langle q_{n-1}^{(2)}, q_n^{(2)} \rangle.$$

This means that the second step of the algorithm is a step of subspace iteration performed on a slightly grown subspace. For every new dependency that is created in the symmetric matrix $A_0^{(m)}$, the dimension of the subspace is increased by one.

One can also see the algorithm as performing in each iteration step a QL step without shift on the semiseparable bottom-right submatrix. Let us partition the matrix $A_0^{(m-1)}$ (from (3.2)) as follows:

$$(7.4) \quad A_0^{(m-1)} = \left(\begin{array}{c|c} A_{m-1} & u_{m-1}v_{m-1}^T \\ \hline v_{m-1}u_{m-1}^T & S_{m-1} \end{array} \right),$$

with $A^{(j+1)} = A_{m-1} \in \mathbb{R}^{(n-m) \times (n-m)}$, $m = n - j$, $u_{m-1} \in \mathbb{R}^{(n-m) \times 1}$, $v_{m-1} \in \mathbb{R}^{m \times 1}$, and $S_{m-1} \in \mathbb{R}^{m \times m}$. From the semiseparable structure, we know that $[v_{m-1}, S_{m-1}e_1]$ has rank 1. In the first part of the m th iteration step, an $(n-m) \times (n-m)$ orthogonal matrix $Q_{m,1}$ is constructed such that

$$Q_{m,1}^T u_{m-1} = [0, 0, \dots, 0, \eta]^T,$$

with $\eta = \pm \|u_{m-1}\|$. Hence, our matrix $A_0^{(m-1)}$ is transformed into the following orthogonally similar matrix:

$$(7.5) \quad \begin{aligned} & \left(\begin{array}{c|c} Q_{m,1}^T & 0 \\ \hline 0 & I_m \end{array} \right) \left(\begin{array}{c|c} A_{m-1} & u_{m-1}v_{m-1}^T \\ \hline v_{m-1}u_{m-1}^T & S_{m-1} \end{array} \right) \left(\begin{array}{c|c} Q_{m,1} & 0 \\ \hline 0 & I_m \end{array} \right) \\ &= \left(\begin{array}{c|c} \tilde{A}_{m-1} & \eta e_{n-m}v_{m-1}^T \\ \hline \eta v_{m-1}e_{n-m}^T & S_{m-1} \end{array} \right). \end{aligned}$$

Let us partition the matrix a little differently now, adding one more column and row to the semiseparable part. Let

$$\tilde{A}_{m-1} = \begin{pmatrix} A_m & a \\ a^T & \alpha \end{pmatrix}.$$

Then we can define the semiseparable matrix $\tilde{S}_m$ as follows:

$$\tilde{S}_m = \begin{pmatrix} \alpha & \eta v_{m-1}^T \\ \eta v_{m-1} & S_{m-1} \end{pmatrix}.$$

In the second part of the m th iteration step, we perform a QL step on the semiseparable matrix $\tilde{S}_m$ of order $m+1$; i.e., we construct an orthogonal matrix $Q_{m,2}$ such that

$$\tilde{S}_m = Q_{m,2}L_m.$$

Hence, the matrix of (7.5) is transformed into the orthogonal similar matrix

$$\left(\begin{array}{c|c} I_{n-m-1} & 0 \\ \hline 0 & Q_{m,2}^T \end{array} \right) \left(\begin{array}{c|c} A_m & ae_1^T \\ \hline e_1a^T & \tilde{S}_m \end{array} \right) \left(\begin{array}{c|c} I_{n-m-1} & 0 \\ \hline 0 & Q_{m,2} \end{array} \right) = \left(\begin{array}{c|c} A_m & u_mv_m^T \\ \hline v_mu_m^T & S_m \end{array} \right).$$

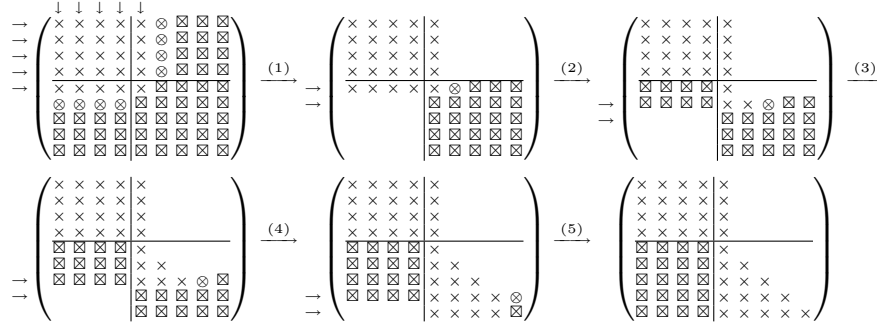


FIG. 7.1. Description of QL step without shift applied at each iteration of the semiseparable reduction.

The execution of the two steps is illustrated in Figure 7.1. At the beginning of the iteration the last four rows and columns already have a semiseparable structure. In step (1), a similarity transformation (either an Householder matrix or a product of Givens transformations) is considered in order to annihilate the entries indicated by $\otimes$. Due to the semiseparable structure, all the entries in the first four rows (columns) with column (row) indexes greater than 5 are annihilated, too. In the second step, consisting of substeps (2), (3), (4), and (5), the Givens rotations are applied only to the left, transforming the (5×5) principal submatrix in the bottom-right corner to lower triangular form. To retrieve the semiseparable structure in the last 5 rows and columns, the transpose of the latter Givens rotation must be applied to the right.

Summarizing, this means that from step m , $m = 0, 1, \dots, n$ (so there is dependency between rows n up to $n - m$), all the consecutive steps perform subspace iterations on the subspace of dimension $m + 1$. From [23], we know that these consecutive iterations on subspaces tend to create block upper triangular matrices. Hence, for a symmetric matrix these are block diagonal. Furthermore, the process works for all the nested subspaces at the same time, and so the semiseparable part of the matrices $A_0^{(m)}$ generated by the proposed algorithm becomes more and more block diagonal, and the blocks contain the eigenvalues having a comparable absolute value. This explains why the lower-right block already gives a good estimate of the largest eigenvalues, since they are connected to a subspace on which the subspace iteration step is already performed several times. Hence as a special case, denoting by $z^{(m)}$ the eigenvector corresponding to the largest eigenvalue in absolute value λ of $A_0^{(m)}$, $m = 0, 1, 2, \dots$, we can say that if $z^{(0)}$ has a nonzero last component and if the largest eigenvalue is unique, the sequence $\{z^{(m)}\}$ converges to e_n , and, consequently, the lower-right element of $A_0^{(m)}$ converges to λ .

This insight also opens several new perspectives. In many problems, only the few largest eigenvalues (in absolute value) need to be computed [1, 8, 20, 21, 24, 25]. In such cases, the proposed algorithm gives the required information after only a few steps, without running the algorithm to completion. Moreover, because the sequence of similar matrices generated at each step of the algorithm converges to a block-diagonal matrix, the original problem can be divided into smaller independent subproblems (deflation). This is illustrated in the first numerical experiment of section 8.

Furthermore, if the second step of the algorithm is “iterated,” the sequence of semiseparable matrices generated converges to a block-diagonal matrix (see the remark at the end of section 3.2).

We finish this section with a theorem from [23] concerning the speed of convergence of subspace iterations [23, Theorem 5.4].

DEFINITION 7.1. Let $\mathcal{S}$ and $\mathcal{T}$ be two nonempty subsets of $\mathbb{C}^n$. The distance $d(\mathcal{S}, \mathcal{T})$ is defined by

$$d(\mathcal{S}, \mathcal{T}) = \sup_{s \in \mathcal{S}, \|s\|_2=1} \inf_{t \in \mathcal{T}} \|s - t\|_2.$$

Using this definition, we can state the following convergence theorem.

THEOREM 7.2. Let $A \in \mathbb{C}^{n \times n}$ be a Hermitian matrix with eigenvalues $\lambda_1, \dots, \lambda_n$ and let p be a polynomial of degree $\leq n$ such that $|p(\lambda_1)| \geq |p(\lambda_2)| \geq \dots \geq |p(\lambda_n)|$ and $|p(\lambda_k)| > |p(\lambda_{k+1})|$ for $k = 1, \dots, n-1$. Let $\mathcal{S}_0$ be a subspace of $\mathbb{C}^n$ of dimension k such that $\mathcal{S}_0 \cap \mathcal{U} = \{0\}$ and $\rho < \hat{\rho} < 1$ where $\rho = |p(\lambda_k)|/|p(\lambda_{k+1})|$ and $\hat{\rho} = \max_{j=1, \dots, k} |p(\lambda_j)|/|p(\lambda_{k+1})|$. Let $\mathcal{S}_i = p_i(A)\mathcal{S}_{i-1}$ for $i = 1, 2, 3, \dots$. Then $d(\mathcal{S}_i, \mathcal{T}) \leq \hat{C}\hat{\rho}^i$ for $i = 1, 2, 3, \dots$ where $\mathcal{T} = \mathcal{U}$ and $\hat{C} = \max_{j=1, \dots, k} |p(\lambda_j)|/|p(\lambda_{k+1})|$.

$$\mathcal{S}_i = p_i(A)\mathcal{S}_{i-1},$$

where $\mathcal{S}_0$ is a subspace of $\mathbb{C}^n$ of dimension k such that $\mathcal{S}_0 \cap \mathcal{U} = \{0\}$ and $\rho < \hat{\rho} < 1$ where $\rho = |p(\lambda_k)|/|p(\lambda_{k+1})|$ and $\hat{\rho} = \max_{j=1, \dots, k} |p(\lambda_j)|/|p(\lambda_{k+1})|$.

$$d(\mathcal{S}_i, \mathcal{T}) \leq \hat{C}\hat{\rho}^i, \quad i = 1, 2, 3, \dots$$

In our case Theorem 7.2 can be applied with the polynomials $p_i(z)$ and $p(z)$ chosen in the following sense: $p_i(z) = p(z) = z$.

7.2. Practical interpretation. Taking a look at the numerical examples, sometimes it seems that initially there is no sign of subspace iteration. The proofs stated above are correct, but there is an interaction with the Lanczos behavior of the algorithm, which will sometimes lead to a delay in the convergence behavior of the subspace iteration. This can happen only when all the vectors $\langle e_{n-i}, e_{n-i+1}, \dots, e_n \rangle$ have a small component in one or more directions of the eigenspace connected to the dominant eigenvalues. Before we show by an example that this happens in practice, we first give a condition under which we are sure that the subspace iteration lets the matrix converge to one having a diagonal block containing the dominant eigenvalues. As soon as some of the Ritz values approximate all of the dominant eigenvalues, this convergence behavior appears. To show this we assume that we can split up the eigenvalues of the initial matrix A into two subsets, $\Lambda_1 = \{\lambda_{1,j} | j \in J\}$ and $\Lambda_2 = \{\lambda_{2,i} | i \in I\}$, with a gap between cluster 2 and cluster 1, $\min_{i \in I} |\lambda_{2,i}| \gg \max_{j \in J} |\lambda_{1,j}|$. Using the notation $\Delta_1 = \text{diag}(\Lambda_1)$ and $\Delta_2 = \text{diag}(\Lambda_2)$, we can write the matrix $A_0^{(m)}$ in the following way:

$$(7.6) \quad \begin{pmatrix} \tilde{A} & R \\ R & S \end{pmatrix} = \begin{pmatrix} V_1 & W_1 \\ V_2 & W_2 \end{pmatrix} \begin{pmatrix} \Delta_1 & 0 \\ 0 & \Delta_2 \end{pmatrix} \begin{pmatrix} V_1^T & V_2^T \\ W_1^T & W_2^T \end{pmatrix},$$

where $\tilde{A} = A^{(j)} = A_m$, $R = u_m v_m^T$, and $S = S_m$ in correspondence with (3.2). From (7.6), we have the following equation for S :

$$S = V_2 \Delta_1 V_2^T + W_2 \Delta_2 W_2^T.$$

Let the eigenvalue decomposition of S be denoted as

$$(7.7) \quad S = \begin{pmatrix} V_S & W_S \end{pmatrix} \begin{pmatrix} \Delta_{S,1} & 0 \\ 0 & \Delta_{S,2} \end{pmatrix} \begin{pmatrix} V_S^T \\ W_S^T \end{pmatrix},$$

where we assume that some of the eigenvalues of S (in fact, the Ritz values) approximate already the real dominant eigenvalues, meaning we assume $|\lambda_{S,2,i}| \approx |\lambda_{2,i}|$, $i \in I$.

If W_2 has full rank and is far from being not of full rank, there is no delay in the convergence behavior of the subspace iteration. This corresponds to the condition that the last vectors $\{e_{n-m}, e_{n-m+1}, \dots, e_n\}$ projected onto the invariant subspace connected to the dominant eigenvalues $\lambda_{2,i}$ have a large component in every direction of this subspace.

Via a Householder similarity transformation we can transform (7.6) into

$$\begin{pmatrix} \bar{A} & \begin{pmatrix} 0 \\ v^T \end{pmatrix} \\ \begin{pmatrix} 0 & v \end{pmatrix} & S \end{pmatrix} = \begin{pmatrix} \bar{V}_1 & \bar{W}_1 \\ V_2 & W_2 \end{pmatrix} \begin{pmatrix} \Delta_1 & 0 \\ 0 & \Delta_2 \end{pmatrix} \begin{pmatrix} \bar{V}_1^T & V_2^T \\ \bar{W}_1^T & W_2^T \end{pmatrix}.$$

Then we can substitute S by its eigenvalue decomposition (7.7).

$$\begin{pmatrix} \bar{A} & \begin{pmatrix} 0 \\ \bar{v}_1^T & \bar{v}_2^T \end{pmatrix} \\ \begin{pmatrix} 0 & \bar{v}_1 \\ & \bar{v}_2 \end{pmatrix} & \begin{pmatrix} \Delta_{S,1} & 0 \\ 0 & \Delta_{S,2} \end{pmatrix} \end{pmatrix} = \begin{pmatrix} \bar{V}_1 & \bar{W}_1 \\ V_2 & W_2 \end{pmatrix} \begin{pmatrix} \Delta_1 & 0 \\ 0 & \Delta_2 \end{pmatrix} \begin{pmatrix} \bar{V}_1^T & \bar{V}_2^T \\ \bar{W}_1^T & \bar{W}_2^T \end{pmatrix}.$$

Note that W_2 is of full rank if and only if $\bar{W}_2$ is of full rank. We get

$$\Delta_{2,S} = \begin{pmatrix} P\bar{V}_2\Delta_1 & P\bar{W}_2\Delta_2 \end{pmatrix} \begin{pmatrix} \bar{V}_2^T P^T \\ \bar{W}_2^T P^T \end{pmatrix},$$

where P is the projection matrix $[0, I]$ of size $n_{S,2} \times n$, where I is the identity matrix of size $n_{S,2} \times n_{S,2}$, with $n_{S,2}$ the dimension of $\Delta_{S,2}$. This means that

$$(7.8) \quad \Delta_{2,S} - P\bar{V}_2\Delta_1\bar{V}_2^T P^T = P\bar{W}_2\Delta_2\bar{W}_2^T P^T.$$

Note that $P\bar{W}_2$ is a square matrix.

Because there is a gap between the eigenvalues $\lambda_{1,j}$, $j \in J$, and $\lambda_{2,i}$, $i \in I$, there will be a comparable gap between $\lambda_{S,2,i}$, $i \in I$, and $\lambda_{1,j}$, $j \in J$. Hence the matrix at the left-hand side of (7.8) is far from singular. Therefore this is also true for the right-hand side, and $\bar{W}_2$ is of full rank.

In the remaining part of this paragraph, a brief explanation is given why, in certain situations, the subspace iteration behavior does not show up immediately from the start. Suppose the size of the block Δ_2 is 2. We can write the matrix A in decomposed form:

$$(7.9) \quad A_0^{(0)} = A = \begin{pmatrix} \bar{A} & a_n \\ a_n^T & \bar{\alpha} \end{pmatrix} = \begin{pmatrix} V & W \\ v^T & w^T \end{pmatrix} \begin{pmatrix} \Delta_1 & 0 \\ 0 & \Delta_2 \end{pmatrix} \begin{pmatrix} V^T & v \\ W^T & w \end{pmatrix}.$$

Applying already the Householder similarity transformation on the matrix A gives us the following decomposition:

$$\begin{aligned} & \begin{pmatrix} \tilde{H} & 0 \\ q^T & 0 \\ 0 & 1 \end{pmatrix} \begin{pmatrix} V & W \\ v^T & w^T \end{pmatrix} \begin{pmatrix} \Delta_1 & 0 \\ 0 & \Delta_2 \end{pmatrix} \begin{pmatrix} V^T & v \\ W^T & w \end{pmatrix} \begin{pmatrix} \tilde{H}^T & q & 0 \\ 0 & 0 & 1 \end{pmatrix} \\ &= \begin{pmatrix} \tilde{A} & a^T & 0 \\ a & \gamma & \beta \\ 0 & \beta & \alpha \end{pmatrix} = \tilde{A}^{(1)}. \end{aligned}$$

The Ritz values are the eigenvalues of the 2×2 matrix:

$$\begin{pmatrix} \gamma & \beta \\ \beta & \alpha \end{pmatrix}.$$

In fact, to complete the previous step, a Givens transformation should still be performed on matrix $\tilde{A}^{(1)}$ to make the last two rows and columns dependent, and thereby transforming the matrix into $A_0^{(1)}$. Note that this last Givens transformation does not change the eigenvalues of the bottom-right submatrix.

We show now that the assumptions for an immediate convergence behavior of the subspace iteration are not satisfied. We prove that

$$[e_{n-1}, e_n]$$

does not have two large linearly independent components in the span of

$$\begin{pmatrix} \tilde{H} & 0 \\ q^T & 0 \\ 0 & 1 \end{pmatrix} \begin{pmatrix} W \\ w^T \end{pmatrix}.$$

We can write $[e_{n-1}, e_n]$ as a linear combination of the eigenvectors:

$$[e_{n-1}, e_n] = \begin{pmatrix} \tilde{H} & 0 \\ q^T & 0 \\ 0 & 1 \end{pmatrix} \begin{pmatrix} V & W \\ v^T & w^T \end{pmatrix} \begin{pmatrix} C_1 \\ C_2 \end{pmatrix}.$$

The coordinates C_2 of $[e_{n-1}, e_n]$ with respect to the dominant eigenvectors are the following:

$$\begin{aligned} C_2 &= \begin{pmatrix} W^T & w \end{pmatrix} \begin{pmatrix} \tilde{H}^T & q & 0 \\ 0 & 0 & 1 \end{pmatrix} [e_{n-1}, e_n] \\ &= \begin{pmatrix} W^T & w \end{pmatrix} \begin{pmatrix} q & 0 \\ 0 & 1 \end{pmatrix} = [W^T q, w]. \end{aligned}$$

Using (7.9), we get that

$$\begin{pmatrix} V^T & v \\ W^T & w \end{pmatrix} \begin{pmatrix} a_n \\ \bar{\alpha} \end{pmatrix} = \begin{pmatrix} \Delta_1 & 0 \\ 0 & \Delta_2 \end{pmatrix} \begin{pmatrix} v \\ w \end{pmatrix}.$$

Hence,

$$(7.10) \quad W^T a_n + w \bar{\alpha} = \Delta_2 w.$$

Because of the Householder transformation we know that $q = a_n / \|a_n\|$, hence

$$(7.11) \quad W^T q = (\Delta_2 - \bar{\alpha} I) \frac{w}{\|a_n\|}.$$

Therefore,

$$\begin{aligned} C_2 &= [W^T q, w] \\ &= [(\Delta_2 - \bar{\alpha} I) \frac{w}{\|a_n\|}, w]. \end{aligned}$$

This means that if $\Delta_2 w$ lies almost in the same direction as w that C_2 is almost singular. If the two largest eigenvalues $\lambda_{2,1} \approx \lambda_{2,2}$, this is the case. Note, however, that when $\lambda_{2,1} \approx -\lambda_{2,2}$, the subspace iteration converges immediately, because they are both extreme and immediately located by the Lanczos procedure.

8. A numerical experiment. In this experiment we want to illustrate the interplay between the Lanczos convergence behavior and the nested subspace iteration of the new method. More numerical experiments can be found in [16].

Let us construct a symmetric matrix having the eigenvalues as shown in Figure 8.1. There are three clusters of eigenvalues where the relative gap between the first and second clusters is equal to 0.5 and the same as the relative gap between the second and third clusters. The first cluster has 20 eigenvalues geometrically distributed between 1 and 10^{-2} , the second cluster 20 eigenvalues between $5 \cdot 10^{-3}$ and $5 \cdot 10^{-5}$, and the third 60 eigenvalues between $2.5 \cdot 10^{-5}$ and $2.5 \cdot 10^{-7}$. The symmetric matrix A is obtained by transforming the diagonal matrix Δ containing the prescribed eigenvalues by an orthogonal similarity transformation $A = Q^T \Delta Q$. The orthogonal matrix used is taken as the Q -factor in the QR factorization of a random matrix, built by the MATLAB command `rand(n)`, where n is the dimension of the matrix.

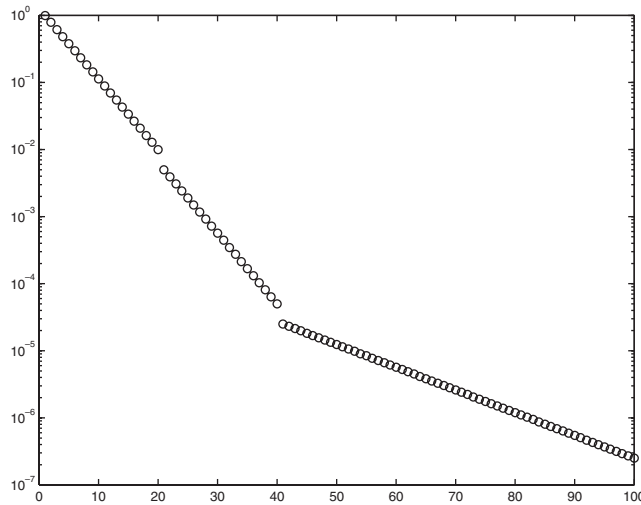


FIG. 8.1. *Eigenvalues of the symmetric matrix of experiment 1: 3 clusters.*

Let us first look at Figure 8.2, which shows the behavior of the Ritz values for this example. In each step of the algorithm (x -axis), a cross is placed if a Ritz value approximates a real eigenvalue up to eight correct digits. Note that the same behavior is also obtained when using the classical tridiagonalization approach (starting at the bottom right of the initial symmetric matrix). However, the tridiagonal matrix obtained using this approach does not give a clear indication of the possible clusters or of possible deflation possibilities; i.e., nowhere on the subdiagonal does the magnitude of the element become very small. The magnitude of the subdiagonal elements is plotted in Figure 8.3. During the execution of the algorithm to obtain a similar semiseparable matrix, there will be a clear point where a diagonal block corresponding to the first cluster can be deflated and another point where a diagonal block corresponding to the second cluster can be deflated. Looking at Figure 8.2, we see that the eigenvalues of the first cluster are approximated well by the Lanczos–Ritz values from step 25 on. This means that at that moment the subspace iteration where the subspace has dimension 20 will converge towards the subspace corresponding to the first cluster. The convergence factor will be 0.5. This can clearly be seen in Figure 8.4. This figure shows for each step $j = 1, 2, \dots, n - 1$ in the algorithm the norms of the subdiagonal

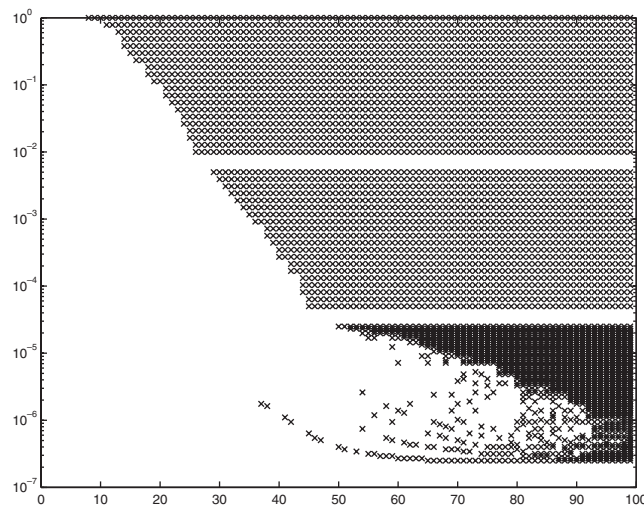


FIG. 8.2. Behavior of the Ritz values.

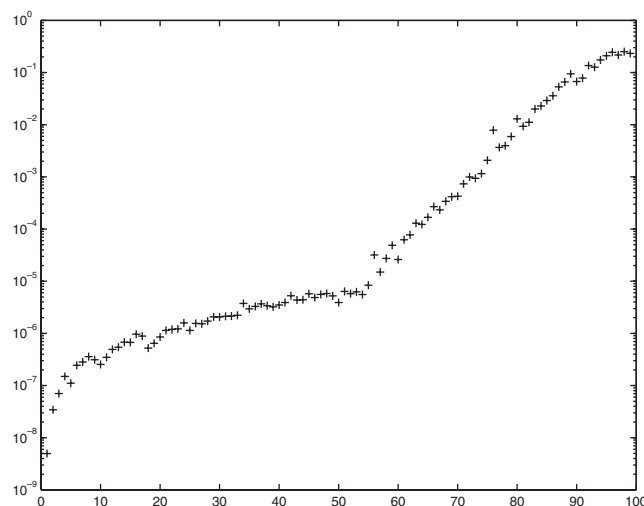


FIG. 8.3. Magnitude of subdiagonal elements of similar tridiagonal matrix.

blocks $S(i : n, 1 : i - 1)$ for $i = n - j : n$. Each line in the plot indicates the change of the norm at each step of the algorithm. The line indicated by “o” corresponds to the norm of the subdiagonal block $S(81 : 100, 1 : 80)$. For the eigenvalues of the second cluster a similar convergence behavior occurs around step 45. The line indicated by “*” corresponds to the norm of the subdiagonal block $S(61 : 100, 1 : 60)$. The parallel lines in the figure having a smaller slope correspond to the subspace convergence behavior inside each of the clusters. Hence, at step 68 a first diagonal 20×20 block can be deflated, while at step 80 the next diagonal 20×20 block can be deflated. This example shows clearly the influence of the convergence of the Lanczos–Ritz values on the convergence behavior of the nested subspace iteration.

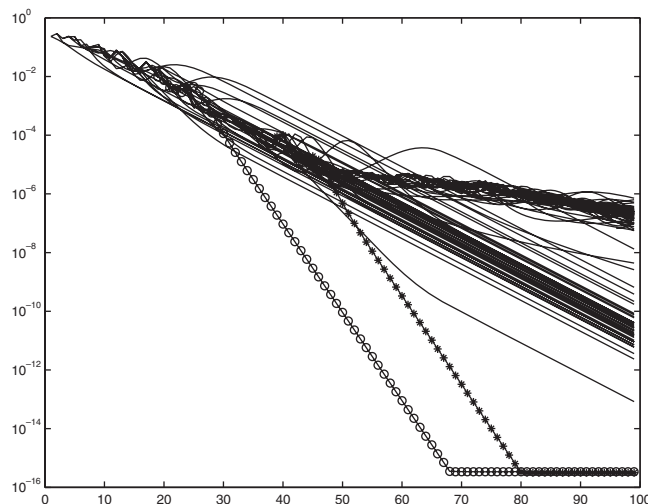


FIG. 8.4. Norms of subdiagonal matrices during the semiseparable reduction.

9. Conclusions. In this paper, we have designed a new algorithm to transform any symmetric matrix into a similar semiseparable one by orthogonal similarity transformations. We have shown that during execution of the algorithm semiseparable matrices are generated whose eigenvalues are nothing else but the Ritz values obtained by the Lanczos process. Moreover, the algorithm can be interpreted as a nested subspace iteration method where a QL step without shift is performed in each iteration step. In [18] an implicit QR algorithm is described for semiseparable matrices. This algorithm can be used to compute the eigenvalues of the resulting matrix as well as of the subblocks which are cut off by the nested subspace iteration.

REFERENCES

- [1] M. BRAUN, S. A. SOFIANOS, D. G. PAPAGEORGIOU, AND I. E. LAGARIS, *An efficient Chebyshev Lanczos method for obtaining eigensolutions of the Schrödinger equation on a grid*, J. Comput. Phys., 126 (1996), pp. 315–327.
- [2] S. CHANDRASEKARAN AND M. GU, *Fast and stable eigendecomposition of symmetric banded plus semi-separable matrices*, Linear Algebra Appl., 313 (2000), pp. 107–114.
- [3] S. CHANDRASEKARAN AND M. GU, *A divide and conquer algorithm for the eigendecomposition of symmetric block-diagonal plus semi-separable matrices*, Numer. Math., 96 (2004), pp. 723–731.
- [4] J. DEMMEL AND K. VESELIĆ, *Jacobi's method is more accurate than QR*, SIAM J. Matrix Anal. Appl., 13 (1992), pp. 1204–1245.
- [5] J. W. DEMMEL, *Applied Numerical Linear Algebra*, SIAM, Philadelphia, 1997.
- [6] D. FASINO, N. MASTRONARDI, AND M. VAN BAREL, *Fast and stable algorithms for reducing diagonal plus semiseparable matrices to tridiagonal and bidiagonal form*, in Fast Algorithms for Structured Matrices: Theory and Applications, V. Olshevsky, ed., Contemp. Math. 323, AMS, Providence, RI, SIAM, Philadelphia, 2003, pp. 105–118.
- [7] G. H. GOLUB AND C. F. VAN LOAN, *Matrix Computations*, 3rd ed., The Johns Hopkins University Press, Baltimore, MD, 1996.
- [8] I. T. JOLLIFFE, *Principal Component Analysis*, Springer-Verlag, New York, 1986.
- [9] R. B. LEHOUCQ AND D. C. SORESENSEN, *Deflation techniques for an implicitly restarted Arnoldi iteration*, SIAM J. Matrix Anal. Appl., 17 (1996), pp. 789–821.
- [10] R. S. MARTIN AND J. H. WILKINSON, *Householder's tridiagonalization of a symmetric matrix*, Numer. Math., 11 (1968), pp. 181–195.
- [11] N. MASTRONARDI, E. VAN CAMP, AND M. VAN BAREL, *Divide and conquer algorithms for com-*

- puting the eigendecomposition of symmetric diagonal-plus-semiseparable matrices, *Numer. Algorithms*, 39 (2005), pp. 379–398.
- [12] B. N. PARLETT AND I. S. DHILLON, *Relatively robust representations of symmetric tridiagonals*, *Linear Algebra Appl.*, 309 (2000), pp. 121–151.
 - [13] G. W. STEWART, *Matrix Algorithms, Vol. II. Eigensystems*, SIAM, Philadelphia, 2001.
 - [14] G. STRANG AND T. NGUYEN, *The Interplay of Ranks of Submatrices*, *SIAM Rev.*, 46 (2004), pp. 637–646.
 - [15] L. N. TREFETHEN AND D. BAU, *Numerical Linear Algebra*, SIAM, Philadelphia, 1997.
 - [16] M. VAN BAREL, R. VANDEBRIL, AND N. MASTRONARDI, *The Lanczos-Ritz Values Appearing in an Orthogonal Similarity Reduction of a Matrix into Semiseparable Form*, Report TW 360, Department of Computer Science, K. U. Leuven, Leuven, Belgium, 2003. Available online from <http://www.cs.kuleuven.ac.be/publicaties/rapporten/tw/TW360.abs.html>.
 - [17] E. VAN CAMP, N. MASTRONARDI, AND M. VAN BAREL, *Two fast algorithms for solving diagonal-plus-semiseparable linear systems*, *J. Comput. Appl. Math.*, 164–165 (2004), pp. 731–747.
 - [18] R. VANDEBRIL, M. VAN BAREL, AND N. MASTRONARDI, *An implicit QR algorithm for symmetric semiseparable matrices*, *Numer. Linear Algebra Appl.*, to appear.
 - [19] R. VANDEBRIL, M. VAN BAREL, AND N. MASTRONARDI, *A note on the representation and definition of semiseparable matrices*, *Numer. Linear Algebra Appl.*, to appear.
 - [20] M. E. WALL, P. A. DYCK, AND T. S. BRETIN, *Svdman—singular value decomposition analysis of microarray data*, *Bioinform.*, 17 (2001), pp. 566–568.
 - [21] M. E. WALL, A. RECHTSTEINER, AND L. M. ROCHA, *Singular value decomposition and principal component analysis*, in *A Practical Approach to Microarray Data Analysis*, D. P. Berrar, W. Dubitzky, and M. Granzow, eds., Kluwer Academic, Dordrecht, The Netherlands, 2003, pp. 91–109.
 - [22] D. S. WATKINS, *QR-like algorithms: An overview of convergence theory and practice*, in *The Mathematics of Numerical Analysis*, *Lectures in Appl. Math.* 32, AMS, Providence, RI, 1996, pp. 879–893.
 - [23] D. S. WATKINS AND L. ELSNER, *Convergence of algorithms of decomposition type for the eigenvalue problem*, *Linear Algebra Appl.*, 143 (1991), pp. 19–47.
 - [24] M. K. S. YEUNG AND W. L. RUZZO, *Principal component analysis for clustering gene expression data*, *Bioinform.*, 17 (2001), pp. 763–774.
 - [25] M. K. S. YEUNG, J. TEGNÉR, AND J. J. COLLINS, EDS., *Reverse engineering gene networks using singular value decomposition and robust regression*, *Proc. Natl. Acad. Sci.*, 99 (2002), pp. 6163–6168.

AN ALTERNATIVE ALGORITHM FOR THE REFINEMENT OF ULV DECOMPOSITIONS*

JESSE L. BARLOW[†], HASAN ERBAY[‡], AND IVAN SLAPNIČAR[§]

Abstract. The ULV decomposition (ULVD) is an important member of a class of rank-revealing two-sided orthogonal decompositions used to approximate the singular value decomposition (SVD). It is useful in applications of the SVD such as principal components where we are interested in approximating a matrix by one of lower rank. It can be updated and downdated much more quickly than an SVD. In many instances, the ULVD must be refined to improve the approximation it gives for the important right singular subspaces or to improve the matrix approximation. Present algorithms to perform this refinement require $O(mn)$ operations if the rank of the matrix is k , where k is very close to 0 or n , but these algorithms require $O(mn^2)$ operations otherwise. Presented here is an alternative refinement algorithm that requires $O(mn)$ operations no matter what the rank is. Our tests show that this new refinement algorithm produces similar improvement in matrix approximation and in the subspaces. We also propose slight improvements on the error bounds on subspaces and singular values computed by the ULVD.

Key words. subspace approximation, orthogonal decompositions, refinement

AMS subject classifications. 65F25, 65F15

DOI. 10.1137/S0895479801372621

1. Introduction. The singular value decomposition (SVD) of a matrix $X \in \mathbf{R}^{m \times n}$, $m \geq n$, is written

$$(1.1) \quad X = Y \Sigma W^T,$$

where $W \in \mathbf{R}^{n \times n}$ is orthogonal and $Y \in \mathbf{R}^{m \times n}$ is, $Y^T Y = I_n$, that is, $Y^T Y = I_n$, and

$$(1.2) \quad \Sigma = \text{diag}(\sigma_1, \dots, \sigma_n) \in \mathbf{R}^{n \times n}, \quad \sigma_1 \geq \dots \geq \sigma_n.$$

It contains useful information about a linear operator, including rank, important subspaces, and conditioning. It also gives us the ability to compute low rank approximations.

Some of the most important applications of the SVD include solving ill-conditioned least squares problems [5], total least squares problems [22], subspace tracking, and isolating principal components [15].

In these applications, our interest in the SVD is to write X in the form

$$(1.3) \quad X = Y_1 \Sigma_1 W_1^T + Y_2 \Sigma_2 W_2^T,$$

*Received by the editors January 31, 2001; accepted for publication (in revised form) by L. Eldén September 28, 2004; published electronically July 13, 2005. This research was supported by the National Science Foundation under contract CCR-9732081.

<http://www.siam.org/journals/simax/27-1/37262.html>

[†]Department of Computer Science and Engineering, Pennsylvania State University, University Park, PA 16802-6106 (barlow@cse.psu.edu).

[‡]Corresponding author. Mathematics Department, Arts and Science Faculty, Kırıkkale University, Kırıkkale 71100, Turkey (hxe68@yahoo.com).

[§]Faculty of Electrical Engineering, Mechanical Engineering and Naval Architecture, University of Split, R. Boškovića b.b., 21000 Split, Croatia (ivan.slapnicar@fesb.hr).

where

$$\Sigma_1 = \text{diag}(\sigma_1, \dots, \sigma_k), \quad \Sigma_2 = \text{diag}(\sigma_{k+1}, \dots, \sigma_n)$$

and Y_1 , Y_2 , W_1 , and W_2 are the corresponding matrices of left and right singular vectors. The other information from the SVD is not needed. Moreover, accurate representations of $\text{Range}(W_1)$ and/or $\text{Range}(W_2)$ are often satisfactory. In principal component analysis, we only need a good low rank approximation of X .

We use $\|\cdot\|$ to denote the two norm (matrix and vector), and $\|\cdot\|_F$ to denote the Frobenius norm. The notation $\sigma_k(X)$ denotes the k th largest singular value of X .

Complete orthogonal decompositions (CODs) can compute an approximation to the dichotomy in (1.3) slightly faster than an SVD. However, if the decomposition must be modified by adding rows (updating) and/or deleting rows (downdating), then the SVD can be computationally expensive [13], whereas CODs may be updated and downdated in $O(mn)$ flops [8, 3, 24, 20, 21]. Our algorithms for construction and refinement lead to a form that can be updated by any of these procedures.

The CODs have the form

$$(1.4) \quad X = UCV^T,$$

where C is a triangular matrix partitioned according to

$$C = \begin{pmatrix} k & n-k \\ C_1 & C_2 \end{pmatrix}, \quad \|C_2\| \approx \|\Sigma_2\| = \sigma_{k+1}$$

and $U \in \mathbf{R}^{m \times n}$ is left orthogonal, $V \in \mathbf{R}^{n \times n}$ is orthogonal.

The first such decompositions were proposed independently in [9] and in [14]. These used maximal column pivoting [6]. More recent enhancements, such as the URV and ULV decompositions, discussed in [20, 21], made use of condition estimators. This paper uses the ULV decomposition (ULVD), which we define in the form (1.4), where, for a fixed integer k and tolerance ϵ , we have

$$(1.5) \quad U = \begin{pmatrix} k & m-k \\ U_1 & U_2 \end{pmatrix}, \quad V = \begin{pmatrix} k & n-k \\ V_1 & V_2 \end{pmatrix},$$

$$(1.6) \quad C = \begin{pmatrix} k & n-k \\ n-k & \end{pmatrix} \begin{pmatrix} L & 0 \\ F & G \end{pmatrix},$$

and

$$(1.7) \quad \|L^{-1}\|^{-1} \leq \epsilon, \quad \|(F \ G)\| \leq \Phi(n)\epsilon.$$

Here $\Phi(n)$ is a modestly growing function of n , say, $\sqrt{n}$. This formulation is a variant of the one in [24, 8, 4]. We refer to k as the ϵ -pseudorank of X , corresponding to a definition used by Lawson and Hanson [16, p. 77].

The matrix V yields approximations to the two right singular subspaces, $\text{Range}(W_1)$ and $\text{Range}(W_2)$, the matrix U yields approximations to the two left singular subspaces, $\text{Range}(Y_1)$ and $\text{Range}(Y_2)$. The accuracy of these approximations and approximations to the singular values of C are discussed in section 2. The results given there are slight enhancements of those by Mathias and Stewart [17] and Fierro and Bunch [10].

In this work, we give improvements to a refinement procedure for the ULVD in [17]. Often after an update or downdate, the conditions (1.7) are not satisfied,

and/or $\|(F \ G)\|_F$ is not small enough. Thus procedures which make $\|(F \ G)\|_F$ smaller are necessary. Procedures are given in [17, 10], but these require $O(mn^2)$ flops (floating point operations; see [12, pp. 18–19]) unless k is very close to 0 or n . Our new procedure (Algorithm 4.2) requires $O(mn)$ flops regardless of the value of k .

Throughout the paper, we use MATLAB notation for submatrices; thus $X(i:j, :)$ denotes rows i through j of X , $X(:, k:\ell)$ denotes columns k through ℓ , and $X(i:j, k:\ell)$ denotes the intersection of the two.

In the next section, we give some improvements of singular value and vector bounds in [17, 10]. In section 3, we introduce the necessary matrix computational tools to build the algorithms for the ULVD. Our new refinement procedure (Algorithm 4.2) is given in section 4 along with some results on its properties. Numerical tests of the algorithm given in section 5 show that the new algorithm is much faster than the one in [17], but just as effective. A conclusion is given in section 6.

2. Subspaces and singular values. The ULVD (1.4)–(1.5) generates approximations to the right and left singular subspaces associated with the first k and last $n - k$ singular values. In the Davis–Kahan [7] framework the errors in these subspaces are characterized by

$$(2.1) \quad |\sin \Theta_R| = \|W_1^T V_2\|_2 = \|V_1^T W_2\|_2,$$

$$(2.2) \quad |\sin \Theta_L| = \|Y_1^T U_2\|_2 = \|U_1^T Y_2\|_2.$$

The following proposition is a slight improvement on bounds by Mathias and Stewart [17] and Fierro and Bunch [10]. It characterizes the error in the singular subspaces and singular values. The long but elementary proof is in the longer technical report version of this paper [2].

PROPOSITION 2.1. . . . X (1.4) (1.5)

$$(2.3) \quad \rho \stackrel{\text{def}}{=} \min \left\{ \frac{\sigma_{k+1}(X)}{\sigma_k(L)}, \frac{\sigma_1(G)}{\sigma_k(X)} \right\},$$

$$(2.4) \quad \gamma_R \stackrel{\text{def}}{=} \max \left\{ \sigma_k^2(X) - \sigma_1^2(G), \sigma_k^2 \left[\begin{pmatrix} L \\ F \end{pmatrix} \right] - \sigma_{k+1}^2(X) \right\},$$

$$(2.5) \quad \gamma_L \stackrel{\text{def}}{=} \max \left\{ \frac{\sigma_k^2(X) - \sigma_1^2(G)}{\sigma_k(X)}, \frac{\sigma_k^2(L) - \sigma_{k+1}^2(X)}{\sigma_k(L)} \right\}.$$

$$(2.6) \quad |\sin \Theta_R| \leq \frac{\|F^T G\|_2}{\gamma_R}, \quad |\sin \Theta_L| \leq \frac{\|F\|_2}{\gamma_L},$$

$$(2.7) \quad |\sin \Theta_R| \leq \rho |\sin \Theta_L|,$$

$$(2.8) \quad |\cos \Theta_L| \leq \frac{\sigma_j(L)}{\sigma_j(X)}, \frac{\sigma_{i+k}(X)}{\sigma_i(G)} \leq 1, \quad \begin{array}{l} j = 1, 2, \dots, k, \\ i = 1, 2, \dots, n - k. \end{array}$$

The bounds in [17] and [10] are the right side of the following two inequalities; thus the bounds in (2.6) are sharper.

$$\begin{aligned} |\sin \Theta_R| &\leq \frac{\|F^T G\|_2}{\gamma_R} \leq \frac{\|F\|_2 \|G\|_2}{\sigma_k^2(L) - \sigma_1^2(G)}, \\ |\sin \Theta_L| &\leq \frac{\|F\|_2}{\gamma_L} \leq \frac{\|F\|_2 \sigma_k(L)}{\sigma_k^2(L) - \sigma_1^2(G)}. \end{aligned}$$

Through the use of trigonometric identities, (2.8) implies the relative error bound

$$(2.9) \quad 0 \leq \frac{\sigma_j(X) - \sigma_j(L)}{\sigma_j(X)}, \frac{\sigma_i(G) - \sigma_{i+k}(X)}{\sigma_{i+k}(X)} \leq \frac{\sin^2 \Theta_L}{1 + |\cos \Theta_L|}$$

for $j = 1, 2, \dots, k$ and $i = 1, 2, \dots, n - k$. Thus as pointed out in [17], the relative error in the singular values is proportional to $\|F\|_2^2$.

3. Matrix computational tools for the ULVD.

3.1. Norm and condition estimates. For a lower triangular matrix $C \in \mathbf{R}^{n \times n}$, we need to estimate its smallest singular value and associated vectors given by the triple $(\sigma_n, \mathbf{u}_n, \mathbf{v}_n)$. Realistically, we need to compute $\bar{\sigma}_n$, $\bar{\mathbf{u}}_n$, and $\bar{\mathbf{v}}_n$ such that

$$(3.1) \quad C^{-1} \bar{\mathbf{u}}_n = \bar{\sigma}_n^{-1} \bar{\mathbf{v}}_n,$$

$$(3.2) \quad C^{-T} \bar{\mathbf{v}}_n = \bar{\sigma}_n^{-1} \bar{\mathbf{u}}_n + \mathbf{r}_{inv},$$

$$(3.3) \quad \mathbf{r}_{inv}^T \bar{\mathbf{u}}_n = 0, \quad \|\mathbf{r}_{inv}\| \leq \bar{\sigma}_n^{-1} \delta,$$

where δ is some prescribed tolerance which should be smaller than ϵ in (1.5). In our tests in section 5, we used $\delta = 10^{-3}\epsilon$. In practice, we expect to use only a few Lanczos or power iterations and some back and forward solves with C to perform this computation.

Likewise, to estimate the largest singular value and associated vectors, given by the triple $(\sigma_1, \mathbf{u}_1, \mathbf{v}_1)$, we compute a triple $(\bar{\sigma}_1, \bar{\mathbf{u}}_1, \bar{\mathbf{v}}_1)$ such that

$$(3.4) \quad C \bar{\mathbf{v}}_1 = \bar{\sigma}_1 \bar{\mathbf{u}}_1,$$

$$(3.5) \quad C^T \bar{\mathbf{u}}_1 = \bar{\sigma}_1 \bar{\mathbf{v}}_1 + \mathbf{r},$$

$$(3.6) \quad \mathbf{r}^T \bar{\mathbf{v}}_1 = 0, \quad \|\mathbf{r}\| \leq \bar{\sigma}_1 \delta.$$

Again, this can be achieved with a few Lanczos or power iterations. Here we do not restrict C to be lower triangular or even square. For complexity estimates, we assume a fixed number of iterations, c_{iter} , for both of these procedures.

When these operations are counted in complexity estimates, we count (3.1)–(3.3) as $c_{iter}n^2$ flops, and for $C \in \mathbf{R}^{m \times n}$ we count (3.4)–(3.6) as $2c_{iter}mn$ flops.

When using these procedures in the analysis of other algorithms, we ignore the errors $\mathbf{r}$ and $\mathbf{r}_{inv}$, though they affect the accuracy.

3.2. Chasing algorithms. Once we have computed the approximate singular vectors in section 3.1, we need the two chasing procedures described in this section. Such algorithms are discussed in [21, 10, 1, 4], so we just summarize them here.

First, suppose that we have computed $(\bar{\sigma}_n, \bar{\mathbf{u}}_n, \bar{\mathbf{v}}_n)$ from (3.1)–(3.3). To “deflate” this smallest singular value, we find an orthogonal matrix $\bar{U}$ such that

$$(3.7) \quad \bar{U}^T \bar{\mathbf{u}}_n = \pm \mathbf{e}_n$$

and an orthogonal matrix $\bar{V}$ such that

$$(3.8) \quad \bar{U}^T C \bar{V} = \hat{C}$$

is lower triangular. The matrices $\bar{U}$ and $\bar{V}$ are each the products of $n - 1$ Givens rotations.

The algorithm (3.7)–(3.8) has interesting consequences for $\hat{C}$. If

$$(3.9) \quad \hat{\mathbf{v}}_n = \bar{V}^T \bar{\mathbf{v}}_n, \quad \hat{\mathbf{r}}_{inv} = U^T \mathbf{r}_{inv},$$

then (3.1)–(3.3) becomes

$$(3.10) \quad \hat{C} \hat{\mathbf{v}}_n = \pm \bar{\sigma}_n \mathbf{e}_n,$$

$$(3.11) \quad \pm \hat{C}^T \mathbf{e}_n = \bar{\sigma}_n \hat{\mathbf{v}}_n + \bar{\sigma}_n \hat{C}^T \hat{\mathbf{r}}_{inv}, \quad \|\hat{\mathbf{r}}_{inv}\| \leq \delta \bar{\sigma}_n^{-1}.$$

It is easily concluded from (3.10) that

$$\begin{aligned} \hat{\mathbf{v}}_n &= \pm \mathbf{e}_n, & \hat{c}_{nn} &= \bar{\sigma}_n, \\ \|\hat{C}^T \mathbf{e}_n\|^2 &= \bar{\sigma}_n^2 + \bar{\sigma}_n^2 \|\hat{C}^T \hat{\mathbf{r}}_{inv}\|^2 \leq \bar{\sigma}_n^2 + \|\hat{C}\|^2 \delta^2 \end{aligned}$$

and that

$$(0 \quad \hat{C}(n, 2 : n)) = \mp \bar{\sigma}_n \hat{\mathbf{r}}_{inv}^T \hat{C}.$$

Thus

$$\|\hat{C}(n, 2 : n)\| \leq \delta \|C\|.$$

Similarly, if C is lower trapezoidal with more columns than rows, we can develop a chasing procedure similar to that above that starts with $\bar{\mathbf{u}}_1$ in (3.4)–(3.6) and produces orthogonal matrices $\bar{U}$ and $\bar{V}$ such that

$$(3.12) \quad \bar{U}^T \bar{\mathbf{u}}_1 = \pm \mathbf{e}_1,$$

and $\bar{V}$ such that $\hat{C}$ in (3.8) remains lower trapezoidal.

If $\hat{\mathbf{v}}_1 = V^T \bar{\mathbf{v}}_1$ and $\hat{\mathbf{r}} = V^T \mathbf{r}$, then

$$\begin{aligned} \hat{C} \hat{\mathbf{v}}_1 &= \pm \bar{\sigma}_1 \mathbf{e}_1, \\ \pm \hat{C}^T \mathbf{e}_1 &= \bar{\sigma}_1 \hat{\mathbf{v}}_1 + \hat{\mathbf{r}}, \quad \|\hat{\mathbf{r}}\| \leq \delta \bar{\sigma}_1. \end{aligned}$$

Since $\hat{\mathbf{r}}^T \hat{\mathbf{v}}_1 = 0$, we have

$$(3.13) \quad \|\hat{C}^T \mathbf{e}_1\|^2 = \bar{\sigma}_1^2 + \|\hat{\mathbf{r}}\|^2,$$

and thus

$$(3.14) \quad \bar{\sigma}_1 \leq \|\hat{C}^T \mathbf{e}_1\| \leq \bar{\sigma}_1 (1 + \delta^2)^{1/2}.$$

4. Refining a ULVD. Refinement is a process of improving the accuracy of the ULVD as an approximation to the SVD. We discuss how to measure that improvement below.

The notion of refinement was popularized in [17], where the following was presented. It is equivalent to the QR algorithm without shifts.

ALGORITHM 4.1 (Mathias–Stewart refinement procedure).

Step 1 $\tilde{U} \in \mathbf{R}^{n \times n}$

$$\begin{aligned} \tilde{U}^T C &= \tilde{U}^T \begin{pmatrix} L & 0 \\ F & G \end{pmatrix} = \begin{pmatrix} L^{(1)} & F^{(1)} \\ 0 & G^{(1)} \end{pmatrix} = C^{(1)}, \\ U &\leftarrow U \tilde{U}. \end{aligned}$$

Step 2 $\rightarrow$ $C^{(1)} \tilde{V} \in \mathbf{R}^{n \times n}$, $\tilde{V} \in \mathbf{R}^{n \times n}$

$$C^{(1)} \tilde{V} = \begin{pmatrix} L^{(1)} & F^{(1)} \\ 0 & G^{(1)} \end{pmatrix} \tilde{V} = \begin{pmatrix} \bar{L} & 0 \\ \bar{F} & \bar{G} \end{pmatrix},$$

$$V \leftarrow V \tilde{V}.$$

We note three important properties of Algorithm 4.1. Other properties were shown in [17].

(1) Since this is the QR algorithm without shifts, it follows that asymptotically

$$\|\bar{F}\| \approx \|F\| \frac{\sigma_{k+1}^2(C)}{\sigma_k^2(C)}.$$

(2) One can show that

$$\|(\bar{F} \quad \bar{G})\| \leq \|G\|, \quad \|(\bar{F} \quad \bar{G})\|_F \leq \|G\|_F.$$

(3) Repeated application of this algorithm forces $\|(\bar{F} \quad \bar{G})\|$ to converge to $\sigma_{k+1}(C)$, but at about the rate of the shiftless QR algorithm.

This algorithm requires $O((m+n)(n-k)k)$ flops, where the constant in $O(\cdot)$ depends upon the implementation. Thus, if $\min\{k, n-k\} = O(1)$, then this is an $O(mn)$ algorithm, but if $\min\{k, n-k\} = O(n)$, this is an $O(mn^2)$ algorithm. In our tests, we used a chasing procedure that combines Steps 1 and 2.

An alternative to Algorithm 4.1, proposed in [10], just computes a ULVD of C , and thereby produce a better ULVD of X . This costs about the same as Algorithm 4.1— $O(mn(n-k))$ flops—making it $O(mn)$ flops if $n-k = O(1)$ and $O(mn^2)$ flops if $n-k = O(n)$.

Below we give an algorithm that computes a refinement in $O(mn)$ flops no matter what the value of k is. For our statement of this refinement procedure, C has the form (1.6).

ALGORITHM 4.2 (alternative refinement algorithm).

Step 1 $\rightarrow$ $U^{(1)} \in \mathbf{R}^{n \times n}$, $V^{(1)} \in \mathbf{R}^{n \times n}$, $G^{(1)} \in \mathbf{R}^{(n-k) \times (n-k)}$, $F^{(1)} \in \mathbf{R}^{n \times (n-k)}$, $\bar{\mathbf{u}}_1 \in \mathbf{R}^{n-k}$, $\bar{\mathbf{v}}_1 \in \mathbf{R}^{n-k}$

$$(F \quad G) \bar{\mathbf{v}}_1 = \bar{\sigma}_1 \bar{\mathbf{u}}_1,$$

$$\bar{\sigma}_1 \bar{\mathbf{u}}_1 \in \text{range}(F \quad G) \quad \bar{\mathbf{v}}_1 \in \text{null}(F \quad G)$$

Step 2 $\rightarrow$ $U^{(1)} \in \mathbf{R}^{n \times n}$, $V^{(1)} \in \mathbf{R}^{n \times n}$, $G^{(1)} \in \mathbf{R}^{(n-k) \times (n-k)}$, $F^{(1)} \in \mathbf{R}^{n \times (n-k)}$, $\bar{\mathbf{u}}_1 \in \mathbf{R}^{n-k}$, $\bar{\mathbf{v}}_1 \in \mathbf{R}^{n-k}$

$$[U^{(1)}]^T \bar{\mathbf{u}}_1 = \pm \mathbf{e}_1,$$

$$G^{(1)} \leftarrow [U^{(1)}]^T G V^{(1)}$$

$$F^{(1)} \leftarrow [U^{(1)}]^T F.$$

$$(F^{(1)} \quad G^{(1)}) = \frac{1}{n-k-1} \begin{pmatrix} k & n-k \\ F^{(1)}(2:n-k,:) & G^{(1)}(2:n-k,:) \end{pmatrix} \begin{pmatrix} [\mathbf{f}_1^{(1)}]^T \\ g_{11}^{(1)} \mathbf{e}_1^T \end{pmatrix},$$

$$\begin{aligned} \left\| \begin{pmatrix} \mathbf{f}_1^{(1)T} & g_{11}^{(1)} \mathbf{e}_1^T \end{pmatrix} \right\| &= \sigma_1 \left[\begin{pmatrix} F & G \end{pmatrix} \right], \\ \left\| \begin{pmatrix} F^{(1)}(2:n-k,:) & G^{(1)}(2:n-k,:) \end{pmatrix} \right\| &= \sigma_2 \left[\begin{pmatrix} F & G \end{pmatrix} \right]. \end{aligned}$$

Step 3

$$S = \begin{pmatrix} L & 0 \\ \mathbf{f}_1^{(1)T} & g_{11}^{(1)} \end{pmatrix}.$$

$$\bar{\mathbf{u}}_{k+1} = \bar{\mathbf{v}}_{k+1}, \quad \bar{\sigma}_{k+1} = \sigma_{k+1}, \quad \bar{S} = S.$$

$$S \bar{\mathbf{v}}_{k+1} = \bar{\sigma}_{k+1} \bar{\mathbf{u}}_{k+1},$$

Step 4

$$U^{(2)}, V^{(2)} \in \mathbf{R}^{(k+1) \times (k+1)} \quad 3.2$$

$$[U^{(2)}]^T \bar{\mathbf{u}}_{k+1} = \pm \mathbf{e}_{k+1},$$

$$S^{(1)} \leftarrow \begin{pmatrix} L^{(1)} & 0 \\ 0 & \bar{\sigma}_{k+1} \end{pmatrix} = [U^{(2)}]^T S V^{(2)},$$

$$L^{(1)} \leftarrow L^{(1)}, \quad \bar{\sigma}_{k+1} \leftarrow \bar{\sigma}_{k+1},$$

$$F^{(2)} \leftarrow \begin{pmatrix} F^{(1)}(2:n-k,:) & G^{(1)}(2:n-k,1) \end{pmatrix} V^{(2)}.$$

$$\bar{\sigma}_{k+1} \geq \epsilon \quad \text{or} \quad \bar{\sigma}_{k+1} < \epsilon, \quad k+1 \leftarrow k+1,$$

$$\bar{L} \leftarrow S^{(1)}, \quad \bar{F} \leftarrow F^{(2)}, \quad \bar{G} \leftarrow G^{(1)}(2:n-k, 2:n-k);$$

$$\bar{\sigma}_{k+1} < \epsilon, \quad k \leftarrow k+1,$$

$$\bar{L} \leftarrow L^{(1)}, \quad \bar{F} \leftarrow \begin{pmatrix} 0 \\ F^{(2)}(:, 1:k) \end{pmatrix},$$

$$\bar{G} \leftarrow \begin{pmatrix} 0 \\ F^{(2)}(:, k+1) & G^{(2)}(2:n-k, 2:n-k) \end{pmatrix}$$

end

$$\bar{U} \leftarrow U \begin{pmatrix} I_k & 0 \\ 0 & U^{(1)} \end{pmatrix} \begin{pmatrix} U^{(2)} & 0 \\ 0 & I_{n-k-1} \end{pmatrix},$$

$$\bar{V} \leftarrow V \begin{pmatrix} I_k & 0 \\ 0 & V^{(1)} \end{pmatrix} \begin{pmatrix} V^{(2)} & 0 \\ 0 & I_{n-k-1} \end{pmatrix}.$$

return $\bar{L}, \bar{F}, \bar{G}, \bar{U}, \bar{V}$

$$X = \bar{U} \begin{pmatrix} \bar{L} & 0 \\ \bar{F} & \bar{G} \end{pmatrix} \bar{V}^T.$$

The refinement algorithm requires $6(m+n)n + 12nk + c_{iter}n^2$ flops. Thus, this is an $O(mn)$ algorithm for all k .

Algorithm 4.2 does not have a property similar to property (1) of Algorithm 4.1. Fortunately, it does have two properties similar to properties (2) and (3), as follows:

1. $\|(\bar{F} \ \bar{G})\|_F \leq \|(F \ G)\|_F$, where equality occurs if and only if $\|(F \ G)\| = \sigma_{k+1}(C)$.

2. Repeated application forces $\|(\bar{F} \ \bar{G})\|$ to converge to $\sigma_{k+1}(C)$. Under reasonable assumptions, a good approximation of $\sigma_{k+1}(C)$ is produced after one iteration.

The following lemma exactly quantifies the decrease in $\|(F \ G)\|_F$.

LEMMA 4.1. *Let $(F \ G) \in \mathbb{R}^{(n-k) \times k}$ and $S \in \mathbb{R}^{k \times k}$ be symmetric positive definite. Then*

$$(4.1) \quad \|(\bar{F} \ \bar{G})\|_F^2 = \|(F \ G)\|_F^2 - \sigma_1^2[(F \ G)] + \sigma_{k+1}^2(S)$$

$$(4.2) \quad = \|(F \ G)\|_F^2 - \left\| \begin{pmatrix} \mathbf{f}_1^{(1)} \\ g_{11}^{(1)} \end{pmatrix} \right\|^2 + \bar{\sigma}_{k+1}^2.$$

where $\bar{\sigma}_{k+1} \geq \epsilon$.

$$(4.3) \quad \|(\bar{F} \ \bar{G})\|_F^2 = \|(F \ G)\|_F^2 - \sigma_1^2[(F \ G)]$$

if

$$\|(\bar{F} \ \bar{G})\| = \sigma_2(F \ G) \leq \|(F \ G)\|.$$

First, due to orthogonal invariance of the Frobenius norm, we observe that

$$(4.4) \quad \|(F^{(1)} \ G^{(1)})\|_F = \|(F \ G)\|_F,$$

where $\|(F^{(1)} \ G^{(1)})^T \mathbf{e}_1\|$ is the largest singular value of the matrix $(F \ G)$. Second, we observe that

$$(4.5) \quad \left\| \begin{pmatrix} \bar{L} & 0 \\ 0 & \bar{\sigma}_{k+1} \end{pmatrix} \right\|_F = \left\| \begin{pmatrix} L & 0 \\ \mathbf{f}_1^{(1)T} & g_{11}^{(1)} \end{pmatrix} \right\|_F,$$

where $\bar{\sigma}_{k+1}$ is the smallest singular value of S . Thus (4.1) follows from combining (4.4)–(4.5). If $\bar{\sigma}_{k+1} > \epsilon$, then the last row of $S^{(1)}$ is not included in $(\bar{F} \ \bar{G})$, and thus (4.3).

Since $\|(\mathbf{f}_1^{(1)T} \ g_{11}^{(1)})^T\| = \sigma_1[(F \ G)]$ and $\mathbf{e}_1$ is the left singular vector associated with this singular value, it follows that

$$\|(\bar{F} \ \bar{G})\| = \|(F^{(1)}(2:n-k, :) \ G^{(2)}(2:n-k, :))\| = \sigma_2[(F \ G)]. \quad \square$$

The computation of (4.2) requires $O(n)$ flops.

The following theorem shows that repeated application of Algorithm 4.2 forces $\|(F \ G)\|$ to converge to $\sigma_{k+1}(C)$.

THEOREM 4.2. *Let $(F_0 \ G_0) \in \mathbb{R}^{(n-k) \times k}$ and $C_0 = \begin{pmatrix} L_0 & 0 \\ F_0 & G_0 \end{pmatrix} \in \mathbb{R}^{n \times n}$ be symmetric positive definite. Then*

$$(4.6) \quad \lim_{t \rightarrow \infty} \|(F_t \ G_t)\| = \sigma_{k+1}(C_0).$$

By using Lemma 4.1 we have

$$(4.7) \quad \|(F_{t+1} \ G_{t+1})\|_F^2 - \|(F_t \ G_t)\|_F^2 = \|(F_t \ G_t)\|^2 - \sigma_{k+1}^2 \left[\begin{pmatrix} L_{t+1} & 0 \\ \mathbf{f}_1^{(t+1)T} & g_{11}^{(t+1)} \end{pmatrix} \right]^2,$$

where $\mathbf{f}_1^{(t+1)T} = F_{t+1}^T \mathbf{e}_1$, $g_{11}^{(t+1)} = \mathbf{e}_1^T G_{t+1} \mathbf{e}_1$.

By the Cauchy interlace theorem [23, pp. 103–104], we have

$$(4.8) \quad \sigma_{k+1} \left[\begin{pmatrix} L_{t+1} & 0 \\ \mathbf{f}_1^{(t+1)T} & g_{11}^{(t+1)} \end{pmatrix} \right] \leq \sigma_{k+1}(C_0) \leq \| (F \ G) \|.$$

Thus from (4.7),

$$(4.9) \quad \|(F_{t+1} \ G_{t+1})\|_F \leq \|(F_t \ G_t)\|_F$$

unless $\|(F_t \ G_t)\| = \sigma_{k+1}(C_0)$. Furthermore, we have

$$(4.10) \quad \sigma_{k+1}(C_0) \leq \|(F_t \ G_t)\|_F \leq \|(F_0 \ G_0)\|_F.$$

Thus by (4.9) and (4.10), $\{\|(F_t \ G_t)\|_F\}$ defines a monotone, bounded sequence which, in turn, has a limit [18, pp. 47–55]. So we have

$$(4.11) \quad \lim_{t \rightarrow \infty} \|(F_t \ G_t)\|_F = \psi$$

for some $\psi \in \mathbf{R}$.

By (4.7), since the sequence $\{\|(F_t \ G_t)\|_F\}$ is a Cauchy sequence, and using (4.8), we have

$$\lim_{t \rightarrow \infty} \|(F_t \ G_t)\| = \lim_{t \rightarrow \infty} \sigma_{k+1} \left[\begin{pmatrix} L_t & 0 \\ \mathbf{f}_1^{(t)T} & g_{11}^{(t)} \end{pmatrix} \right] = \sigma_{k+1}(C_0). \quad \square$$

The next theorem shows that just one step of the refinement procedure can yield a value of $\bar{\sigma}_{k+1}$ that is a very good approximation of the $(k+1)$ st singular value of C .

THEOREM 4.3. *Let C be a symmetric matrix of order n with*

$$(4.12) \quad C = \begin{pmatrix} k & n-k \\ L & 0 \\ F & G \end{pmatrix},$$

where L is a $k \times k$ symmetric matrix, F is a $(n-k) \times k$ matrix, and G is a $(n-k) \times (n-k)$ symmetric matrix.

$$(4.13) \quad \bar{C} = \begin{pmatrix} k & n-k \\ \bar{L} & 0 \\ 0 & \bar{\sigma}_{k+1} \mathbf{e}_1^T \\ \bar{F} & \tilde{G} \end{pmatrix}$$

where $\bar{L}$ is a $k \times k$ symmetric matrix, $\bar{F}$ is a $(n-k) \times k$ matrix, and $\tilde{G}$ is a $(n-k) \times (n-k)$ symmetric matrix.

$$(4.14) \quad 0 \leq \sigma_{k+1}(C) - \bar{\sigma}_{k+1} \leq \frac{\sigma_2^2[(F \ G)]}{\sigma_{k+1}(C) + \bar{\sigma}_{k+1}}.$$

By orthogonal equivalence, $\bar{C}$ and C have the same singular values. First, we note that

$$\bar{C}\bar{C}^T = \begin{pmatrix} k & 1 & n-k-1 \\ \bar{L}\bar{L}^T & 0 & \bar{L}\tilde{F}^T \\ 0 & \bar{\sigma}_{k+1}^2 & \bar{\sigma}_{k+1} \mathbf{e}_1^T \tilde{G}^T \\ \bar{F}\bar{L}^T & \bar{\sigma}_{k+1} \tilde{G} \mathbf{e}_1 & \tilde{F}\tilde{F}^T + \tilde{G}\tilde{G}^T \end{pmatrix}.$$

The Cauchy interlace theorem applied to $\bar{C}\bar{C}^T$ gives us that

$$\bar{\sigma}_{k+1}^2 \leq \sigma_{k+1}^2(\bar{C}) = \sigma_{k+1}^2(C),$$

thereby yielding the upper bound in (4.14).

If we note that

$$\bar{C}^T \bar{C} = \begin{pmatrix} \bar{L}^T \bar{L} & 0 \\ 0 & \bar{\sigma}_{k+1}^2 \mathbf{e}_1 \mathbf{e}_1^T \end{pmatrix} + (\tilde{F} \quad \tilde{G})^T (\tilde{F} \quad \tilde{G}),$$

then Weyl's monotonicity theorem applied to the $(k+1)$ st eigenvalue of $\bar{C}\bar{C}^T$ is

$$\sigma_{k+1}^2(C) \leq \sigma_{k+1}^2 \left[\begin{pmatrix} \bar{L} & 0 \\ 0 & \bar{\sigma}_{k+1} \end{pmatrix} \right] + \sigma_1^2 [(\tilde{F} \quad \tilde{G})].$$

Steps 1–2 of Algorithm 4.2 give us that

$$\sigma_1 [(\tilde{F} \quad \tilde{G})] = \sigma_2 [(F \quad G)].$$

Steps 3–4 give us that

$$\sigma_{k+1} \left[\begin{pmatrix} \bar{L} & 0 \\ 0 & \bar{\sigma}_{k+1} \end{pmatrix} \right] = \bar{\sigma}_{k+1}.$$

Thus

$$\bar{\sigma}_{k+1}^2 \leq \sigma_{k+1}^2(C) \leq \bar{\sigma}_{k+1}^2 + \sigma_2^2[(F \quad G)].$$

Solving for $\sigma_{k+1}(C) - \bar{\sigma}_{k+1}$ yields the upper bound. $\square$

In the next section, we give some numerical tests that confirm the stated properties of these alternative algorithms.

5. Numerical tests. The following tests were done using MATLAB 6.

Let $\omega = 1.1\epsilon$, and for a fixed value of k , let $\zeta_k = \ln \omega/k$. Then let

$$D_k = \text{diag}(e^{\zeta_k}, e^{2\zeta_k}, \dots, e^{k\zeta_k}),$$

noting that $\omega = e^{k\zeta_k}$. For $p = \lfloor 100/k \rfloor$, we let

$$\Sigma_k = \text{diag}(D_k, \xi * D_k, \dots, \xi^p D_k, \xi^{p+1} D_k(1 : 100 - p * k)), \quad \xi = 0.9\epsilon.$$

Using a technique for generating random orthogonal matrices given by Stewart [19], we generated $U, V \in \mathbf{R}^{100 \times 100}$ and let

$$X_k = U \Sigma_k V^T.$$

We then computed the QR decomposition

$$X_k = Q C_k^T$$

and used the lower triangular matrix C_k . This is a matrix with exactly k singular values greater than ϵ , but with the singular values clustered.

For each value of $k = 10, 20, \dots, 90$, we generated 10 such matrices. We used values of $\epsilon = 10^{-\ell}$, $\ell = 2, \dots, 8$. The results were similar for all values of k , so we give the results for $\ell = 3$ only.

We computed and plotted the following:

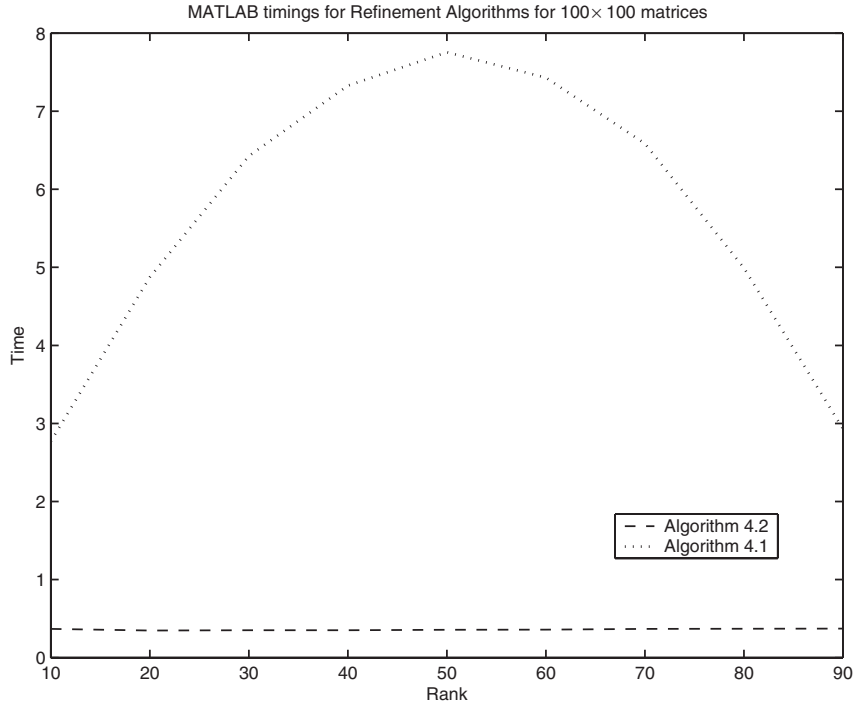


FIG. 5.1. Flop counts of refinement algorithms, $\epsilon = 10^{-3}$.

- MATLAB run times, using the **tic** and **toc** commands (see Figure 5.1).
- Relative error in the k th (in Figure 5.2 (bottom)) and $(k+1)$ st (in Figure 5.2 (top)) singular values. That is, we compute the $\log_{10}$ of

$$\frac{|\sigma_k(C) - \sigma_k(L)|}{\sigma_k(C)}, \quad \frac{|\sigma_{k+1}(C) - \sigma_1(G)|}{\sigma_{k+1}(C)}.$$

- Error in the subspaces. We computed the $\log_{10}$ of

$$|\sin \Theta_R| = \|W_1^T V_2\|_2$$

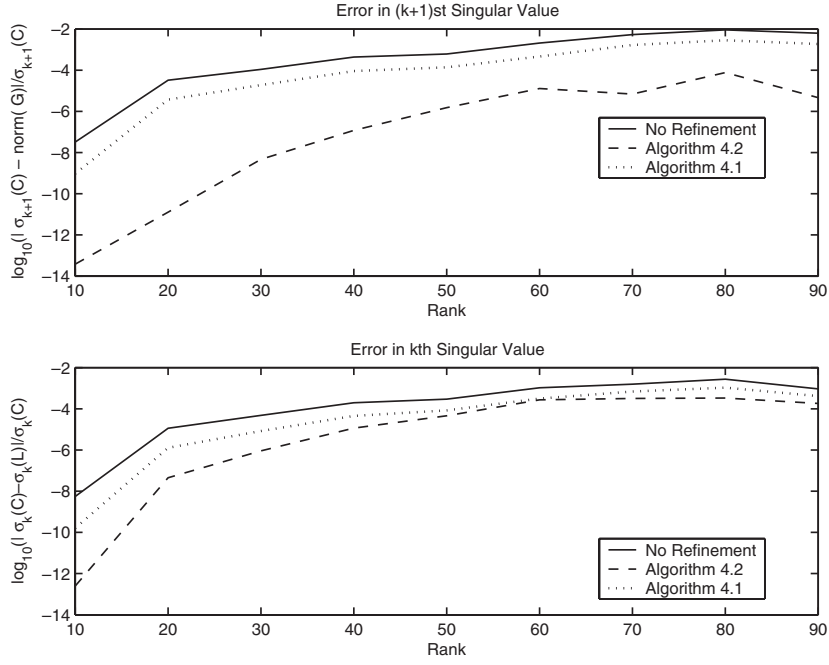
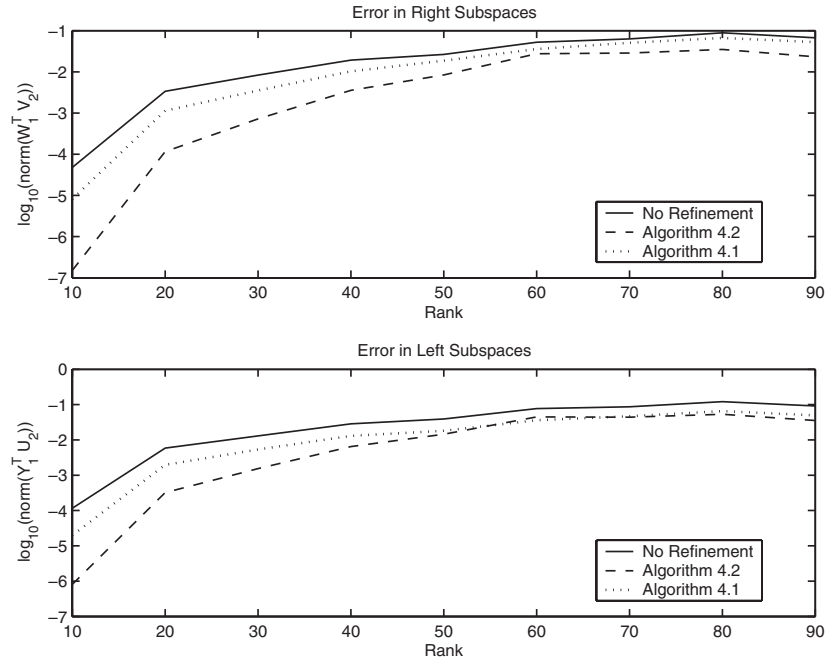
in Figure 5.1 and the $\log_{10}$ of

$$|\sin \Theta_L| = \|Y_1^T U_2\|_2$$

in Figure 5.3.

The unrefined ULVD was computed using the m-file **hulv_a** from the UTV tools package in [11]. Except for rare cases resulting from inaccuracy of the condition estimators, the ϵ -pseudoranks of these matrices were computed correctly by all of the procedures.

We applied each of Algorithm 4.1 and Algorithm 4.2 after a decomposition computed by **hulv_a**. In Figure 5.1, we plotted the flop counts of the two refinement procedures and the subspace errors, including those from computing a ULVD using **hulv_a** with no refinement at all. In Figure 5.2 we graphed the error in the $(k+1)$ st singular value and the rank error in the k th singular value using **hulv_a**. In Figure 5.3, we graphed the error in the right and left subspaces.

FIG. 5.2. Error in $(k + 1)$ st and k th singular values of refinement algorithms, $\epsilon = 10^{-3}$.FIG. 5.3. Error in right and left subspaces of refinement algorithms, $\epsilon = 10^{-3}$.

As one would predict, Algorithm 4.2 is much faster than Algorithm 4.1. Algorithm 4.2 significantly improves the estimate of $\sigma_{k+1}(C)$ given by $\sigma_1(G)$ and slightly improves the estimate of $\sigma_k(C)$ given by $\sigma_k(L)$. It does both better than Algorithm 4.1. It also makes the angles Θ_R and Θ_L with the right and left singular subspaces small and does so as well as or better than Algorithm 4.1.

6. Conclusion. The ULVD has already been shown to be a good substitute for the SVD in circumstances where frequent updates and downdates are done.

We propose a refinement procedure that requires $O(mn)$ flops that is just as effective as the $O(mn^2)$ flops procedures proposed in [17]. Our tests above show that this procedure successfully improves both the singular value and subspace estimates.

REFERENCES

- [1] J. L. BARLOW, *Modification and maintenance of ULV decompositions*, in Applied Mathematics and Scientific Computing, Z. Drmač et al., eds., Kluwer Academic, New York, 2003, pp. 31–63.
- [2] J. L. BARLOW, H. ERBAY, AND I. SLAPNIČAR, *An Alternative Algorithm for Refinement of ULV Decompositions*, Tech. report, Department of Computer Science and Engineering, Pennsylvania State University, University Park, PA, 2004; available online from http://www.cse.psu.edu/~barlow/ULV_Hasan3.ps.
- [3] J. L. BARLOW AND P. A. YOON, *Solving recursive TLS problems using the rank-revealing ULV decomposition*, in Recent Advances in Total Least Squares Techniques and Errors-in-Variables Modeling, S. Van Huffel, ed., SIAM, Philadelphia, PA, 1997, pp. 117–126.
- [4] J. L. BARLOW, P. A. YOON, AND H. ZHA, *An algorithm and a stability theory for downdating the ULV decomposition*, BIT, 36 (1996), pp. 14–40.
- [5] A. BJÖRCK, *Numerical Methods for Least Squares Problems*, SIAM, Philadelphia, PA, 1996.
- [6] P. A. BUSINGER AND G. H. GOLUB, *Linear least squares solutions by Householder transformations*, Numer. Math., 7 (1965), pp. 269–278.
- [7] C. DAVIS AND W. M. KAHAN, *The rotation of eigenvectors by a perturbation*. III, SIAM J. Numer. Anal., 7 (1970), pp. 1–46.
- [8] H. ERBAY, J. L. BARLOW, AND Z. ZHANG, *A modified Gram–Schmidt–based downdating technique for ULV decompositions with applications to recursive TLS problems*, Comput. Statist. Data Anal., 41 (2003), pp. 195–211.
- [9] D. K. FADDEEV, V. N. KUBLANOVSKAYA, AND V. N. FADDEEVA, *Sur les Systèmes Linéaires Algébriques de Matrices Rectangulaires et Mal-Conditionnées*, Vol. VII, Editions Centre Nat. Recherche Sci., Paris, 1968, pp. 161–170.
- [10] R. D. FIERRO AND J. R. BUNCH, *Bounding the subspaces from rank revealing two-sided orthogonal decompositions*, SIAM J. Matrix Anal. Appl., 16 (1995), pp. 743–759.
- [11] R. FIERRO, P. C. HANSEN, AND P. S. KIRK, *UTV tools: Matlab templates for rank-revealing UTV decompositions*, Numer. Algorithms, 20 (1999), pp. 165–194.
- [12] G. H. GOLUB AND C. F. VAN LOAN, *Matrix Computations*, 3rd ed., The Johns Hopkins University Press, Baltimore, MD, 1996.
- [13] M. GU AND S. C. EISENSTAT, *Downdating the singular value decomposition*, SIAM J. Matrix Anal. Appl., 16 (1995), pp. 793–810.
- [14] R. J. HANSON AND C. L. LAWSON, *Extensions and applications of the Householder algorithm for solving linear least squares problems*, Math. Comp., 23 (1969), pp. 787–812.
- [15] I. T. JOLIFFE, *Principal Component Analysis*, Springer-Verlag, New York, 1986.
- [16] C. L. LAWSON AND R. J. HANSON, *Solving Least Squares Problems*, Prentice-Hall, Englewood Cliffs, NJ, 1974.
- [17] R. MATHIAS AND G. W. STEWART, *A block QR algorithm and the singular value decomposition*, Linear Algebra Appl., 182 (1993), pp. 91–100.
- [18] W. RUDIN, *Principals of Mathematical Analysis*, 3rd ed., McGraw-Hill, New York, 1976.
- [19] G. W. STEWART, *The efficient generation of random orthogonal matrices with an application to condition estimators*, SIAM J. Numer. Anal., 17 (1980), pp. 403–409.
- [20] G. W. STEWART, *An updating algorithm for subspace tracking*, IEEE Trans. Signal Process., 40 (1992), pp. 1535–1541.
- [21] G. W. STEWART, *Updating a rank-revealing ULV decomposition*, SIAM J. Matrix Anal. Appl., 14 (1993), pp. 494–499.

- [22] S. VAN HUFFEL AND J. VANDEWALLE, *The Total Least Squares Problem: Computational Aspects and Analysis*, SIAM, Philadelphia, PA, 1991.
- [23] J. H. WILKINSON, *The Algebraic Eigenvalue Problem*, Oxford University Press, London, 1965.
- [24] P. A. YOON AND J. L. BARLOW, *An efficient rank detection procedure for modifying the ULV decomposition*, BIT, 38 (1998), pp. 781–801.

BOUNDS FOR INVERSES OF TRIANGULAR TOEPLITZ MATRICES*

KENNETH S. BERENHAUT[†], DANIEL C. MORTON[†], AND PRESTON T. FLETCHER[†]

Abstract. This short note provides an improvement on a recent result of Vecchio on a norm bound for the inverse of a lower triangular Toeplitz matrix with nonnegative entries. A sharper asymptotic bound is obtained in addition to a version for matrices of finite order. The results are shown to be nearly best possible under the given constraints.

Key words. Toeplitz matrix, inverse matrix, recurrence relation

AMS subject classifications. 15A09, 39A10, 15A57, 15A60

DOI. 10.1137/04060826X

1. Introduction. This paper provides an improvement on a recent result of Vecchio on inverses of lower triangular Toeplitz matrices. We refer the reader to Vecchio [13] for a discussion of applications particularly those to the stability analysis of linear methods for solving Volterra integral equations. Example 1 displays an improvement in that realm. Other references on the topic, mentioned in [13], include [1], [2], [7], [10], [11], and [12].

The matrices of interest here are $(n + 1) \times (n + 1)$ truncations of infinite lower triangular matrices generated by sequences $\{a_i\}_{i \geq 0}$, i.e.,

$$(1.1) \quad A_n = \begin{bmatrix} a_0 & & & & \\ a_1 & a_0 & & & \\ a_2 & a_1 & a_0 & & \\ \cdot & \cdot & \cdot & \cdot & \\ a_n & \cdots & \cdot & a_1 & a_0 \end{bmatrix}.$$

We will prove the following theorem.

THEOREM 1.1. Let $\{a_i\}_{i \geq 0}$ be a sequence of nonnegative real numbers such that

$$(1.2) \quad a_0 \geq a_1 \geq a_2 \geq \cdots a_n \geq a \geq 0$$

for all $n \geq 0$. Then the inverse of A_n exists and

$$(1.3) \quad \|A_n^{-1}\|_\infty \leq \begin{cases} \frac{2}{a} \left(1 - \left(1 - \frac{a}{a_0}\right)^{\lfloor \frac{n}{2} \rfloor + 1}\right) & \text{if } a > 0, \\ \frac{2}{a_0} \left(\left\lfloor \frac{n}{2} \right\rfloor + 1\right) & \text{if } a = 0, \end{cases}$$

where $\lfloor \cdot \rfloor$ denotes the integer part of its argument.

$$(1.4) \quad \|A_n^{-1}\|_\infty \leq \frac{2}{a}$$

if $a > 0$.

*Received by the editors April 13, 2004; accepted for publication (in revised form) by L. Reichel April 14, 2005; published electronically August 31, 2005.

<http://www.siam.org/journals/simax/27-1/60826.html>

[†]Department of Mathematics, Wake Forest University, Winston-Salem, NC 27109 (berenhks@wfu.edu, mortdc0@wfu.edu, fletpt1@wfu.edu). The first author acknowledges financial support from a Sterge Faculty Fellowship.

Note that Vecchio [13] obtained a comparable (to (1.4)) though less sharp result

$$(1.5) \quad \|A_n^{-1}\|_\infty \leq \frac{2}{a} + \frac{1}{a_0},$$

and the methods used therein do not seem amenable to a bound for finite n as in (1.3). Since we are primarily interested in reliable and applicable, explicit bounds for finite n , standard Lyapunov stability methods (cf. [3], [4], [5], [6], [8], [9], and references therein) are also not directly applicable.

(optimality of Theorem 1.1). Suppose $a_1 = a_0 > 0$ and $a_i \equiv a$ for $2 \leq i \leq n$. Also, for $0 \leq i \leq n$, let

$$(1.6) \quad b_i = (-1)^i \frac{1}{a_0} \left(1 - \frac{a}{a_0}\right)^{\left[\frac{i}{2}\right]}.$$

It is easy to verify that in this case

$$(1.7) \quad A_n^{-1} = \begin{bmatrix} b_0 & & & & \\ b_1 & b_0 & & & \\ b_2 & b_1 & b_0 & & \\ \cdot & \cdot & \cdot & \cdot & \cdot \\ b_n & \cdots & \cdot & b_1 & b_0 \end{bmatrix},$$

and hence if $a > 0$,

$$(1.8) \quad \begin{aligned} \|A_n^{-1}\|_\infty &= \sum_{i=0}^n |b_i| \\ &= \sum_{i=0}^n \frac{1}{a_0} \left(1 - \frac{a}{a_0}\right)^{\left[\frac{i}{2}\right]} \\ &= \frac{2}{a} \left(1 - \left(1 - \frac{a}{a_0}\right)^{\left[\frac{n}{2}\right]+1}\right) - \begin{cases} 0 & \text{if } n \text{ is odd,} \\ |b_n| & \text{if } n \text{ is even.} \end{cases} \end{aligned}$$

If $a = 0$, $b_i = \frac{1}{a_0}(-1)^i$ and hence

$$(1.9) \quad \begin{aligned} \|A_n^{-1}\|_\infty &= \sum_{i=0}^n |b_i| \\ &= \frac{n+1}{a_0} \\ &= \frac{2}{a_0} \left(\left[\frac{n}{2}\right] + 1\right) - \begin{cases} 0 & \text{if } n \text{ is odd,} \\ \frac{1}{a_0} & \text{if } n \text{ is even.} \end{cases} \end{aligned}$$

Note that (1.8) and (1.9) compare quite favorably with (1.3), and that for $a > 0$, (1.4) is in fact optimal under the monotonicity assumption in (1.2).

We close this section with an example.

1. In [13], the difference, E_n , between the analytical and numerical solutions due to the application of a direct quadrature (DQ) method to the Volterra integral equation (VIE)

$$(1.10) \quad y(t) = g(t) + \int_0^t k(t-s)y(s) ds, \quad t \in [0, T], y, g, k \in \Re$$

was considered.

In particular, it was shown that if $k(t) \leq 0 \leq k'(t)$, for $t \geq 0$, $\lim_{t \rightarrow \infty} k(t) = \tilde{k} < 0$ and $h|k(0)| < c_1$, then

$$(1.11) \quad \|E_n\|_\infty \leq \left(-\frac{2}{h\tilde{k}} + \frac{1}{c_2}\right) \|\Gamma_n\|_\infty,$$

where $E_n = A_n \Gamma_n$, and c_1 , c_2 , and A_n vary according to the DQ method used (see [13]).

Applying (1.3) in place of (1.4) gives

$$(1.12) \quad \begin{aligned} \|E_n\|_\infty &\leq -\frac{2}{h\tilde{k}} \left(1 - \left(1 + \frac{h\tilde{k}}{c_2}\right)^{\lfloor \frac{n}{2} \rfloor + 1}\right) \|\Gamma_n\|_\infty \\ &\leq -\frac{2}{h\tilde{k}} \|\Gamma_n\|_\infty. \end{aligned}$$

The next section comprises a proof of Theorem 1.1.

2. Proof of Theorem 1.1. As in Vecchio [13], we have that the entries in A_n and A_n^{-1} are related by a linear recurrence with $b_0 = \frac{1}{a_0}$ and

$$(2.1) \quad b_i = \sum_{j=0}^{i-1} -\gamma_{i-1-j} b_j$$

for $1 \leq i \leq n$, where

$$(2.2) \quad \gamma_v = \frac{a_{v+1}}{a_0}.$$

Note that if we set

$$(2.3) \quad h_i = a_0 b_i$$

for $0 \leq i \leq n$, then $h_0 = 1$ and

$$(2.4) \quad h_i = \sum_{j=0}^{i-1} -\gamma_{i-1-j} h_j$$

for $1 \leq i \leq n$, and hence we will restrict attention to the simpler sequence $\{h_i\}$.

Now, set $\gamma_{-1} = 1$ and for $0 \leq i \leq n$, let

$$(2.5) \quad d_i = \gamma_{i-1} - \gamma_i$$

be the successive difference, which is nonnegative under the monotonicity assumption in (1.2).

Also, define the nonnegative sequence $\{S_i\}$ by $S_i = 0$ for $i \leq -2$, $S_{-1} = 1$ and

$$(2.6) \quad S_i = \sum_{j=-1}^{i-1} d_{i-1-j} S_j$$

for $i \geq 0$.

A key to obtaining our bound will be a splitting of h_i into positive and negative parts as given by Lemma 2.1. Note that (2.8) may be compared with the use of the fundamental matrix $\{u_j\}$ in [13] given by

$$(2.7) \quad u_n = 1 - \sum_{l=0}^{n-1} \gamma_{n-1-l} u_l$$

for $n \geq 0$. In fact, it is not difficult to show that $S_i = u_{i+1}$ for all $i \geq -1$. The following lemma follows from (2.6) in [13]. For completeness, we give a proof here.

LEMMA 2.1. . . .

$$(2.8) \quad h_i = S_{i-1} - S_{i-2}$$

for $0 \leq i \leq n$

Note that taking differences in (2.6) gives

$$(2.9) \quad S_i - S_{i-1} = \sum_{j=0}^i d_{i-j} (S_{j-1} - S_{j-2})$$

for $i \geq 1$. In addition, by (2.4)

$$(2.10) \quad h_{i+1} - h_i = \sum_{j=0}^i (\gamma_{i-1-j} - \gamma_{i-j}) h_j - \gamma_0 h_i$$

for $i \geq 1$, and hence

$$(2.11) \quad \begin{aligned} h_{i+1} &= \sum_{j=0}^i (\gamma_{i-1-j} - \gamma_{i-j}) h_j + (1 - \gamma_0) h_i \\ &= \sum_{j=0}^i d_{i-j} h_j. \end{aligned}$$

Since $S_{-1} - S_{-2} = 1 = h_0$ and $S_0 - S_{-1} = d_0 - 1 = h_1$, $\{S_j - S_{j-1}\}_{j \geq -1}$ and $\{h_j\}_{j \geq 0}$ satisfy the same recurrence and the lemma follows. $\square$

Now, let $c_m = \sum_{j=0}^m d_j = 1 - \gamma_m = 1 - a_{m+1}/a_0$. We require the following lemma concerning alternating sums over subsequences of $\{S_i\}$.

LEMMA 2.2. . . . $-2 \leq i_1 < i_2 < \dots < i_k = m$

$$(2.12) \quad \sum_{j=1}^k (-1)^j S_{i_j} \leq \sum_{j=0}^{\lfloor \frac{m+1}{2} \rfloor} c_m^j.$$

If $m \leq 0$, (2.12) is immediate. Suppose (2.12) holds for $m < N$ where $N > 0$. For $m = N$, we have

$$\begin{aligned} \sum_{j=1}^k (-1)^j S_{i_j} &= \sum_{j=1}^k (-1)^j \sum_{v=0}^N d_v S_{i_j-v-1} \\ &= \sum_{v=0}^N d_v \sum_{j=1}^k (-1)^j S_{i_j-v-1} \end{aligned}$$

$$\begin{aligned}
& \leq \sum_{v=0}^N d_v \sum_{j=0}^{\left[\frac{i_k-v}{2}\right]} c_m^j \\
& \leq \sum_{v=0}^N d_v \sum_{j=0}^{\left[\frac{N}{2}\right]} c_m^j \\
& = c_m \sum_{j=0}^{\left[\frac{N}{2}\right]} c_m^j \\
& \leq \sum_{j=0}^{\left[\frac{N+1}{2}\right]} c_m^j,
\end{aligned}
\tag{2.13}$$

where the first inequality in (2.13) follows by induction. $\square$

We are now in a position to prove Theorem 1.1.

Proof of Theorem 1.1. From Lemma 2.1, we have

$$\begin{aligned}
\sum_{j=0}^n |h_j| &= \sum_{j=0}^n |S_{j-1} - S_{j-2}| \\
&= 1 + \sum_{j=0}^{n-1} (-1)^{l_j} (S_j - S_{j-1}) \\
&= 2S_{-1} + \sum_{j=0}^{n-2} Z_j S_j + (-1)^{l_{n-1}} S_{n-1},
\end{aligned}
\tag{2.14}$$

where for $0 \leq j \leq n-1$,

$$l_j = \begin{cases} 0 & \text{if } S_j \geq S_{j-1}, \\ 1 & \text{if } S_j < S_{j-1}, \end{cases}
\tag{2.15}$$

and for $0 \leq j \leq n-2$,

$$Z_j = (-1)^{l_j} - (-1)^{l_{j+1}} \begin{cases} 2 & \text{if } l_j = 0 \text{ and } l_{j+1} = 1, \\ 0 & \text{if } l_j = l_{j+1}, \\ -2 & \text{if } l_j = 1 \text{ and } l_{j+1} = 0. \end{cases}
\tag{2.16}$$

Lemma 2.2 and (2.14) give for some sequence $-2 \leq i_1 < i_2 < \cdots < i_k = n-1$,

$$\begin{aligned}
\sum_{j=0}^n |h_j| &= 2 \left(S_{-1} + \sum_{j=1}^{k-1} (-1)^j S_{i_j} \right) + (-1)^{l_{n-1}} S_{n-1} \\
&\leq 2 \sum_{j=0}^{\left[\frac{n}{2}\right]} c_{n-1}^j.
\end{aligned}
\tag{2.17}$$

Hence from (2.3) and (2.17),

$$\begin{aligned}
 \|A_n^{-1}\|_\infty &= \sum_{j=0}^n |b_j| \\
 &= \sum_{j=0}^n \frac{1}{a_0} |h_j| \\
 &\leq \frac{2}{a_0} \sum_{j=0}^{\lfloor \frac{n}{2} \rfloor} c_{n-1}^j \\
 &= \frac{2}{a_0} \sum_{j=0}^{\lfloor \frac{n}{2} \rfloor} \left(1 - \frac{a_n}{a_0}\right)^j \\
 &\leq \frac{2}{a_0} \sum_{j=0}^{\lfloor \frac{n}{2} \rfloor} \left(1 - \frac{a}{a_0}\right)^j,
 \end{aligned}
 \tag{2.18}$$

and (1.3) and (1.4) follow. $\square$

... Note that if knowledge of a_n is available, then (2.18) provides an improvement on (2.19).

Acknowledgment. We are very thankful to a referee for comments and insights that improved this manuscript.

REFERENCES

- [1] H. BRUNNER, M. R. CRISCI, V. B. E. RUSSO, AND A. VECCHIO, *A family of methods for Abel integral equations of the second kind*, J. Comput. Appl. Math., 34 (1991), pp. 211–219.
- [2] H. BRUNNER AND P. J. VAN DER HOUWEN, *The Numerical Solution of Volterra Equations*, CWI Monogr. 3, North-Holland, Amsterdam, 1986.
- [3] M. R. CRISCI, V. B. KOLMANOVSKII, E. RUSSO, AND A. VECCHIO, *Stability of continuous and discrete Volterra integro-differential equations by Liapunov approach*, J. Integral Equations Appl., 7 (1995), pp. 393–411.
- [4] M. R. CRISCI, V. B. KOLMANOVSKII, E. RUSSO, AND A. VECCHIO, *Boundedness of discrete Volterra equations*, J. Math. Anal. Appl., 211 (1997), pp. 106–130.
- [5] M. R. CRISCI, V. B. KOLMANOVSKII, E. RUSSO, AND A. VECCHIO, *Stability of difference Volterra equations: Direct Liapunov method and numerical procedure*, Comput. Math. Appl., 36 (1998), pp. 77–97.
- [6] J. T. EDWARDS, N. J. FORD, J. A. ROBERTS, AND L. SHAIKHET, *Stability of a discrete nonlinear integro-differential equation of convolution type*, Stab. Control Theory Appl., 3 (2000), pp. 24–37.
- [7] P. HENRICI, *Applied and Computational Complex Analysis*, Vol. 1, John Wiley and Sons, New York, 1974.
- [8] V. B. KOLMANOVSKII AND L. E. SHAIKHET, *General method of Lyapunov functionals construction for stability investigations of stochastic difference equations*, in Dynamical Systems and Applications, World Sci. Ser. Appl. Anal. 4, World Scientific, River Edge, NJ, 1995, pp. 397–439.
- [9] V. B. KOLMANOVSKII AND L. E. SHAIKHET, *Some peculiarities of the general method of Lyapunov functionals construction*, Appl. Math. Lett., 15 (2002), pp. 355–360.
- [10] P. LINZ, *Analytical and Numerical Methods for Volterra Equations*, SIAM, Philadelphia, 1985.
- [11] M. MENEGUETTE, *A note on bordered isoclinical matrices*, J. Comput. Appl. Math., 17 (1987), pp. 375–377.
- [12] A. VECCHIO, *Stability results on some direct quadrature methods for Volterra integro-differential equations*, Dynam. Systems Appl., 7 (1998), pp. 501–518.
- [13] A. VECCHIO, *A bound for the inverse of a lower triangular Toeplitz matrix*, SIAM J. Matrix Anal. Appl., 24 (2003), pp. 1167–1174.

INVERSE PROBLEMS WITH $I \otimes A$ KRONECKER STRUCTURE*

FRED GREENSITE†

Abstract. A problem arising in various scientific settings is the estimation of matrix X in the equation $Y = AX + N$, given matrices A and Y with the noise matrix N . In the standard format of inverse problems (with a single data vector and a single source vector), this takes the form $\text{vec}(Y) = (I \otimes A)\text{vec}(X) + \text{vec}(N)$. This problem has been approached with a variety of methods, none of which appears optimally suited to the common setting where A is ill-conditioned and there are no given constraints on the sequence of columns of X . Consider the space spanned by the zero mean Gaussian random vector columns of X with inner product defined as the expectation of the usual combination of vector components. We derive the minimum-mean-square-error estimate of X under the assumption that the covariance of any two members of the space is invariant under any isometry of the space that leaves the origin fixed. The latter stipulation has the favorable characteristics that it is consistent with the lack of supplied constraints regarding the set of columns of X and allows for imposition of an arbitrary autocovariance matrix for a column of X . A feature of the derived formalism is that the regularization parameter that characterizes Tikhonov or empirical Bayesian treatments of ordinary inverse problems generalizes to a regularization parameter operator. The methodology is capable of effecting a marked reduction in computational complexity as well as increased accuracy over usual methods.

Key words. regularization, multiple regression, inverse problems

AMS subject classifications. 62J05, 15A29, 62J07, 15A52

DOI. 10.1137/S0895479803434197

1. Introduction. Occasionally, inverse problems are encountered in which the defining operator does not address all variables of the unknown function to be estimated. In the discretized linear setting, given an additive noise model, the relevant equation is

$$(1.1) \quad Y = AX + N,$$

where it is required to estimate matrix X , given data matrix Y and matrix A (the discretized operator whose dimensions and rank are unrestricted). Thus, the unknown and the data can be written as matrices, where the variation down a column is with respect to variables that are addressed by the operator, and the variation along a row is with respect to variables that are not addressed by the operator. It is easy to find instances of such problems. For example, inversion of a noise corrupted matrix, some approaches to numerical partial differentiation, and the problem of selection of a regression matrix in multivariate multilinear prediction, all have this form. Since (1.1) is equivalent to the equation sequence

$$(1.2) \quad Y_{:,i} = AX_{:,i} + N_{:,i},$$

$i = 1, 2, 3, \dots$, a large number of examples are also provided by engineering problems potentially subject to a state space treatment using the measurement model (1.2), but where a process model is not immediately evident (subscript “ $:i$ ” indicates the i th column of the subscripted matrix).

*Received by the editors September 2, 2003; accepted for publication (in revised form) by P. C. Hansen February 18, 2005; published electronically August 31, 2005.

<http://www.siam.org/journals/simax/27-1/43419.html>

†Department of Radiological Sciences, University of California–Irvine Medical Center, Orange, CA 92868 (fredg@uci.edu).

The above problems, described by (1.1), are ill-posed if A is poorly conditioned—so that regularization is required. However, these problems have features that contrast sharply with the usual character of an ordinary discretized linear inverse problem of the form

$$(1.3) \quad y = Ax + e,$$

where y , x , and e are vectors. Specifically, (1.1) can be written in that standard form (with a single data vector and a single source vector) as

$$(1.4) \quad \text{vec}(Y) = (I \otimes A)\text{vec}(X) + \text{vec}(N),$$

where “ $\text{vec}(M)$ ” indicates the vector created by sequentially appending the columns of matrix M to each other. However, since the operator in this case is a Kronecker product of a well-conditioned matrix (the identity matrix I), and the poorly conditioned matrix A , one’s intuition might suggest that there is a well-posed component of the problem that can be “factored out” (i.e., the problem is only “partially” ill-posed). Along these lines, an obvious way to proceed is to sequentially treat each of the equations of (1.2) in isolation of each other. This “independent columns method” (ICM) seems dissatisfying because regularization of (1.4) (in a Bayesian sense) is achieved by utilizing the autocovariance of $\text{vec}(X)$ (e.g., the Wiener filter [5]), yet ICM imposes the unjustified assumption that the columns of X are independent of each other. Furthermore, while ICM is by far the most popular way of treating (1.1), it is well known to have the unpleasant consequence of introducing noise-like features into the rows of the solution matrix. Thus, there have been several other methods introduced to counter that unwelcome phenomenon. For the state space approaches, (1.2) is taken to be the measurement model, and a process model is then imposed by some means. Under Gauss–Markov assumptions, the latter takes the form $X_{:i+1} = \Phi_i X_{:i} + u_i$, where Φ_i is a transition matrix and u_i is “process noise.” This results in the Kalman filter or smoother [10]. Since a specific valid process model is not available in the general case, some generic process model must be imposed (e.g., of autoregressive form), perhaps with some adjustable parameters. As an alternative, modifications of the Tikhonov regularization have been proposed. In the “joint regularization” method [2], one sums a “temporal” regularization matrix T with a “spatial” regularization matrix R , each with its own regularization parameter. Thus, one treats the ill-posedness of (1.4) using the regularizing matrix

$$(1.5) \quad \lambda_1 I \otimes R + \lambda_2 T \otimes I,$$

where λ_1 and λ_2 are regularization parameters (the inverse of the expression corresponds to the imposed form of the autocovariance of $\text{vec}(X)$). However, the imposition of a particular T appears to be no better justified in a general treatment of (1.1) than is the imposition of a particular “generic” process model in the state space approach.

The key feature of these problems is that there is an additional structure available for exploitation, not present in an “ordinary” inverse problem (i.e., an inverse problem where the operator lacks the $I \otimes A$ Kronecker structure). The fact that the operator in (1.4) has the form $I \otimes A$ implies that X is a matrix in (1.1)—so that the unknown has both a column space and a row space. Each of these becomes inner product spaces when the inner product is defined as the expectation of the usual combination of vector components. Given that we presumably know how to approach a single member of sequence (1.2) in isolation of all other equations of the sequence, the

difficulty is that there are no given constraints on the covariance between members of the column space. Accordingly, in this paper we provide the minimum-mean-square-error estimate of X in (1.4) under the assumption that covariance of any two members of the column space is invariant under any isometry of the space that leaves the origin fixed. The attraction of the latter condition is that it requires invariance of the implied prior under the latter transformations, which themselves leave invariant the supplied specifications of the problem. The method is thus consistent with the principle that two problems subject to the same prior knowledge should be treated with the same priors.

Notational conventions. Apart from section 4, upper case Roman letters denote matrices, lower case Greek letters denote scalars, and lower case Roman letters denote column vectors, except for i, j, k, m, n, p, q , which denote positive integers. For any matrix M , the i th column is denoted as $M_{:i}$, the i th row as $M_{i:}$, the (i, j) entry as M_{ij} , the Frobenius norm as $\|M\|$, matrix transpose as M' , and the pseudoinverse as $M^\dagger$. $\text{vec}(M)$ is the vector resulting from sequentially appending each column of M to the preceding column. $\mathcal{E}[M]$ denotes the expectation of the random matrix M . An expression of the form $[M(i, j)]_{i,j}$ denotes the block matrix whose (i, j) block is the matrix $M(i, j)$. The values over which i and j range will be indicated in the text as necessary. The symbol $\otimes$ denotes the Kronecker product. The scalar product of matrices having the same dimensions is utilized, i.e., $W \cdot Z \equiv \text{trace}(W'Z)$.

2. Stochastic vector spaces. In what follows, we assume that X and N are independent random matrices with zero mean jointly Gaussian random variable entries, X and N each has n columns, and X has m rows. Then as implied by (1.1), Y also has n columns.

In view of (1.4), the Wiener filter supplies the minimum mean square (MAP) estimate of $\text{vec}(X)$ as

$$(2.1) \quad \text{vec}(X)_{\text{map}} = \{C_{\text{vec}(X)}[I \otimes A]'([I \otimes A]C_{\text{vec}(X)}[I \otimes A]' + C_{\text{vec}(N)})^{-1}\}\text{vec}(Y),$$

where

$$(2.2) \quad \begin{aligned} C_{\text{vec}(X)} &\equiv \mathcal{E}[\text{vec}(X)\text{vec}(X)'], \\ C_{\text{vec}(N)} &\equiv \mathcal{E}[\text{vec}(N)\text{vec}(N)'] \end{aligned}$$

are the autocovariance matrices of $\text{vec}(X)$ and $\text{vec}(N)$, respectively. The term in braces on the right-hand side of (2.1) is the Wiener filter.

Evidently, supplying $C_{\text{vec}(X)}$ requires specification of the covariance of each pair of entries in X . In general, we are given far less—only (perhaps) the presumption that we know how to get a good estimate for one of the constituent ordinary inverse problems in the collection (1.2) in isolation of the available data referable to all the other inverse problems in the collection. The question is, How one should generalize the knowledge of how to approach a single one of the equations of (1.2) (i.e., knowledge of how to approach an “ordinary” inverse problem lacking operator structure $I \otimes A$) to the whole sequence in aggregate?

In treating the above question, we take advantage of the fact that X has both a column space and a row space:

$$\begin{aligned} \text{col}(X) &\equiv \{Xu : u \in \mathbb{R}^n\}, \\ \text{row}(X) &\equiv \{X'z : z \in \mathbb{R}^m\}. \end{aligned}$$

These become inner product spaces upon taking the inner product to be the expectation of the usual combination of components. That is, the inner product on $\text{col}(X)$ is defined by

$$(2.3) \quad \langle v'X' | Xu \rangle \equiv \mathcal{E}[v'X'Xu] = v'\mathcal{E}[X'X]u.$$

With respect to this inner product structure, $\text{col}(X)$ can be identified with the vector space $\mathbb{R}^n$ with the metric $\mathcal{E}[X'X]$. In the same way, $\text{row}(X)$ is identified with $\mathbb{R}^m$ with the metric $\mathcal{E}[XX']$.

However, we take the outer product of two stochastic vectors to be the expectation of their Kronecker product, i.e.,

$$(2.4) \quad \langle \cdot | \mathcal{E}[Xuv'X'] | \cdot \rangle.$$

Note that, in distinction to the usual situation with vector spaces, the outer product of two stochastic vectors is quite different from their tensor product, the latter being given by $\langle \cdot | Xu \rangle \langle v'X' | \cdot \rangle$.

The outer product here performs a special function of linking one stochastic vector space ($\text{col}(X)$) to the other ($\text{row}(X)$), which the tensor product cannot do. As we shall see, under symmetry conditions appropriate to our problem, the outer product in one space has a dual role as an inner product in the other space—a possibility already suggested by the form of (2.4)—because that expression formally makes sense if the entries (to be inserted where the dots are) are either stochastic vectors in $\text{col}(X)$ or nonrandom vectors in $\mathbb{R}^m$ (the inner product space identified with $\text{row}(X)$).

Evidently, the stochastic vector space will be characterized by the statistical properties of X . We have already specified that its entries are zero mean Gaussian random variables. Apart from that specification, the utilized prior should have no bias that would imply a constraint that is actually not given in the antecedent specifications of the problem. Note that the trace of the covariance of two members of $\text{col}(X)$ is the inner product of the two members. Thus, if the prior is invariant under some transformation of $\text{col}(X)$, then the inner product must be invariant under the same transformation—and thus, the transformation must be an isometry that leaves the origin fixed (an orthogonal transformation). On the other hand, we would not wish to impose a priori which changed under an orthogonal transformation, since all directions in the space should be indistinguishable—in the absence of given constraints. This implies that covariance between members of $\text{col}(X)$ (outer product) should be invariant under an orthogonal transformation of $\text{col}(X)$. In physics, one speaks of the isotropy of space with respect to the invariance of a physical property under an orthogonal transformation. This inspires the following in reference to covariance.

DEFINITION 2.1. X is *column-isotropic* if $\mathcal{E}[X'X] = \lambda I_n$ for some $\lambda > 0$. X is *row-isotropic* if $\mathcal{E}[XX'] = \lambda I_m$ for some $\lambda > 0$. X is *isotropic* if it is both column-isotropic and row-isotropic.

Column-isotropy is sufficient to fully provide an appropriate generalization of the Tikhonov regularization to inverse problems with the operator structure $I \otimes A$.

DEFINITION 2.2. X is *column-simple* if $\mathcal{E}[X'X] = I_n$. X is *row-simple* if $\mathcal{E}[XX'] = I_m$. X is *simple* if it is both column-simple and row-simple. X is *unit trace* if $\text{tr}(\mathcal{E}[XX']) = m$ and $\text{tr}(\mathcal{E}[X'X]) = n$.

In other words, X is column-simple if

$$(2.5) \quad \frac{\mathcal{E}[Xuv'X']}{\mathcal{E}[v'X'Xu]}$$

is invariant over all pairs of nonorthogonal stochastic vectors Xu, Xv in $\text{col}(X)$ (and (2.5) is the unit trace outer product on $\text{col}(X)$). Column-simple means that up to a scalar magnitude, there is only one nontrivial outer product, so that all nonzero cross-covariance and autocovariance matrices are proportional to each other. Note that the definition of a simple random matrix has nothing to do with the usual definition of a simple nonrandom matrix.

DEFINITION 2.3. Let X be an $(m \times n)$ random matrix, X separable if there exist T and S $(n \times n)$ and $(m \times m)$ such that

$$(2.6) \quad C_{\text{vec}(X)} = T \otimes S.$$

Separability evidently expresses the notion that the “large” matrix on the left-hand side of (2.6) (defined in (2.2) as the autocovariance matrix of $\text{vec}(X)$) is completely characterized by a single “row prior” and a single “column prior” determined by the matrices on the right-hand side of (2.6).

We will show the following theorems.

THEOREM 2.4. Let X be a random matrix. Then X is column-simple if and only if X is separable.

THEOREM 2.5. Let X be a random matrix. Then the following are equivalent:

1. X is column-simple;
2. X is separable;
3. X is isotropic.

Despite their equivalence, the three statements in Theorem 2.5 play three distinct roles:

- isotropy directly expresses the nonbiased statistical properties of the Gaussian random matrix X ;
- separability is a direct expression of the existence of a single row prior and a single column prior fully characterizing the autocovariance matrix of the vector of matrix entries;
- simplicity allows us to directly identify the row prior from statistics of the available uninverted data.

To begin, we show the following lemma.

LEMMA 2.6. Let X be a random matrix. Then X is column-simple if and only if X is isotropic.

Suppose the random matrix X is column-isotropic. For any two nonzero $Xu, Xv \in \text{col}(X)$, consider the corresponding members of the unit sphere in $\text{col}(X)$, $Xu/\mathcal{E}[u'X'Xu]^{1/2}$, $Xv/\mathcal{E}[v'X'Xv]^{1/2}$. There is clearly a rotation that maps these two latter vectors onto each other (i.e., $\text{col}(X)$ is identified with $\mathbb{R}^n$ with a particular metric). Since autocovariance of members of $\text{col}(X)$ is invariant to any rotation of the space, it follows that

$$(2.7) \quad \frac{\mathcal{E}[Xuu'X']}{\mathcal{E}[u'X'Xu]} = \frac{\mathcal{E}[Xvv'X']}{\mathcal{E}[v'X'Xv]}.$$

This means that all autocovariances in $\text{col}(X)$ are proportional to each other, e.g., proportional to some matrix S . Also, the existence of the above rotation and invariance

of covariance under a rotation imply that

$$\frac{\mathcal{E}[Xuv'X']}{\mathcal{E}[u'X'Xu]^{1/2}\mathcal{E}[v'X'Xv]^{1/2}} = \frac{\mathcal{E}[Xvu'X']}{\mathcal{E}[u'X'Xu]^{1/2}\mathcal{E}[v'X'Xv]^{1/2}},$$

so that all cross-covariance matrices are symmetric. Now, since all autocovariance matrices are proportional to S , we have

$$\mathcal{E}[X(u+v)(u'+v')X'] = \mathcal{E}[Xuu'X'] + \mathcal{E}[Xvv'X'] + \mathcal{E}[Xuv'X'] + \mathcal{E}[Xvu'X'] \propto S.$$

But the first and second terms on the right-hand side of the above equality, being autocovariances, are each proportional to S . Thus, the sum of the third and fourth terms is proportional to S . But these third and fourth terms are identical, because cross-covariances are symmetric. Thus, all cross-covariances between members of $\text{col}(X)$ are proportional to S . Hence, X is column-simple.

Conversely, suppose X is column-simple, so its column autocovariance and cross-covariance matrices are proportional to each other (e.g., proportional to S). Given two members of $\text{col}(X)$, the outer product of the two members they are mapped to under some orthogonal transformation is proportional to the outer product of the original two members. Since the trace of the outer products is preserved under an orthogonal transformation (being equal to their inner product), the outer products preceding and following the transformation must be the same. Thus, X is column-isotropic. $\square$

Now we have the following lemma.

LEMMA 2.7. By definition, $\text{vec}(X)$ can be written as a block vector whose i th block entry is $X_{:i}$. It is thus evident that

$$(2.8) \quad C_{\text{vec}(X)} = \mathcal{E}[\text{vec}(X)(\text{vec}(X))'] = [\mathcal{E}[X_{:i}(X_{:j})']]_{i,j},$$

where the expression on the right-hand side of the second equality is the block matrix with (i, j) entry $\mathcal{E}[X_{:i}(X_{:j})']$ for $i = 1, \dots, n$ and $j = 1, \dots, n$. Suppose X is column-isotropic. Then by Lemma 2.6, X is column-simple, and the matrix entries of the block matrix on the right-hand side above are each proportional to the same matrix S . Let T_{ij} be the proportionality constant between the cross-covariance matrix of the i th and j th columns of X with matrix S . We can then write

$$(2.9) \quad [\mathcal{E}[X_{:i}(X_{:j})']]_{i,j} = [T_{ij}S]_{i,j} = T \otimes S.$$

Thus, X is separable.

On the other hand, if X is separable, (2.8) and (2.9) hold and imply that the column autocovariance and column cross-covariance matrices are proportional to the same matrix S , and so (by Lemma 2.6) X is column-isotropic. $\square$

LEMMA 2.8. $C_{\text{vec}(X)} = T \otimes S$ and $C_{\text{vec}(X')} = S \otimes T$

Note that $C_{\text{vec}(X)} = [\mathcal{E}[X_{:i}(X_{:j})']]_{i,j} = [[\mathcal{E}[X_{pi}X_{qj}]]_{p,q}]_{i,j} = T \otimes S = [[T_{ij}S_{pq}]_{p,q}]_{i,j}$ for $p = 1, \dots, m$ and $q = 1, \dots, m$ with $i = 1, \dots, n$ and $j = 1, \dots, n$. The first equality is the definition of $C_{\text{vec}(X)}$, the second and fourth equalities follow from the definition of the symbols $[\cdot]_{i,j}$, $[\cdot]_{p,q}$, and the third equality is the lemma hypothesis. Similarly, since the preceding equation implies that $T_{ij}S_{pq} = X_{pi}X_{qj}$,

$$\begin{aligned} C_{\text{vec}(X')} &= [\mathcal{E}[(X')_{:p}((X')_{:q})']]_{p,q} = [[\mathcal{E}[X_{pi}X_{qj}]]_{i,j}]_{p,q} = [[T_{ij}S_{pq}]_{i,j}]_{p,q} \\ &= S \otimes T. \quad \square \end{aligned}$$

Theorem 2.4 is now evident.

It is sufficient to show that X is column-isotropic if and only if it is row-isotropic. If X is column-isotropic, then by Lemma 2.7 it is separable with $C_{\text{vec}(X)} = T \otimes S$. By Lemma 2.8, X' is then separable with $C_{X'} = S \otimes T$. By Lemma 2.7 and Definition 2.1, this means that X is row-isotropic, so that column-isotropic implies row-isotropic.

Conversely, if X is row-isotropic, then by definition X' is column-isotropic, so that X' is separable. This means that X itself is separable. By Lemma 2.7, X is then column-isotropic. $\square$

Theorem 2.4 indicates that there is no need to modify “isotropic” with the terms “row” or “column.”

Theorem 2.5 now follows.

Lemma 2.7 and Theorem 2.4 indicate that statements 2 and 3 of Theorem 2.5 are equivalent. Lemma 2.6 and Theorem 2.4 indicate that column-simple is equivalent to isotropy. But isotropy of X implies that X' is column-isotropic, so that X' is column-simple, which implies that X is row-simple. On the other hand, if X is row-simple, then X' is column-simple, so that X' is isotropic, which implies that X is isotropic. Hence, row-simplicity is also equivalent to isotropy. Thus, statements 1 and 3 of Theorem 2.5 are equivalent. $\square$

For an isotropic random matrix X , Theorem 2.5 implies the splitting of the autocovariance matrix of $\text{vec}(X)$ into a portion defining the “row prior” of X and a portion defining the “column prior” of X . That is, in the proof of Lemma 2.7, S is clearly proportional to any column autocovariance or nonzero column cross-covariance matrix of X (as in (2.9)). Similarly, from Lemma 2.8, T is clearly proportional to any column autocovariance or column cross-covariance matrix of X' —i.e., proportional to any row autocovariance or row cross-covariance matrix of X .

Presumably, the column prior S is provided, up to a scalar amplitude, by our knowledge of how to deal with an ordinary inverse problem (e.g., an isolated member of equation sequence (1.2)). Note, however, that the unknown scalar amplitude can be transferred to the matrix T in the decomposition (2.6), so that S can always be taken as having unit trace. Thus, in terms of our program to generalize the treatment of ordinary inverse problems to the case where the operator takes the form $I \otimes A$, the unit trace column prior matrix S can be assumed to have been fully supplied a priori.

The separability of isotropic random matrices actually takes the following pleasing form.

THEOREM 2.9. X is isotropic if and only if

$$(2.10) \quad C_{\text{vec}(X)} = \frac{\mathcal{E}[X'X] \otimes \mathcal{E}[XX']}{\mathcal{E}[\|X\|^2]}.$$

Sufficiency follows since (2.10) implies that X is separable, so by Theorem 2.5 it is isotropic. As for necessity, $\mathcal{E}[XX']$ is the sum of the autocovariance matrices of each of the columns of isotropic X . From Theorem 2.5, isotropy implies simplicity. Thus, the isotropy of X implies that there is a scalar $\gamma(i, j)$ such that

$$(2.11) \quad \mathcal{E}[X_{:i}(X_{:j})'] = \gamma(i, j)\mathcal{E}[XX'].$$

Taking the trace of both sides, it follows that

$$\mathcal{E}[(X_{:i})'X_{:j}] = \gamma(i, j)\mathcal{E}[\|X\|^2].$$

Thus,

$$(2.12) \quad \mathcal{E}[X_{:i}(X_{:j})'] = \frac{\mathcal{E}[(X_{:i})'X_{:j}]}{\mathcal{E}[\|X\|^2]} \mathcal{E}[XX'].$$

Since $C_{\text{vec}(X)}$ is given by (2.8), it follows that (2.12) implies (2.10). $\square$

Equivalently, the following corollary holds.

COROLLARY 2.10. *Let X be an isotropic random matrix.*

1. $C_{\text{vec}(X)} = \text{col}(X) \text{row}(X) + \text{row}(X) \text{col}(X) + \text{vec}(X) \text{vec}(X)'$
2. $\text{row}(X) \text{col}(X) = \text{col}(X) \text{row}(X) = \text{col}(X) \text{col}(X)'$

Conclusion 1 is immediate from (2.10) since $\mathcal{E}[XX']$ is the inner product matrix of $\text{row}(X)$ and $\mathcal{E}[X'X]$ is the inner product matrix of $\text{col}(X)$. Now, $\mathcal{E}[XX']$ (the inner product matrix of $\text{row}(X)$) is the sum of the autocovariance matrices of the columns of X . By simplicity, this means it is proportional to the unit trace outer product of $\text{col}(X)$. Conclusion 2 follows by applying the analogous observation to $\mathcal{E}[X'X]$. $\square$

$\text{row}(X) \text{col}(X)$ -simplicity implies that $R_x \equiv \mathcal{E}[XX']/\mathcal{E}[\|X\|^2]$ is a unit trace matrix proportional to the cross-covariance of any pair of columns of X as well as proportional to the autocovariance of any column of X . Such a matrix is a priori implicitly provided by any Tikhonov treatment of any single member of equation sequence (1.2). The importance of $\text{row}(X) \text{col}(X)$ -simplicity is that it implies that $\mathcal{E}[X'X]$ (according to (2.10), the rest of $C_{\text{vec}(X)}$) can be obtained from $\mathcal{E}[X'(A'A)X] = \mathcal{E}[(AX)'(AX)]$, which relates directly to the measurement matrix Y rather than to the “source” matrix X .

COROLLARY 2.11. *Let X be an isotropic random matrix.*

$$(2.13) \quad C_{\text{vec}(X)} = \frac{\mathcal{E}[X'X] \otimes \mathcal{E}[XX']}{I \cdot \mathcal{E}[XX']} = \frac{\mathcal{E}[X'(A'A)X] \otimes \mathcal{E}[XX']}{(A'A) \cdot \mathcal{E}[XX']}.$$

Since $I \cdot \mathcal{E}[XX'] = \mathcal{E}[\|X\|^2]$, so the first equality above follows from Theorem 2.9. Since X is isotropic, it is simple and the invariance of expression (2.5) pertains. Now,

$$\mathcal{E}[X'(A'A)X] = \sum_i \mathcal{E}[X'(A_{i:})'A_{i:}X],$$

i.e., the left-hand side above is the sum of the autocovariance matrices of the rows of AX . Row-simplicity implies that each term in the sum on the right-hand side above is proportional to $\mathcal{E}[X'X]$ (the latter being the sum of the row autocovariance matrices of X). Thus,

$$(2.14) \quad \frac{\mathcal{E}[X'X]}{I \cdot \mathcal{E}[XX']} = \frac{\mathcal{E}[X'(A'A)X]}{\text{trace}(\mathcal{E}[X'(A'A)X])}.$$

(That is, the numerators on both sides of the equation are proportional, so that division by their respective traces must yield the same unit trace matrix.) Now, write the product singular value decomposition of (A, X') [4] as

$$\begin{aligned} A &= U_a S_a Z', \\ X' &= U_x S_x Z^{-1} \end{aligned}$$

for S_a, S_x diagonal and U_a, U_x orthogonal. Simple substitution shows that

$$\begin{aligned}
 (A'A) \cdot \mathcal{E}[XX'] &= \text{trace}(\mathcal{E}[ZS'_a U'_a U_a S_a Z' (Z^{-1})' S'_x U'_x U_x S_x Z^{-1}]) \\
 &= \text{trace}(\mathcal{E}[ZS'_a S_a S'_x S_x Z^{-1}]) \\
 (2.15) \quad &= \text{trace}(\mathcal{E}[S'_a S_a S'_x S_x]).
 \end{aligned}$$

But

$$\begin{aligned}
 \text{trace}(\mathcal{E}[X'(A'A)X]) &= \text{trace}(\mathcal{E}[U_x S_x Z^{-1} ZS'_a U'_a U_a S_a Z' (Z^{-1})' S'_x U'_x]) \\
 (2.16) \quad &= \text{trace}(\mathcal{E}[S_x S'_a S_a S'_x]),
 \end{aligned}$$

again using the feature that a trace is invariant to a similarity transformation. Straightforward multiplication demonstrates that the right-hand sides of (2.15) and (2.16) are equivalent. Thus,

$$(2.17) \quad \text{trace}(\mathcal{E}[X'(A'A)X]) = (A'A) \cdot \mathcal{E}[XX'].$$

(2.14) and (2.17) imply the second equality in (2.13). $\square$

The preceding corollary allows one to express $C_{\text{vec}(X)}$ in terms of the “measurement” and noise statistics.

COROLLARY 2.12. *Let $Y = AX + N$ where X is isotropic and N is white noise. Then the covariance matrix of the vectorized data $\text{vec}(X)$ is given by*

$$(2.18) \quad C_{\text{vec}(X)} = \mathcal{E}[X'X] \otimes R_x = \frac{(\mathcal{E}[Y'Y] - \mathcal{E}[N'N]) \otimes R_x}{(A'A) \cdot R_x}.$$

Since X is isotropic, and $\mathcal{E}[XX']$ is the sum of the autocovariance matrices of the columns of X , it follows that $R_x = \mathcal{E}[XX']/\text{trace}(\mathcal{E}[XX'])$ is the unit trace matrix proportional to any column autocovariance matrix. The first equality in (2.18) then follows from the first equality in (2.13) and the identity, $\text{trace}(\mathcal{E}[XX']) = I \cdot \mathcal{E}[XX']$. Dividing the numerator and denominator on the right-hand side of the second equality in (2.13) by the $\text{trace}(\mathcal{E}[XX'])$, we obtain

$$(2.19) \quad C_{\text{vec}(X)} = \mathcal{E}[X'X] \otimes R_x = \frac{\mathcal{E}[(AX)'AX] \otimes R_x}{(A'A) \cdot R_x}.$$

The second equality in (2.18) then follows from substituting $(Y - N)$ for AX on the right-hand side of the second equality above, and noting that $\mathcal{E}[N'(Y - N)]$ and $\mathcal{E}[(Y - N)'N]$ are zero matrices since the zero mean entries of N are independent of the zero mean entries of X . $\square$

Note that the use of any Tikhonov regularizing matrix for treatment of some member of equation sequence (1.2) implies the assumption that the autocovariance of the corresponding column of X is proportional to the inverse of that regularizing matrix. Thus, R_x^{-1} can as well be assumed to be a Tikhonov regularizing matrix for one of the equations of (1.2) (assuming the corresponding column of N is white). In the same way, zero mean isotropic N implies

$$(2.20) \quad C_{\text{vec}(N)} = \mathcal{E}[N'N] \otimes R_e,$$

where R_e is the unit trace matrix supplied by some regularization treatment of an equation of sequence (1.2) (e.g., proportional to the identity matrix for the case of the Tikhonov regularization). Thus, given that X and N are isotropic and independent, the requisite ingredients for estimation of X via (2.1) are supplied by $\mathcal{E}[Y'Y]$, $\mathcal{E}[N'N]$, and the regularization matrices R_x and R_e appropriate to ordinary inverse problems.

In practical terms, it is important to note that $Y'Y$, obtained from the data realization Y , is a “close” estimate of $\mathcal{E}[Y'Y]$. Indeed, the only variation in Y given repeated identical experimental setups is due to the noise component, so that

$$(2.21) \quad \mathcal{E}[\|Y'Y - \mathcal{E}[Y'Y]\|^2] = O(\epsilon^2),$$

where ϵ^2 is the largest variance of an entry of N . Thus, in low noise settings, the noisy data $Y'Y$ is actually a good approximation of $\mathcal{E}[Y'Y]$. A nontrivial estimate of $\mathcal{E}[N'N]$ may also be available (in white noise settings, $\mathcal{E}[N'N] \propto I$).

In summary, column-isotropy describes a presumptive symmetry of the space spanned by a set of random vector columns of X . This condition specifies a generalization of the Tikhonov treatment to inverse problems with the $I \otimes A$ Kronecker structure as follows.

1. Column-isotropy implies that the relevant random matrices are simple, separable, and isotropic (Theorems 2.4 and 2.5).
2. Separability allows us to decompose the large autocovariance matrix ($C_{\text{vec}(X)}$ or $C_{\text{vec}(N)}$) into a Kronecker product of a “row prior” and a “column prior” (Theorem 2.9).
3. Column-simplicity allows us to use our knowledge of a regularizing operator for one of the ordinary inverse problems in collection (1.2) (ideally, proportional to the inverse of the autocovariance matrix of a column—from the Bayesian standpoint) to supply the column prior (Definition 2.2).
4. Row-simplicity allows us to use uninverted data to estimate the row prior (Corollary 2.12 and (2.21)).
5. These procedures then fully provide the requisites for estimation of X in an unambiguous fashion via (2.1).

In particular, we have established Theorem 2.5 and the meaning of the paragraph immediately following its statement.

3. Algorithmic considerations.

3.1. General algorithm. In the last section, it was shown that an isotropic random matrix X is simple and separable. Thus, for the inverse problem expressed by $Y = AX + N$, with isotropic independent zero mean X and N , we have $C_{\text{vec}(X)}$ of the form (2.18) and $C_{\text{vec}(N)}$ of the form (2.20). Given R_x and R_e (supplied by knowledge of how to treat ordinary inverse problems), the choice of $\mathcal{E}[Y'Y]$ (taken from the measured data as $Y'Y$), and the choice of $\mathcal{E}[N'N]$, one uses (2.1) to obtain an estimate for X . Nominally, the resulting calculation has high computational complexity due to the large matrix multiplications and the inverse on the right-hand side of that expression.

As an aside, we observe that the above program could also be applied to the nonlinear setting. If X is isotropic and a is any m -dimensional linear functional, then Theorem 2.5 implies that the autocovariance matrix of $(a[X_{:1}], \dots, a[X_{:n}])$ is proportional to $\mathcal{E}[X'X]$. If we assume that this remains true for appropriate nonlinear functionals as well (e.g., continuous nonconstant functions on the domain $\mathbb{R}^m$), then it can be shown that Corollary 2.12 continues to hold if the $(p \times m)$ matrix A is

replaced by nonlinear $\mathbf{A}: \mathbb{R}^m \rightarrow \mathbb{R}^p$ with $\mathbf{A}[x] = (a_1(x), \dots, a_p(x))'$, where a_i are functions of the appropriate class. That is, defining $\mathbf{A}[X] \equiv (\mathbf{A}[X_{:1}], \dots, \mathbf{A}[X_{:n}])$, it follows that $\mathcal{E}[(\mathbf{A}[X])'(\mathbf{A}[X])]$ (which is equal to $\mathcal{E}[Y'Y] - \mathcal{E}[N'N]$) is the sum of the row autocovariance matrices of $\mathbf{A}[X]$ —and hence proportional to the desired $\mathcal{E}[X'X]$ —since by hypothesis, each row of $\mathbf{A}[X]$ has an autocovariance proportional to the autocovariance of any row of X . Thus, for the nonlinear case, one can potentially provide the autocovariance of $\text{vec}(X)$ using data $Y'Y$, Theorem 2.9, and a single regularization parameter expressing the above unknown proportionality constant. We do not pursue this further here.

3.2. Complexity reduction. Suppose the sequence of noise columns is white, i.e.,

$$(3.1) \quad \mathcal{E}[N'N] = \epsilon^2 I.$$

Let Q be the eigenvector matrix of $\mathcal{E}[Y'Y]$. Then from (2.18) and (2.20), it is seen that C_{XQ} and C_{NQ} will both be block diagonal. Since their off-diagonal terms are the column cross-covariance matrices (and hence zero matrices due to the block diagonality), it follows that the following is a sequence of independent equations:

$$(3.2) \quad YQ_{:i} = A(XQ)_{:i} + NQ_{:i},$$

$i = 1, \dots, n$. In this case, a $\dots$ estimate, $(XQ_{:k})_{\text{map}}$, is obtained from isolated treatment of the k th equation of sequence (3.2)—which is then the k th column of the $\dots$ estimate $(XQ)_{\text{map}}$. Note that choices of R_x and R_e provide the regularization matrices relevant to each member of equation sequence (3.2). Importantly, since Q is orthogonal, it follows that

$$(3.3) \quad X_{\text{map}} = (XQ)_{\text{map}} Q'.$$

Note that if $\mathcal{E}[N'N]$ is not proportional to I , the requisite Q providing block diagonals C_{XQ} and C_{NQ} would be nonorthogonal. That is, to provide block diagonals C_{XQ} and C_{NQ} , Q would have to simultaneously diagonalize $\mathcal{E}[Y'Y]$ and $\mathcal{E}[N'N]$ (i.e., solve a generalized eigenvalue problem). Such a matrix cannot be chosen to be orthogonal in general (that is, unless $\mathcal{E}[N'N] \propto I$). In this case,

$$X_{\text{reg}} \equiv (XQ)_{\text{map}} Q^{-1} \neq X_{\text{map}}.$$

Indeed, X_{reg} can considerably amplify any error in the estimate $(XQ)_{\text{map}}$. For example, suppose $(XQ)_{\text{map}} - XQ = W$ with $\mathcal{E}[W]$ the zero matrix and $\mathcal{E}[\|W\|^2] = \nu^2$. Then $\mathcal{E}[\|(XQ)_{\text{map}} Q^{-1} - X\|^2] = \mathcal{E}[\|WQ^{-1}\|^2] = \nu^2 \|Q^{-1}\|^2$. To avoid this error amplification, one is stuck with the high computational complexity calculation resulting from direct substitution of (2.18) and (2.20) into the right-hand side of (2.1)—unless (3.1) holds.

To place this in context, $Y = AX + N$ is equivalent to the equation sequence (1.2). The standard method (ICM) posits that each of the equations of that sequence is independent. In the block matrix format of (2.8), this would imply that $C_{\text{vec}(X)}$ is block diagonal (and similarly for $C_{\text{vec}(N)}$). That is, based on (2.18) and (2.20), the standard method imposes the assumptions that $\mathcal{E}[Y'Y]$ and $\mathcal{E}[N'N]$ are each diagonal. This is a radical assumption, and in practical settings far from accurate (as we have pointed out, in low noise situations $\mathcal{E}[Y'Y]$ is essentially the same as $Y'Y$).

For example, it is a practical observation in dynamic medical imaging that $\mathcal{E}[Y'Y]$ is never close to being diagonal (there are few abrupt changes in the motion of a beating heart, or breathing lungs, throughout the time series of images). This is simply an observation that a row of the data matrix is a discrete approximation of a piecewise continuous function. The most frequently used method, positing the independence of the constituent univariate problems (i.e., diagonal $\mathcal{E}[Y'Y]$) is thereby typically based on an inherently inaccurate assumption. The time series solution estimates resulting from that flawed assumption have obvious excess noise; e.g., see section 4. In our approach, we do not impose that assumption. As we have seen that inaccurate “default” assumption can be effectively replaced by an alternative assumption (isotropy) much more consistent with reality.

Nominally, computations involving (2.1) with choices for $\mathcal{E}[Y'Y]$ and $\mathcal{E}[N'N]$ that are not diagonal would impose much greater computational complexity than occurs with ICM. However, if $\mathcal{E}[N'N] \propto I$ (often a fairly accurate assumption), our approach does reduce to a set of low complexity computations (i.e., treatment of the individual equations of (3.2) in isolation of each other—comparable to the independent treatment of each equation of (1.2)). Of course, one is also required to compute the eigenvectors of $\mathcal{E}[Y'Y]$, but in practice the dimension of this matrix can be controlled by limiting the number of time points (columns). More importantly, the complexity reduction can actually be expected to be radically reduced compared to the independent columns method, since the eigenvalues of $\mathcal{E}[X'X]$ typically tend to zero rapidly. This means that $\mathcal{E}[\|XQ_{:i}\|^2]$ rapidly becomes comparable to or smaller than $\mathcal{E}[\|NQ_{:i}\|^2]$, which means that a regularized estimate of $XQ_{:i}$ in (3.2) is essentially zero unless i is small.

In summary, for $Y = AX + N$ with $\mathcal{E}[N'N] = \epsilon^2 I$, we have the following algorithm. (1) Compute the vectors $Y\hat{Q}_{:i}$, $i = 1, \dots, n$, where $\hat{Q}$ is the matrix of right singular vectors in the SVD of data matrix Y . (2) If the i th singular value of Y is not dominated by noise, compute $(X\hat{Q}_{:i})_{\text{reg}}$ as the regularized estimate for the isolated equation $Y\hat{Q}_{:i} = A(X\hat{Q}_{:i}) + N\hat{Q}_{:i}$, using any desired linear or nonlinear regularization method. If the i th singular value of Y is dominated by noise, then $(X\hat{Q}_{:i})_{\text{reg}}$ is taken to be the zero vector. If it is uncertain whether or not the i th singular value is dominated by noise, then one proceeds with a regularized treatment of the i th equation as above. However, if a regularized estimate of the i' th equation has a very small magnitude, or the regularization parameter used is large, then $(X\hat{Q}_{:i})_{\text{reg}}$ is taken to be the zero vector for all $i > i'$ (in view of the exclusive importance of the dominating singular values and the associated singular vectors, the Lanczos bidiagonalization algorithm is a particularly attractive method for treatment of the SVD computation [6]). (3) Form the matrix $(X\hat{Q})_{\text{reg}}$, whose i th column is $(X\hat{Q}_{:i})_{\text{reg}}$. (4) Compute the estimate of X as $X_{\text{reg}} = (X\hat{Q})_{\text{reg}}(\hat{Q})'$.

We can now consider the comparative complexity of various approaches to $Y = AX + N$ (i.e., treatment of (1.4)). We assume that A is an $m \times m$ matrix, Y is an $m \times n$ matrix, and $m \geq n$. (1) The general MAP estimate (2.1) requires performing various matrix multiplications and a factorization of a (not necessarily sparse) $mn \times mn$ matrix, leading to $O(m^3n^3)$ complexity. This is also the nominal complexity of the joint regularization approach [2] (which specifies that the autocovariance matrix of $\text{vec}(X)$ is the inverse of expression (1.5)). (2) The Kalman filter (via the Markov assumption) results in a marked complexity reduction. Its complexity will be greater than or equal to the complexity of independently regularizing the n equations of sequence (1.2) (ICM). Treatment of each such equation involves multiplication and factorization of $m \times m$ matrices. Since this is repeated for each of the n equations, the

complexity is at least $O(m^3n)$. (3) The method described in the preceding paragraph requires computing the SVD of Y (complexity $O(m^3)$). The treatment of each of the n equations (3.2) involves multiplication and factorization of $m \times m$ matrices (complexity $O(m^3)$). Given that Y has up to n nonzero singular values, this would presumably lead to complexity $O(m^3n)$ for the method of the preceding paragraph. However, in practice, the singular values of Y quickly become dominated by noise, so that only a few (e.g., perhaps only three or four or five) equations of sequence (3.2) need be treated, rather than n , i.e., see step 2 of the preceding paragraph. Thus, in practical circumstances, the complexity of the method of the preceding paragraph can be considered to be $O(m^3)$ versus $O(m^3n^3)$ for the general Wiener or joint regularization, and $O(m^3n)$ for the Kalman or the ICM methods.

Finally, if there is some a priori temporal information available, so that one is supplied with a prior nontrivial estimate of a matrix proportional to $\mathcal{E}[X'X]$, then $\hat{Q}$ in the above algorithm should be taken as the orthogonal matrix diagonalizing this latter matrix estimate, rather than the matrix of right singular vectors of Y .

3.3. The regularization parameter operator. The autocovariance matrix of $\text{vec}(X)$ is given by (2.10). As noted above, $(\mathcal{E}[XX']/\mathcal{E}[\|X\|^2])^{-1}$ can be understood to be a Tikhonov regularizing operator, relevant to the columns of X . The eigenvectors of $\mathcal{E}[X'X]$ are the columns of the eigenvector matrix of $(\mathcal{E}[Y'Y] - [N'N])$ (Corollary 2.12). The i th eigenvalue of $\mathcal{E}[X'X]$ is $\gamma_i^2 = \mathcal{E}[(Q_{:i})'X'XQ_{:i}]$, which is also the expectation of the square of the signal norm for the i th equation of (3.2). Hence, given (3.1) (the sequence of noise columns is white) the optimal regularization parameter for the i th equation of sequence (3.2) is ϵ^2/γ_i^2 . The eigenvalues of $\mathcal{E}[X'X]$ are determined by the optimal regularization parameters for (3.2) (which can be estimated by standard methods (e.g., [7, 13])). Thus, rather than taking $\mathcal{E}[X'X]$ to be proportional to $(\mathcal{E}[Y'Y] - \mathcal{E}[N'N])$ (as in (2.18)), one can anticipate improved performance by taking its eigenvectors to be those of the latter expression, and selecting the individual eigenvalues in the above a posteriori fashion.

Because of identification of its eigenvalues with the relevant regularization parameters, and use of its eigenvectors to effect a complexity reduction, it seems appropriate to designate $\mathcal{E}[X'X]$ as the “regularization parameter operator”—the generalization of the regularization parameter of an ordinary inverse problem to the problems with the Kronecker product structure $I \otimes A$.

We note in passing that our treatment of the statistics of $\text{vec}(X)$ is relevant not only to inverse problems with the operator $I \otimes A$ but also to inverse problems with an operator of the form $\sum_i (B_i \otimes A_i)$, particularly when at least one of the matrices of the sets $\{A_i\}$, $\{B_i\}$ is well-conditioned.

3.4. The case of injective A . Assuming A is noiseless, the component of any column of Y that is not in the range of A cannot be fit by any column of any candidate for the source matrix X . Evidently, this component of Y must represent noise. This noise component is easily eliminated from the data. Thus, we can replace $Y = AX + N$ by

$$(3.4) \quad A[(A'A)^\dagger A']Y = AX + N,$$

where the pseudoinverse of A (in brackets on the left-hand side above) is efficiently supplied via the SVD of A . $\hat{Q}$ could then be taken to be the orthogonal matrix diagonalizing $Y'A(A'A)^\dagger A'Y$ (of course, if A is surjective, then $Y'A(A'A)^\dagger A'Y = Y'Y$).

3.5. The case of transfer matrix noise. The requisite Q , used to produce the sequence of independent equations (3.2), is orthogonal even under the additional complication that there is white Gaussian noise in the transfer matrix A . Thus, suppose

$$(3.5) \quad Y = (A + N_2)X + N_1.$$

Note that the total noise $N = N_1 + N_2X$ is not independent of the signal X . Suppose N_1 and N_2 are each composed of zero mean independent and identically distributed jointly Gaussian random variables and X, N_1 , and N_2 are independent of each other. We then have that

$$(3.6) \quad \mathcal{E}[N'N] = \mathcal{E}[N_1'N_1] + \mathcal{E}[X'N_2'N_2X] = \epsilon_1^2 I + \epsilon_2^2 \mathcal{E}[X'X].$$

But

$$(3.7) \quad \mathcal{E}[Y'Y] = \mathcal{E}[(AX)'AX] + \epsilon_1^2 I + \epsilon_2^2 \mathcal{E}[X'X].$$

From row-isotropy, $\mathcal{E}[(AX)'AX]$ is proportional to $\mathcal{E}[X'X]$. Thus, (3.7) implies that $\mathcal{E}[X'X]$ is a linear combination of $\mathcal{E}[Y'Y]$ and I . This and (3.6) imply that $\mathcal{E}[N'N]$ is also a linear combination of $\mathcal{E}[Y'Y]$ and I . Thus, if Q is the orthogonal matrix diagonalizing $\mathcal{E}[Y'Y]$, then (as before) the individual equations of (3.2) are independent of each other, and we again have the complexity reduction.

4. A numerical illustration. In this section, we will use continuous notation only because it eases formal manipulation in the context of a simulation based on Legendre polynomials and sine functions. The following is not intended to be a rigorous presentation regarding the general Hilbert space setting.

Consider

$$(4.1) \quad h(y, t) = \int_X f(x, y)g(x, t) dx.$$

With respect to our earlier discrete notation (i.e., in (1.1)), a noisy version of $h(y, t)$ (assumed known) corresponds to Y , $\int_X f(x, y)(\cdot)dx$ (assumed known) corresponds to A , and $g(x, t)$ (required to be estimated) corresponds to X . Given that all the above functions are square-integrable, we can write the relevant singular value expansions as

$$(4.2) \quad \begin{aligned} h(y, t) &= \sum_j b_j \alpha_j(t) \beta_j(y), \\ f(x, y) &= \sum_i \lambda_i \phi_i(x) \psi_i(y). \end{aligned}$$

It follows that

$$(4.3) \quad g(x, t) = \sum_i \frac{\phi_i(x)}{\lambda_i} \int_Y h(y, t) \psi_i(y) dy,$$

$$(4.4) \quad = \sum_j \alpha_j(t) \sum_i \frac{\phi_i(x)}{\lambda_i} \int_Y b_j \beta_j(y) \psi_i(y) dy.$$

Ultimately, we imagine there to be contamination by additive white Gaussian noise $\epsilon(y, t)$. Assuming the orthonormal sets of functions in the above singular value expansions are complete, we can then write a noise realization as

$$(4.5) \quad \epsilon(y, t) = \sum_{i,j} \epsilon_{ij} \alpha_j(t) \psi_i(y)$$

with $\mathcal{E}[\epsilon_{ij}^2]$ uniform over all i, j .

Replacing $h(y, t)$ on the right-hand side of (4.3) by $h(y, t) + \epsilon(y, t)$ (so that we can treat the analogue of (1.1) written in continuous notation), the unregularized (maximum likelihood) estimate of $g(y, t)$ is then seen to be

$$(4.6) \quad g_{\text{ml}}(x, t) = \sum_j \alpha_j(t) \sum_i \frac{\phi_i(x)}{\lambda_i} \left(\int_Y b_j \beta_j(y) \psi_i(y) dy + \epsilon_{ij} \right).$$

We assume the setting of the zero-order Tikhonov regularization for the ordinary inverse problem. In the discretized setting, where $g(x, t)$ is a matrix, this corresponds to the assumption that each column of this matrix (a discretized function of a spatial variable) has autocovariance proportional to the identity. Under isotropic conditions relating to the discretized setting, the methodology of this paper thereby essentially supplies the regularized estimate of $g(x, t)$ as

$$(4.7) \quad g_{\text{map}}(x, t) = \sum_j \alpha_j(t) \sum_i \frac{\lambda_i}{\lambda_i^2 + r_j} \phi_i(x) \left(\int_Y b_j \beta_j(y) \psi_i(y) dy + \epsilon_{ij} \right),$$

where r_j is the optimal regularization parameter for the equation

$$b_j \beta_j(y) = \int_X f(x, y) \left(\int_T g(x, t) \alpha_j(t) dt \right) dx + \sum_i \epsilon_{ij} \psi_i(y).$$

r_j is the inverse of the square of the signal-to-noise ratio for this equation—which in our numerical simulation is known from our prior knowledge of $g(x, t)$ and the noise power. Thus, $g_{\text{map}}(x, t)$ is the solution estimate based on an optimal selection of the temporal eigenfunctions ($\{\alpha_k(t)\}$) and eigenvalues (regularization parameters).

On the other hand, the method whereby regularized solutions are obtained independently of each other sequentially in time gives

$$(4.8) \quad g_{\text{seq}}(x, t) = \sum_i \frac{\lambda_i}{\lambda_i^2 + r(t)} \phi_i(x) \left(\int_Y h(y, t) \psi_i(y) dy + \sum_j \epsilon_{ij} \alpha_j(t) \right),$$

where for each fixed t , $r(t)$ is the optimal regularization parameter for the equation

$$h(y, t) = \int_X f(x, y) g(x, t) dx + \epsilon(y, t).$$

$r(t)$ is known from our prior knowledge of $g(x, t)$ and the noise power. Substituting (4.2) into (4.8), we obtain

$$(4.9) \quad g_{\text{seq}}(x, t) = \sum_j \alpha_j(t) \sum_i \frac{\lambda_i}{\lambda_i^2 + r(t)} \phi_i(x) \left(\int_Y b_j \beta_j(y) \psi_i(y) dy + \epsilon_{ij} \right).$$

This represents the optimal solution estimate for the methodology of sequential independent treatment of the members of the time series of (4.1).

In our numerical example, we take each of the domains of x , y , and t (i.e., X , Y , and T , respectively) to be the interval $(-1, 1)$, and the orthonormal sets of functions to be given by

$$\alpha_n(t) = \sin(\pi nt), \quad \beta_n(y) = \sin(\pi ny)$$

and

$$\phi_m(x) = \sqrt{\frac{2m+1}{2}} P_m(x), \quad \psi_m(y) = \sqrt{\frac{2m+1}{2}} P_m(y),$$

where $P_m(w)$ is the m th Legendre polynomial. We take the singular values to be

$$b_n = \begin{cases} 1/n^2 & \text{if } n \leq 10, \\ 0 & \text{otherwise} \end{cases} \quad \text{and} \quad \lambda_m = \begin{cases} 1/m & \text{if } m \leq 10, \\ 0 & \text{otherwise.} \end{cases}$$

We compute $g_{\text{map}}(x, t)$ and $g_{\text{seq}}(x, t)$ (as in (4.7) and (4.9)) for different noise realizations over a range of noise power. All calculations are performed using Mathematica 4.0 [14]. The zero mean Gaussian pseudonoise terms ϵ_{ij} in (4.7) and (4.9) are obtained by combining a random number generator (the Mathematica 4.0 function “Random[.]”) and the inverse error function (the Mathematica 4.0 function “InvErf[.]”) via a standard algorithm [11].

Table 4.1 shows the mean and standard deviation of relative errors of $g_{\text{map}}(x, t)$ and $g_{\text{seq}}(x, t)$ over a wide range of signal-to-noise ratios (SNR).

TABLE 4.1
Simulation results: Relative error^a.

SNR	g_{map} error	g_{seq} error
17824	0.0048 ± 0.00051	0.0050 ± 0.00065
8912	0.0093 ± 0.0010	0.0095 ± 0.0012
4456	0.017 ± 0.0016	0.019 ± 0.0022
2228	0.032 ± 0.0045	0.039 ± 0.0043
1114	0.061 ± 0.010	0.075 ± 0.0090
557	0.11 ± 0.0098	0.15 ± 0.015
278	0.21 ± 0.031	0.30 ± 0.016
139	0.32 ± 0.035	0.62 ± 0.063
70	0.54 ± 0.067	1.1 ± 0.096
35	0.78 ± 0.13	1.8 ± 0.23
17	1.1 ± 0.16	2.1 ± 0.18

^aResults are in the form of “mean relative error ± one standard deviation.” The statistics are determined from 10 different noise realizations at each noise power. SNR is the ratio of the L^2 norms of the right-hand sides of (4.4) and (4.5).

Figure 4.1 shows plots of $f(x, y)$, $g(x, t)$, and $h(y, t)$. For one of the noise realizations at SNR = 139, solution estimate time series at different spatial locations are shown in the first three columns of Figure 4.2, and spatial plots at different time points are shown in the last three columns. For this particular noise realization, the relative error of $g_{\text{map}}(x, t)$ was 0.39, and the relative error of $g_{\text{seq}}(x, t)$ was 0.52.

The symmetry about the origin of all plots in the first three columns of Figure 4.2 is due to the fact that both $h(y, t)$ and $\epsilon(y, t)$ are expanded in $\{\alpha_k(t)\}$,

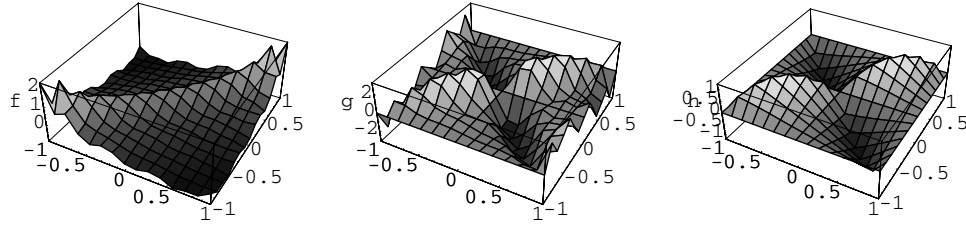
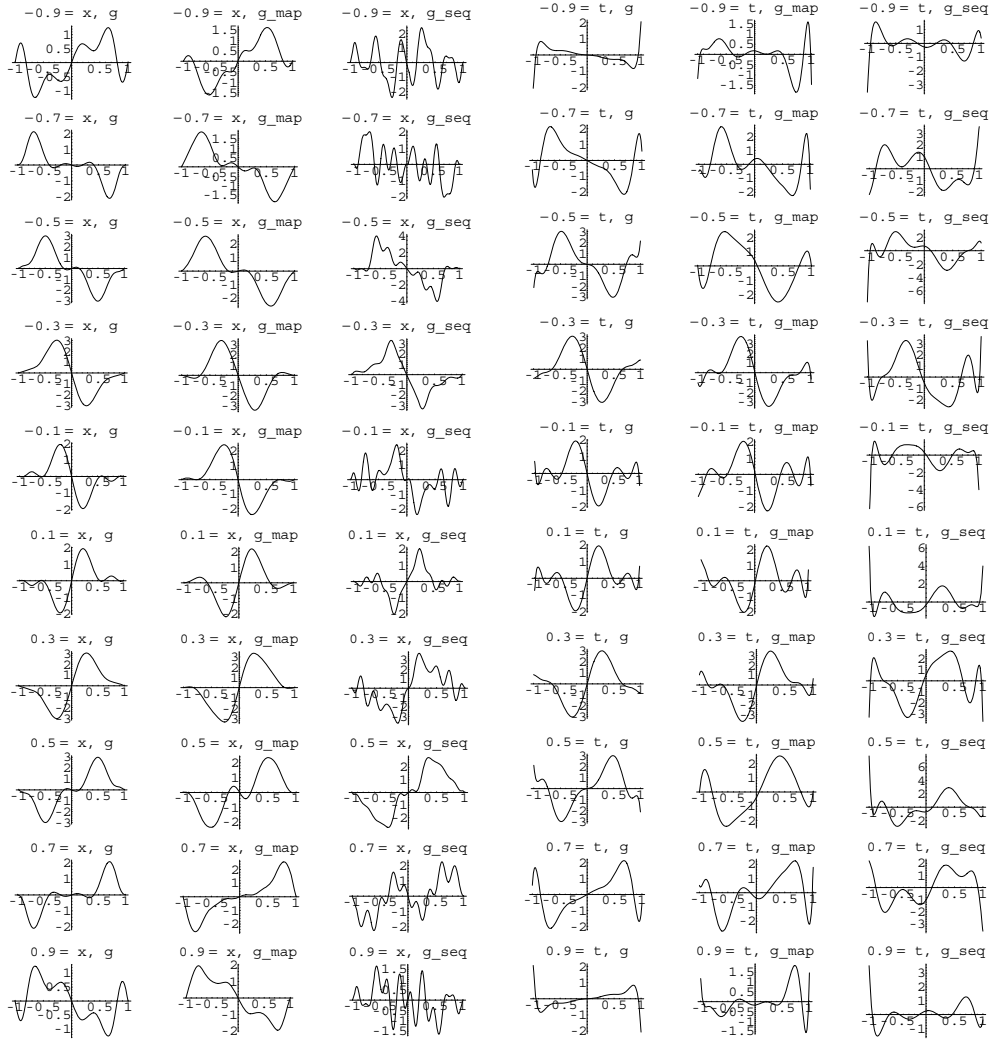
FIG. 4.1. Plots of $f(x, y)$ (left), $g(x, t)$ (middle), and $h(y, t)$ (right).

FIG. 4.2. Columns 1–3: for one of the noise realizations at $\text{SNR} = 139$, each row shows the time series of the functions $g(x, t)$, $g_{\text{map}}(x, t)$, and $g_{\text{seq}}(x, t)$ for a different fixed value of x : -0.9 in the uppermost row, -0.7 in the next row, and so on as indicated. Columns 4–6: for the same noise realization as in the first three columns, each row shows spatial plots of the functions $g(x, t)$, $g_{\text{map}}(x, t)$, and $g_{\text{seq}}(x, t)$ for a different fixed value of t : -0.9 in the uppermost row, -0.7 in the next row, and so on as indicated.

each of whose members is symmetric about the origin (sine functions). The spatial plots of $g(x, t)$, seen in column 4 of Figure 4.2, are symmetric around the origin because $h(y, t)$ is symmetric around the origin (for each fixed t)—and, therefore, only the Legendre polynomials that are symmetric about the origin (rather than symmetric about the ordinate axis) have nonzero coefficients in the expansion of $g(x, t)$ in (4.3). However, the other Legendre polynomials have nonzero coefficients in the expansions of $g_{\text{map}}(x, t)$ and $g_{\text{seq}}(x, t)$ (the fifth and sixth columns of Figure 4.2), because of the presence of the terms ϵ_{ij} in calculation of the Fourier coefficients. Thus, the degree of symmetry loss in the plots in the fifth and sixth columns of Figure 4.2 is another visual clue to the quality of the spatial regularization.

Obviously, for very low noise levels (where the maximum likelihood methodology is appropriate in a discretized setting), and for “high” noise levels (where all methods give useless results), there will be no practical difference between $g_{\text{seq}}(x, t)$ and $g_{\text{map}}(x, t)$. In the intermediate zone, the table and figures indicate that the new method is capable of very impressive improvements in the fidelity and utility of the estimate. This is certainly most marked in terms of the time domain, shown in the first three columns of Figure 4.2, though a general improvement of solution spatial plots at fixed points in time is also appreciated, shown in the last three columns of Figure 4.2. The method varies stably with perturbation of the regularization parameter operator (variation in estimates of $\{\epsilon^2/\gamma_i^2\}$ and the eigenvectors of $\mathcal{E}[Y'Y]$), since minimum-mean-square-error estimates vary stably with perturbation of the covariance matrices. Thus, preference for this methodology will extend to the nonoptimal choices of the regularizing operator and regularization parameter operator. This is analogous to the efficacy of application of a Wiener filter formalism in the white Gaussian setting, when neither the signal-to-noise ratio (the regularization parameter) nor the signal autocovariance matrix is precisely known—the setting of the Tikhonov regularization [12].

5. Comparison with other methods. Problems of the form (1.1) have been extensively considered in engineering contexts. State space methods (using (1.2) as a measurement model) have played a leading role in this regard. These are highly efficacious and have abundant justification in settings where a process model is evident, or can be well-specified utilizing methods of system identification [9]. However, in the general case, a process model is not given (e.g., valid dynamical or evolution equations may not be available). In such circumstances, process models such as the following have been imposed:

$$(5.1) \quad X_{:,i+1} = \alpha X_{:,i} + u_i,$$

where u_i is the process noise (the choice $\alpha = 1$ gives the so-called random walk process model). The above process model happens to imply that X is isotropic (all column autocovariance matrices are proportional to each other), so that $C_{\text{vec}(X)} = T \otimes S$.

Evidently, the model corresponds to prior imposition of a particular choice for T (depending on the value of α). In general, there is no more justification for any such choice than there is justification for the independent columns method (which, in fact, results from the choice of $\alpha = 0$ in the above process model). The implied choices of row covariance matrices are inconsistent with Corollary 2.12. Furthermore, process models based on higher order autoregressive assumptions, in addition to imposing a questionable choice for the autocovariance of $\text{vec}(X)$ in the sense of Corollary 2.12, also violate Theorem 2.9, since $C_{\text{vec}(X)}$ is then no longer separable (the covariance matrices between different columns of X are no longer proportional to each other).

A different approach, based on a generalization of the Tikhonov regularization, is that of joint regularization [2] (as noted in section 1). This fails to be consistent with separability unless $R \propto I$ in (1.5). In any case, its prior choice for T in (1.5) leads to an implied autocovariance matrix that is, again, inconsistent with Corollary 2.12. Ultimately, in the absence of valid prior constraints on the column sequence of X , choices for $C_{\text{vec}(X)}$ resulting from imposed process models or “temporal” regularizing matrices can be expected to be more inaccurate on average than those derived from the theorems of section 2—since the latter are not wedded to specific prior assumptions of uncertain validity.

The problem posed by (1.1) has also been repeatedly treated in statistical contexts, since it embodies the task of selecting a regression matrix in the setting of multivariate multilinear prediction. A common approach has been to apply “ridge regression” [8] independently to each of the equations of sequence (1.2). This essentially corresponds to ICM with the zero-order Tikhonov regularization of each member of (1.2). However, the ridge regression assumption that the autocovariance of a column of X is proportional to the identity can be extended to the assumption that the rows of X are independent. The latter can be exploited to eliminate ICM and better adapt the ridge regression to the multivariate setting [3]. Thus, given the singular value decomposition $A = UD_aV'$, multiplication of both sides of $Y = AX + N$ by U' leads to $U'Y = D_a(V'X) + U'N$, and

$$(5.2) \quad (U'Y)_{i:} = d_i(V'X)_{i:} + (U'N)_{i:},$$

$i = 1, 2, \dots, m$, with d_i the i th diagonal entry of D_a . Note that the rows of $(V'X)$ are independent since V is orthogonal and the above (extended) ridge regression assumption is that X has independent rows (the rows of $U'N$ are also independent, because N is subject to the same assumptions as X). Since $(U'Y)_{i:}$ is a noisy version of something proportional to $(V'X)_{i:}$, one can now apply the standard filter methodology to each of the equations of sequence (5.2) independently of the other equations of the sequence. However, this approach is inextricably tied to the assumption that each column of X is white (and its rows are independent)—since it is only in that circumstance that the members of equation sequence (5.2) are independent. In the general case where the column autocovariance is not proportional to the identity, there is no way to gain access to the left-hand side of X to “whiten” its columns. Thus, the derivation supplied is invalid beyond the ridge regression assumption and cannot be applied to the general problem expressed by (1.4). Furthermore, there are no results comparable to the theorems and corollaries of section 2. Finally, the approach apparently requires that A be a matrix, while the approach presented here in section 2 directly generalizes to the nonlinear setting (as noted in section 3.1).

More recently, the problem of regression matrix selection has been addressed in [1], which presents both a particular approach and an appendix consisting of 21 separate invited discussions of the problem by 27 investigators. The authors’ method, presented in the body of the paper, first proposes a particular treatment of the problem with well-conditioned A . It then uses that method to treat the case of ill-conditioned A , by inputting to it the solution estimate obtained from the standard ICM ridge regression (with the same value of the ridge parameter for all columns). That approach and the others discussed in the appendix of the paper are distinct from the method presented here.

6. Conclusions. We have identified special features of inverse problems when the operator has $I \otimes A$ Kronecker structure. Inherent in such a structure is the fact

that the unknown can be put in the matrix form—reflecting the distinction between “variables” subject to, versus not subject to, the ill-conditioned operator A . One can thereby exploit the column space versus row space structure of the unknown. Symmetries on these spaces, based on the absence of supplied constraints, lead to a robust generalization of the Tikhonov regularization, wherein the solution estimate is derived from a regularization parameter operator which generalizes the ordinary inverse problems notion of a regularization parameter. The theory has been evaluated with a detailed numerical example, which suggests that significant accuracy improvements are possible using the implied methodology. It is also observed that marked computational complexity reduction over standard methodologies typically accrues under suitable noise conditions.

REFERENCES

- [1] L. BREIMAN AND J. H. FRIEDMAN, *Predicting multivariate responses in multiple linear regression*, J. Roy. Statist. Soc. Ser. B, 59 (1997), pp. 3–54.
- [2] D. H. BROOKS, G. AHMAD, R. S. MACLEOD, AND G. M. MARATOS, *Inverse electrocardiography by simultaneous imposition of multiple constraints*, IEEE Trans. Biomed. Eng., BME-46 (1999), pp. 3–18.
- [3] P. BROWN AND J. ZIDEK, *Adaptive multivariate ridge regression*, Ann. Statist., 8 (1980), pp. 64–74.
- [4] B. DE MOOR AND H. ZHA, *A tree of generalizations of the ordinary singular value decomposition*, Linear Algebra Appl., 147 (1991), pp. 469–500.
- [5] M. FOSTER, *An application of the Wiener–Kolmogorov smoothing theory to matrix inversion*, J. SIAM, 9 (1961), pp. 387–392.
- [6] G. H. GOLUB AND C. F. VAN LOAN, *Matrix Computations*, 3rd ed., Johns Hopkins University Press, New York, 1996, pp. 495–496.
- [7] P. C. HANSEN, *Numerical tools for analysis and solution of Fredholm integral equations of the first kind*, Inverse Problems, 8 (1992), pp. 849–872.
- [8] A. E. HOERL AND R. W. KENNARD, *Ridge regression: Biased estimation for nonorthogonal problems*, Technometrics, 12 (1970), pp. 55–67.
- [9] A. SEPPANEN, M. VAUHKONEN, P. J. VAUHKONEN, E. SOMERSALO, AND J. P. KAPIO, *State estimation with fluid dynamical evolution models in process tomography—An application to impedance tomography*, Inverse Problems, 17 (2001), pp. 467–483.
- [10] R. E. KALMAN, *A new approach to linear filtering and prediction problems*, Trans. ASME, 82 (1960), pp. 35–45.
- [11] W. H. PRESS, S. A. TEUKOLSKY, W. T. VETTERLING, AND B. P. FLANNERY, *Numerical Recipes in C*, 2nd ed., Cambridge University Press, Cambridge, UK, 1992.
- [12] A. TIKHONOV AND V. ARSENIN, *Solutions of Ill-Posed Problems*, John Wiley and Sons, New York, 1977.
- [13] G. WAHBA, *Practical approximated solutions to linear operator equations when the data are noisy*, SIAM J. Numer. Anal., 14 (1977), pp. 651–667.
- [14] S. WOLFRAM, *The Mathematica Book*, 3rd ed., Wolfram Media/Cambridge University Press, Cambridge, UK, 1996.

A GLOBAL SOLUTION FOR THE STRUCTURED TOTAL LEAST SQUARES PROBLEM WITH BLOCK CIRCULANT MATRICES*

AMIR BECK[†] AND AHARON BEN-TAL[†]

Abstract. We study the structured total least squares (STLS) problem of system of linear equations $\mathbf{Ax} = \mathbf{b}$, where $\mathbf{A}$ has a block circulant structure with N blocks. We show that by applying the discrete Fourier transform (DFT), the STLS problem decomposes into N *unstructured* total least squares (TLS) problems. The N solutions of these problems are then assembled to generate the optimal global solution of the STLS problem. Similar results are obtained for elementary block circulant matrices. Here the optimal solution is obtained by assembling two solutions: one of an *unstructured* TLS problem and the second of a multidimensional TLS problem.

Key words. structured total least squares, block circulant matrices, discrete Fourier transform

AMS subject classifications. 65F20, 65T50

DOI. 10.1137/040612233

1. Introduction. Many problems in data fitting and estimation give rise to an overdetermined system of linear equations $\mathbf{Ax} \approx \mathbf{b}$, where both the matrix $\mathbf{A}$ and the vector $\mathbf{b}$ are contaminated by noise. The total least squares (TLS) approach to this problem [4, 1, 6] is to seek a perturbation matrix $\Delta\mathbf{A}$ and a perturbation vector $\Delta\mathbf{b}$ that minimize $\|\Delta\mathbf{A}\|^2 + \|\Delta\mathbf{b}\|^2$ subject to the consistency equation $\mathbf{b} - \Delta\mathbf{b} \in \text{Range}(\mathbf{A} - \Delta\mathbf{A})$.¹ The TLS approach was extensively used in a variety of scientific disciplines such as signal processing, automatic control, statistics, physics, economic, biology, and medicine. One of the main reasons for the wide use of TLS is the fact that the problem has essentially an explicit solution, expressed by the singular value decomposition (SVD) of the augmented matrix $(\mathbf{A}, \mathbf{b})$ [4, 1, 6] (cf. section 2).

In many applications, the matrix $\mathbf{A}$ has a specific structure, e.g., Toeplitz or Hankel, which imposes a requirement on the perturbation matrix $\Delta\mathbf{A}$ to possess a corresponding special structure. The TLS solution does not take into account this requirement, and consequently many methods addressing the structured TLS (STLS) problem were introduced in the literature [7, 8, 9, 5, 10, 11].

The STLS problem, even for a *structured* matrix $\Delta\mathbf{A}$ (i.e., a structure that can be represented as $\mathcal{L}(\Delta\mathbf{A}) = 0$, where $\mathcal{L}$ is a linear operator) is generically a nonconvex problem, and as a result the algorithms designed to solve it converge at best to a local solution [7, 9, 5]. This state of affairs is prevailing even for the special case of Toeplitz–Hankel structure [10, 11]. An exception is the case where some given columns of $\mathbf{A}$ must remain fixed (i.e., the corresponding columns of $\Delta\mathbf{A}$ are zero). A global solution for this case is obtained in [12] in terms of the SVD of an appropriate matrix.

*Received by the editors July 26, 2004; accepted for publication (in revised form) by S. Van Huffel January 27, 2005; published electronically August 31, 2005. This research was partially supported by BSF grant 2002038. The U.S. Government retains a nonexclusive, royalty-free license to publish or reproduce the published form of this contribution, or allow others to do so, for U.S. Government purposes. Copyright is owned by SIAM to the extent not limited by these rights.

<http://www.siam.org/journals/simax/27-1/61223.html>

[†]MINERVA Optimization Center, Department of Industrial Engineering, Technion—Israel Institute of Technology, Haifa 32000, Israel (becka@tx.technion.ac.il, morbt@ie.technion.ac.il).

¹Here and elsewhere in this paper a matrix norm is always the Frobenius norm, and a vector norm is the Euclidean one.

In this paper we study the STLS problem, where the matrix $\mathbf{A}$ has either a block circulant (BC) structure,

$$\mathbf{A} = \begin{pmatrix} \mathbf{A}_0 & \mathbf{A}_1 & \cdots & \mathbf{A}_{N-1} \\ \mathbf{A}_{N-1} & \mathbf{A}_0 & \cdots & \mathbf{A}_{N-2} \\ \vdots & \vdots & & \vdots \\ \mathbf{A}_1 & \mathbf{A}_2 & \cdots & \mathbf{A}_0 \end{pmatrix},$$

or a (more special) block elementary circulant structure (EBC),

$$\mathbf{A} = \begin{pmatrix} \mathbf{A}_0 & \mathbf{A}_1 & \cdots & \mathbf{A}_1 \\ \mathbf{A}_1 & \mathbf{A}_0 & \cdots & \mathbf{A}_1 \\ \vdots & \vdots & & \vdots \\ \mathbf{A}_1 & \mathbf{A}_1 & \cdots & \mathbf{A}_0 \end{pmatrix},$$

where $\mathbf{A}_0, \mathbf{A}_1, \dots, \mathbf{A}_{N-1}$ are $m \times n$ matrices. A particular example from signal processing, which gives rise to elementary block circulant structure, is described next.

Consider the problem of estimating N unknown vectors $\mathbf{x}_k, 0 \leq k \leq N-1$, from N vector observations $\mathbf{b}_k, 0 \leq k \leq N-1$, where each observation vector $\mathbf{b}_k$ is related to all of the parameter vectors $\mathbf{x}_0, \mathbf{x}_1, \dots, \mathbf{x}_{N-1}$ through the linear model

$$(1) \quad \mathbf{b}_k = \mathbf{A}_0 \mathbf{x}_k + \sum_{i \neq k} \mathbf{A}_1 \mathbf{x}_i + \mathbf{w}_k, \quad 0 \leq k \leq N-1.$$

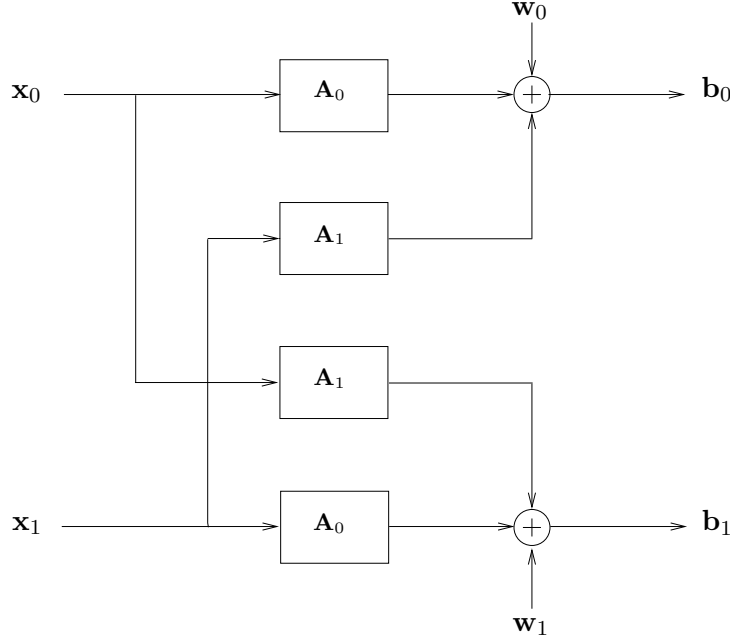
Here $\mathbf{A}_0$ and $\mathbf{A}_1$ represent the direct and cross transfer matrices, respectively, and $\mathbf{w}_k$ is the k th noise vector. For the two channel case ($N = 2$), this system is illustrated in Figure 1.

System (1) reflects the situation where the effect of all of the interfering vectors $\mathbf{x}_i, i \neq k$ on the k th output $\mathbf{b}_k$ is the same (i.e., independent of k).

Systems with BC structure appear in the context of multichannel signal estimation [13, 14], image restoration [15], cyclic convolution filter banks [3], texture synthesis and recognition [16], and more.

We will present efficient algorithms, which obtain a (global) solution for both the BC and the EBC cases. The analysis relies heavily on the theory of discrete Fourier transform (DFT) for block circulant matrices. Elements of this theory that are needed for our purposes are collected in section 3. We show that for the BC case, under the DFT, the original STLS problem, which is of size $Nm \times Nn$, decomposes into N (unstructured) TLS problems of size $m \times n$. The solution is then obtained by solving the N small problems (possibly in parallel), using the SVD of each system, and then taking the inverse DFT. We thus obtain that the solution of the STLS problem with a BC matrix is explicitly expressed by N singular value decomposition of N appropriate matrices (N being the number of different blocks in $\mathbf{A}$). The solution of the EBC is similarly derived with one exception. In the EBC case the STLS problem decomposes under the DFT to two smaller problems: a TLS problem and a multidimensional TLS problem (cf. section 2).

The paper is organized as follows. In section 2 we briefly review both the classical TLS problem and the multidimensional TLS problem. We recall the SVD-based solution of both problems. Section 3 contains a summary of the results on the DFT of BC matrices and block vectors. Sections 4 and 5 present the solution of the STLS problem with BC and EBC matrices, respectively. Section 6 presents computational

FIG. 1. *Two channel model.*

results that demonstrate the fact that the algorithm devised in this paper gives the global optimum while other algorithms in the literature do not necessarily converge to the global optimum.

The results in this paper are valid both for the real and complex case. We denote by $\mathbb{F}$ the real field ($\mathbb{R}$) or the complex field ($\mathbb{C}$). Vectors in $\mathbb{F}^m$ are denoted by boldface lowercase letters, e.g., $\mathbf{y}$. Matrices in $\mathbb{F}^{m \times n}$ are denoted by boldface uppercase letters e.g., $\mathbf{A}$. The Hermitian conjugate and the transpose of a matrix $\mathbf{A}$ are denoted by $\mathbf{A}^*$ and $\mathbf{A}^T$, respectively. Note that when $\mathbb{F} = \mathbb{R}$, $\mathbf{A}^* = \mathbf{A}^T$. The boldface letter $\mathbf{i}$ denotes $\sqrt{-1}$. For a positive integer N and an integer j , we denote by $[j]_N$ the value of j modulo N , e.g., $[N]_N = 0$, $[-2]_N = N - 2$.

2. Review of the TLS Method. For the sake of completeness, we briefly review the known results on the (unstructured) TLS and the multidimensional TLS problems [4, 1, 6]. Given a linear system $\mathbf{A}\mathbf{x} \approx \mathbf{b}$, where $\mathbf{A} \in \mathbb{F}^{m \times n}$ ($m > n$), $\mathbf{b} \in \mathbb{F}^m$, and $\mathbf{x} \in \mathbb{F}^n$. The TLS problem is to find a perturbation matrix $\Delta\mathbf{A} \in \mathbb{F}^{m \times n}$ and a perturbation vector $\Delta\mathbf{b} \in \mathbb{F}^m$ of minimum norm such that the system $(\mathbf{A} - \Delta\mathbf{A})\mathbf{x} = \mathbf{b} - \Delta\mathbf{b}$ is consistent. More precisely, for some positive constant $\alpha > 0$ we seek to solve the following minimization problem:

$$\begin{aligned} \text{(TLS)} \quad & \min_{\Delta\mathbf{A}, \Delta\mathbf{b}, \mathbf{x}} \|\Delta\mathbf{A}\|^2 + \alpha \|\Delta\mathbf{b}\|^2 \\ & \text{subject to } (\mathbf{A} - \Delta\mathbf{A})\mathbf{x} = \mathbf{b} - \Delta\mathbf{b}. \end{aligned}$$

The algorithm for the solution of this problem was derived in [4] and is based on SVD calculation.

ALGORITHM TLS (see [4]).

Input: $\mathbf{A} \in \mathbb{F}^{m \times n}$, $\mathbf{b} \in \mathbb{F}^m$, $\alpha > 0$.

Output: $\Delta \mathbf{A} \in \mathbb{F}^{m \times n}$, $\Delta \mathbf{b} \in \mathbb{F}^m$, $\mathbf{x} \in \mathbb{F}^n$.

1. Calculate the SVD $(\mathbf{A}, \sqrt{\alpha}\mathbf{b}) = \mathbf{U}\Sigma\mathbf{V}^*$, where $\mathbf{V} \in \mathbb{F}^{(n+1) \times (n+1)}$ is a unitary matrix, $\mathbf{U} \in \mathbb{F}^{m \times (n+1)}$ is a matrix that satisfies $\mathbf{U}^*\mathbf{U} = \mathbf{I}$, and $\Sigma = \text{diag}(\sigma_1, \sigma_2, \dots, \sigma_{n+1})$, where $\sigma_1, \dots, \sigma_{n+1} > 0$.
2. If $\sigma_n(\mathbf{A}) > \sigma_{n+1}(\mathbf{A}, \sqrt{\alpha}\mathbf{b})$, then the solution of (TLS) is given by

$$(\Delta \mathbf{A}, \Delta \mathbf{b}) = \sigma_{n+1} \mathbf{u}_{n+1} \mathbf{v}_{n+1}^T \underbrace{\text{diag}(1, 1, \dots, 1, \sqrt{\alpha})}_{n \text{ times}},$$

$$\mathbf{x} = -\frac{1}{\sqrt{\alpha}V_{n+1,n+1}}(V_{1,n+1}, \dots, V_{n,n+1})^T,$$

where $\mathbf{u}_{n+1}$ and $\mathbf{v}_{n+1}$ are the $(n+1)$ th columns of $\mathbf{U}$ and $\mathbf{V}$, respectively, and for every i, j , $V_{i,j}$ is the (i,j) th component of $\mathbf{V}$.

A known generalization of the TLS problem deals with the case in which we have multiple right-hand side vectors; i.e., we are given k linear systems $\mathbf{A}\mathbf{x}_1 \approx \mathbf{b}_1$, $\mathbf{A}\mathbf{x}_2 \approx \mathbf{b}_2, \dots, \mathbf{A}\mathbf{x}_k \approx \mathbf{b}_k$. Here we seek to find minimum weight perturbations $\Delta \mathbf{A} \in \mathbb{F}^{m \times n}$ and $\Delta \mathbf{b}_1, \dots, \Delta \mathbf{b}_k \in \mathbb{F}^m$ such that the k linear systems $(\mathbf{A} - \Delta \mathbf{A})\mathbf{x}_i = \mathbf{b}_i - \Delta \mathbf{b}_i$, $i = 1, 2, \dots, k$, are consistent. Denote $\mathbf{B} = (\mathbf{b}_1, \mathbf{b}_2, \dots, \mathbf{b}_k)$ and $\mathbf{X} = (\mathbf{x}_1, \mathbf{x}_2, \dots, \mathbf{x}_k)$. The problem is equivalent to finding a minimal weight perturbations $\Delta \mathbf{A} \in \mathbb{F}^{m \times n}$ and $\Delta \mathbf{B} \in \mathbb{F}^{m \times k}$ such that the system $(\mathbf{A} - \Delta \mathbf{A})\mathbf{X} = \mathbf{B} - \Delta \mathbf{B}$ is consistent. This is the *multiple total least squares* (MTLS):

$$\begin{aligned} \text{(MTLS)} \quad & \min_{\Delta \mathbf{A}, \Delta \mathbf{B}, \mathbf{X}} \|\Delta \mathbf{A}\|^2 + \alpha \|\Delta \mathbf{B}\|^2 \\ & \text{subject to } (\mathbf{A} - \Delta \mathbf{A})\mathbf{X} = \mathbf{B} - \Delta \mathbf{B}. \end{aligned}$$

ALGORITHM MTLS (see [6, 1]).

Input: $\mathbf{A} \in \mathbb{F}^{m \times n}$, $\mathbf{B} \in \mathbb{F}^{m \times k}$, $\alpha > 0$.

Output: $\Delta \mathbf{A} \in \mathbb{F}^{m \times n}$, $\Delta \mathbf{B} \in \mathbb{F}^{m \times k}$, $\mathbf{X} \in \mathbb{F}^{n \times k}$.

1. Calculate the SVD $(\mathbf{A}, \sqrt{\alpha}\mathbf{B}) = \mathbf{U}\Sigma\mathbf{V}^*$, where here

$$\mathbf{U} = \begin{pmatrix} \mathbf{U}_1 & \mathbf{U}_2 \end{pmatrix}, \quad \mathbf{V} = \begin{pmatrix} \mathbf{V}_{11} & \mathbf{V}_{12} \\ \mathbf{V}_{21} & \mathbf{V}_{22} \end{pmatrix}, \quad \Sigma = \begin{pmatrix} \Sigma_1 & \mathbf{O} \\ \mathbf{O} & \Sigma_2 \end{pmatrix},$$

and where $\mathbf{U}_1 \in \mathbb{F}^{m \times n}$, $\mathbf{U}_2 \in \mathbb{F}^{m \times k}$, $\mathbf{V}_{11} \in \mathbb{F}^{n \times n}$, $\mathbf{V}_{12} \in \mathbb{F}^{n \times k}$, $\mathbf{V}_{21} \in \mathbb{F}^{k \times n}$, $\mathbf{V}_{22} \in \mathbb{F}^{k \times k}$, $\Sigma_1 \in \mathbb{F}^{n \times n}$, and $\Sigma_2 \in \mathbb{F}^{k \times k}$. Σ_1 and Σ_2 are both diagonal matrices with real positive diagonal, $\mathbf{V}$ is a unitary matrix, and $\mathbf{U}$ satisfies $\mathbf{U}^*\mathbf{U} = \mathbf{I}$.

2. If $\sigma_n(\mathbf{A}) > \sigma_{n+1}(\mathbf{A}, \sqrt{\alpha}\mathbf{B})$, then the solution of (MTLS) is given by

$$\begin{pmatrix} \Delta \mathbf{A} & \Delta \mathbf{B} \end{pmatrix} = \mathbf{U}_2 \Sigma_2 (\mathbf{V}_{12}^T, \mathbf{V}_{22}^T) \mathbf{T}^{-1},$$

$$\mathbf{x} = -\frac{1}{\sqrt{\alpha}} \mathbf{V}_{12} \mathbf{V}_{22}^{-1},$$

$$\text{where } \mathbf{T} = \begin{pmatrix} \mathbf{I}_n & \mathbf{O} \\ \mathbf{O} & \sqrt{\alpha} \mathbf{I}_k \end{pmatrix}$$

3. Block circulant matrices and the DFT. The aim of this section is to give a short summary of results on DFT defined on block circulant matrices and block vectors that are used in the paper. Subsection 3.1 (but not subsection 3.2) is based on [13].

A block circulant matrix is a matrix of the form

$$\mathcal{C}(\mathbf{A}_0, \mathbf{A}_1, \dots, \mathbf{A}_{N-1}) \triangleq \begin{pmatrix} \mathbf{A}_0 & \mathbf{A}_1 & \cdots & \mathbf{A}_{N-1} \\ \mathbf{A}_{N-1} & \mathbf{A}_0 & \cdots & \mathbf{A}_{N-2} \\ \vdots & \vdots & & \vdots \\ \mathbf{A}_1 & \mathbf{A}_2 & \cdots & \mathbf{A}_0 \end{pmatrix},$$

where each submatrix $\mathbf{A}_j$ is a $k \times l$ matrix. The dimensions k and l will be clear from the context and therefore are not part of the notation.

3.1. The DFT of block circulant matrices. From the definition of block circulant matrices we have the following facts.

LEMMA 3.1. Let $\mathbf{A}_0, \mathbf{A}_1, \dots, \mathbf{A}_{N-1} \in \mathbb{F}^{k \times l}$ and $\mathbf{B}_0, \mathbf{B}_1, \dots, \mathbf{B}_{N-1} \in \mathbb{F}^{l \times m}$.

1. $\mathcal{C}^T(\mathbf{A}_0, \mathbf{A}_1, \dots, \mathbf{A}_{N-1}) = \mathcal{C}^*(\mathbf{A}_0, \mathbf{A}_1, \dots, \mathbf{A}_{N-1})$

$$\begin{aligned} \mathcal{C}^T(\mathbf{A}_0, \mathbf{A}_1, \dots, \mathbf{A}_{N-1}) &= \mathcal{C}(\mathbf{A}_0^T, \mathbf{A}_{N-1}^T, \dots, \mathbf{A}_1^T), \\ \mathcal{C}^*(\mathbf{A}_0, \mathbf{A}_1, \dots, \mathbf{A}_{N-1}) &= \mathcal{C}(\mathbf{A}_0^*, \mathbf{A}_{N-1}^*, \dots, \mathbf{A}_1^*); \end{aligned}$$

2. $\mathcal{C}(\mathbf{A}_0, \dots, \mathbf{A}_{N-1})\mathcal{C}(\mathbf{B}_0, \dots, \mathbf{B}_{N-1}) = \mathcal{C}(\mathbf{C}_0, \dots, \mathbf{C}_{N-1})$

$$(2) \quad \mathbf{C}_j = \sum_{i=0}^{N-1} \mathbf{A}_j \mathbf{B}_{[j-i]_N}, \quad 0 \leq j \leq N-1.$$

We now define the DFT and its inverse, which are the main mathematical tools used in the paper.

DEFINITION 3.1. Let $\mathbf{A} = \mathcal{C}(\mathbf{A}_0, \mathbf{A}_1, \dots, \mathbf{A}_{N-1})$ and $\mathbf{F}(\mathbf{A}) = (\mathbf{F}_0(\mathbf{A}), \mathbf{F}_1(\mathbf{A}), \dots, \mathbf{F}_{N-1}(\mathbf{A}))$.

$$\mathbf{F}(\mathbf{A}) = (\mathbf{F}_0(\mathbf{A}), \mathbf{F}_1(\mathbf{A}), \dots, \mathbf{F}_{N-1}(\mathbf{A})),$$

where $\mathbf{F}_j(\mathbf{A}), 0 \leq j \leq N-1$,

$$\mathbf{F}_j(\mathbf{A}) \triangleq \sum_{i=0}^{N-1} \omega^{ij} \mathbf{A}_i, \quad 0 \leq j \leq N-1,$$

where $\omega = e^{-\frac{2\pi i}{N}}$ and $\mathbf{F}_j(\mathbf{A})$ is the j -th component of the DFT of $\mathbf{A}$.

DEFINITION 3.2. Let $\mathbf{A} = \mathcal{C}(\mathbf{A}_0, \mathbf{A}_1, \dots, \mathbf{A}_{N-1})$ and $\mathbf{F}^{-1}(\mathbf{A}) = (\mathbf{F}_0^{-1}(\mathbf{A}), \mathbf{F}_1^{-1}(\mathbf{A}), \dots, \mathbf{F}_{N-1}^{-1}(\mathbf{A}))$.

$$\mathbf{F}^{-1}(\mathbf{A}) = (\mathbf{F}_0^{-1}(\mathbf{A}), \mathbf{F}_1^{-1}(\mathbf{A}), \dots, \mathbf{F}_{N-1}^{-1}(\mathbf{A})),$$

where $\mathbf{F}_j^{-1}(\mathbf{A}), 0 \leq j \leq N-1$,

$$\mathbf{F}_j^{-1}(\mathbf{A}) \triangleq \frac{1}{N} \sum_{i=0}^{N-1} \omega^{-ij} \mathbf{A}_i, \quad 0 \leq j \leq N-1.$$

In particular, we have

$$(3) \quad \mathbf{F}_0^{-1}(\mathbf{A}) = \frac{1}{N} \sum_{i=0}^{N-1} \mathbf{A}_i.$$

$\mathbf{F}^{-1}$ is indeed an inverse of $\mathbf{F}$ in the sense that for every $\mathbf{A} = \mathcal{C}(\mathbf{A}_0, \dots, \mathbf{A}_{N-1})$

$$(4) \quad \mathbf{F}^{-1}(\mathbf{F}(\mathbf{A})) = \mathbf{F}(\mathbf{F}^{-1}(\mathbf{A})) = \mathbf{A}.$$

The proof of this fact relies heavily on the useful identity

$$(5) \quad \sum_{j=0}^{N-1} \omega^{jp} = \begin{cases} 0, & p = 1, 2, \dots, N-1, \\ N, & p = 0. \end{cases}$$

The following properties of $\mathbf{F}_j$ are generalizations to the block circulant case of well-known properties of the DFT for vectors.

LEMMA 3.2. Let $\mathbf{A}, \mathbf{B} = \mathcal{C}(\mathbf{A}_0, \mathbf{A}_1, \dots, \mathbf{A}_{N-1})$, $\mathbf{C} = \mathcal{C}(\mathbf{C}_0, \mathbf{C}_1, \dots, \mathbf{C}_{N-1})$, $\mathbf{A}_j, \mathbf{C}_j \in \mathbb{F}^{k \times l}$, $\mathbf{B}_j \in \mathbb{F}^{l \times m}$, $0 \leq j \leq N-1$.

1. $(\mathbf{F}_j(\mathbf{A}))^* = \mathbf{F}_j(\mathbf{A}^*)$.
2. $\mathbf{F}_j(\mathbf{A} + \mathbf{C}) = \mathbf{F}_j(\mathbf{A}) + \mathbf{F}_j(\mathbf{C})$.
3. $\mathbf{F}_j(\mathbf{AB}) = \mathbf{F}_j(\mathbf{A})\mathbf{F}_j(\mathbf{B})$.

An important special case of block circulant matrices are block circulant matrices, which are matrices of the form

$$\mathcal{M}(\mathbf{A}_0, \mathbf{A}_1) \triangleq \mathcal{C}(\mathbf{A}_0, \mathbf{A}_1, \dots, \mathbf{A}_1) = \begin{pmatrix} \mathbf{A}_0 & \mathbf{A}_1 & \cdots & \mathbf{A}_1 \\ \mathbf{A}_1 & \mathbf{A}_0 & \cdots & \mathbf{A}_1 \\ \vdots & \vdots & & \vdots \\ \mathbf{A}_1 & \mathbf{A}_1 & \cdots & \mathbf{A}_0 \end{pmatrix}.$$

In this case there are also only two different DFT components:

$$\begin{aligned} \mathbf{F}_0(\mathcal{M}(\mathbf{A}_0, \mathbf{A}_1)) &= \mathbf{A}_0 + (N-1)\mathbf{A}_1, \\ \mathbf{F}_j(\mathcal{M}(\mathbf{A}_0, \mathbf{A}_1)) &= \mathbf{A}_0 - \mathbf{A}_1, \quad 1 \leq j \leq N-1. \end{aligned}$$

It is also easy to see that there are only two different inverse DFT components:

$$\begin{aligned} \mathbf{F}_0^{-1}(\mathcal{M}(\mathbf{A}_0, \mathbf{A}_1)) &= \frac{1}{N}(\mathbf{A}_0 + (N-1)\mathbf{A}_1), \\ \mathbf{F}_j^{-1}(\mathcal{M}(\mathbf{A}_0, \mathbf{A}_1)) &= \frac{1}{N}(\mathbf{A}_0 - \mathbf{A}_1), \quad 1 \leq j \leq N-1. \end{aligned}$$

3.2. The DFT of block vectors.

DEFINITION 3.3. Let $\mathbf{y}_0, \mathbf{y}_1, \dots, \mathbf{y}_{N-1} \in \mathbb{F}^p$. The DFT of the block vector $\mathbf{y} = (\mathbf{y}_0^T, \mathbf{y}_1^T, \dots, \mathbf{y}_{N-1}^T)^T \in \mathbb{F}^{Np}$ is $\mathbf{f}(\mathbf{y}) = (\mathbf{f}_0(\mathbf{y})^T, \mathbf{f}_1(\mathbf{y})^T, \dots, \mathbf{f}_{N-1}(\mathbf{y})^T)^T$.

$$\mathbf{f}_j(\mathbf{y}) \triangleq \sum_{i=0}^{N-1} \omega^{-ij} \mathbf{y}_i.$$

Notice that the definition of the DFT for block vectors is slightly different from the definition of the DFT for block circulant matrices (ω^{-ij} instead of ω^{ij}). Although the difference seems negligible, it is crucial to define the DFT for block vectors in that manner; otherwise, some critical properties will be lost (cf. Lemma 3.4).

The inverse DFT of $\mathbf{f}$ is defined by

$$\mathbf{f}_j^{-1}(\mathbf{y}) = \frac{1}{N} \sum_{i=0}^{N-1} \omega^{ij} \mathbf{y}_i.$$

For every $0 \leq j \leq N-1$,

$$\mathbf{f}_j^{-1}(\mathbf{f}(\mathbf{y})) = \mathbf{f}_j(\mathbf{f}_j^{-1}(\mathbf{y})) = \mathbf{y}_j.$$

The following properties of the DFT and the inverse DFT will be useful later on.

LEMMA 3.3. $\mathbf{y} = (\mathbf{y}_0^T, \mathbf{y}_1^T, \dots, \mathbf{y}_{N-1}^T)^T$, $\mathbf{z} = (\mathbf{z}_0^T, \mathbf{z}_1^T, \dots, \mathbf{z}_{N-1}^T)^T$, $\mathbf{y}_0, \dots, \mathbf{y}_{N-1} \in \mathbb{F}^m$, $\mathbf{z}_0, \dots, \mathbf{z}_{N-1} \in \mathbb{F}^m$.

1. (Norm preservation): $\|\mathbf{f}(\mathbf{y})\|^2 = N\|\mathbf{y}\|^2$
2. (Linearity): $\mathbf{f}(\alpha\mathbf{v} + \beta\mathbf{z}) = \alpha\mathbf{f}(\mathbf{v}) + \beta\mathbf{f}(\mathbf{z})$

PROOF. 1.

$$\begin{aligned} \|\mathbf{f}(\mathbf{y})\|^2 &= \sum_{j=0}^{N-1} \|\mathbf{f}_j(\mathbf{y})\|^2 = \sum_{j=0}^{N-1} \left\| \sum_{i=0}^{N-1} \omega^{-ij} \mathbf{y}_i \right\|^2 = \sum_{j=0}^{N-1} \left(\sum_{k=0}^{N-1} \omega^{kj} \mathbf{y}_k^T \right) \left(\sum_{i=0}^{N-1} \omega^{-ij} \mathbf{y}_i \right) \\ &= \sum_{j=0}^{N-1} \sum_{i=0}^{N-1} \sum_{k=0}^{N-1} \omega^{j(k-i)} \mathbf{y}_k^T \mathbf{y}_i \\ &= \sum_{i=0}^{N-1} \sum_{k=0}^{N-1} \left(\sum_{j=0}^{N-1} \omega^{j(k-i)} \right) \mathbf{y}_k^T \mathbf{y}_i \stackrel{(5)}{=} \sum_{i=0}^{N-1} N \|\mathbf{y}_i\|^2 = N\|\mathbf{y}\|^2. \end{aligned}$$

2. It directly follows from the definition of the DFT for block vectors. $\square$

Lemma 3.4 shows a connection between the DFT of block circulant matrices and the DFT of block vectors; this connection is one of the key ingredients in the analysis of the total least squares for block circulant systems.

LEMMA 3.4. $\mathbf{A} = \mathcal{C}(\mathbf{A}_0, \mathbf{A}_1, \dots, \mathbf{A}_{N-1})$, $\mathbf{y} = (\mathbf{y}_0^T, \mathbf{y}_1^T, \dots, \mathbf{y}_{N-1}^T)^T$, $\mathbf{y}_0, \dots, \mathbf{y}_{N-1} \in \mathbb{F}^m$, $\mathbf{A}_0, \mathbf{A}_1, \dots, \mathbf{A}_{N-1} \in \mathbb{F}^{m \times n}$, $\mathbf{f}_j(\mathbf{A}\mathbf{y}) = \mathbf{F}_j(\mathbf{A})\mathbf{f}_j(\mathbf{y})$, $0 \leq j \leq N-1$.

PROOF. For every $0 \leq j \leq N-1$,

$$\begin{aligned} \mathbf{f}_j(\mathbf{A}\mathbf{y}) &= \sum_{i=0}^{N-1} \omega^{-ij} (\mathbf{A}\mathbf{y})_i = \sum_{i=0}^{N-1} \omega^{-ij} \left(\sum_{k=0}^{N-1} \mathbf{A}_{[k-i]_N} \mathbf{y}_k \right) = \sum_{i=0}^{N-1} \sum_{k=0}^{N-1} \omega^{-ij} \mathbf{A}_{[k-i]_N} \mathbf{y}_k \\ &= \sum_{k=0}^{N-1} \sum_{i=0}^{N-1} \omega^{-ij} \mathbf{A}_{[k-i]_N} \mathbf{y}_k \\ &= \sum_{k=0}^{N-1} \omega^{-kj} \left(\sum_{i=0}^{N-1} \omega^{(k-i)j} \mathbf{A}_{[k-i]_N} \right) \mathbf{y}_k = \sum_{k=0}^{N-1} \omega^{-kj} \mathbf{F}_j(\mathbf{A}) \mathbf{y}_k \\ &= \mathbf{F}_j(\mathbf{A}) \sum_{k=0}^{N-1} \omega^{-kj} \mathbf{y}_k = \mathbf{F}_j(\mathbf{A}) \mathbf{f}_j(\mathbf{y}). \quad \square \end{aligned}$$

4. STLS in the case of block circulant matrices.

4.1. The algorithm. Suppose that $\mathbf{A}$ has a block circulant structure, i.e., $\mathbf{A} = \mathcal{C}(\mathbf{A}_0, \mathbf{A}_1, \dots, \mathbf{A}_{N-1})$, where $\mathbf{A}_i \in \mathbb{F}^{m \times n}$, and we wish to find a perturbation matrix $\Delta \mathbf{A}$, which also has a block circulant structure. We assume that the system is overdetermined, i.e., $m > n$. The STLS problem for block circulant matrices can be written as

$$(6) \quad \begin{aligned} & \min_{\Delta \mathbf{A}_0, \dots, \Delta \mathbf{A}_{N-1}, \Delta \mathbf{b}, \mathbf{x}} \quad \|\Delta \mathbf{A}\|^2 + \|\Delta \mathbf{b}\|^2 \\ & \text{subject to} \quad (\mathbf{A} - \Delta \mathbf{A})\mathbf{x} = \mathbf{b} - \Delta \mathbf{b}, \\ & \quad \quad \quad \Delta \mathbf{A} = \mathcal{C}(\Delta \mathbf{A}_0, \Delta \mathbf{A}_1, \dots, \Delta \mathbf{A}_{N-1}). \end{aligned}$$

In order to solve this problem, we will first apply the DFT on both sides of the consistency equation

$$(7) \quad (\mathbf{A} - \Delta \mathbf{A})\mathbf{x} = \mathbf{b} - \Delta \mathbf{b}.$$

It will be useful to treat $\mathbf{b} \in \mathbb{F}^{Nm}$ and $\Delta \mathbf{b} \in \mathbb{F}^{Nm}$ as block vectors, i.e., $(\mathbf{b}_0^T, \mathbf{b}_1^T, \dots, \mathbf{b}_{N-1}^T)^T$ and $\Delta \mathbf{b} = (\Delta \mathbf{b}_0^T, \dots, \Delta \mathbf{b}_{N-1}^T)^T$, where $\mathbf{b}_j, \Delta \mathbf{b}_j \in \mathbb{F}^m, 0 \leq j \leq N-1$. By applying the block vector DFT, $\mathbf{f}$, on both sides of (7) we obtain

$$\mathbf{f}((\mathbf{A} - \Delta \mathbf{A})\mathbf{x}) = \mathbf{f}(\mathbf{b} - \Delta \mathbf{b}).$$

Using property 2 of Lemma 3.3 we have

$$\mathbf{f}((\mathbf{A} - \Delta \mathbf{A})\mathbf{x}) = \mathbf{f}(\mathbf{b}) - \mathbf{f}(\Delta \mathbf{b}),$$

which is equivalent to the following system of N equations:

$$\mathbf{f}_j((\mathbf{A} - \Delta \mathbf{A})\mathbf{x}) = \mathbf{f}_j(\mathbf{b}) - \mathbf{f}_j(\Delta \mathbf{b}) \quad \forall 0 \leq j \leq N-1.$$

Finally, using Lemma 3.4 we have that (7) is equivalent to the following N “small” linear systems (in the unknown variables being $\mathbf{f}_0(\mathbf{x}), \dots, \mathbf{f}_{N-1}(\mathbf{x})$):

$$(\mathbf{F}_j(\mathbf{A}) - \mathbf{F}_j(\Delta \mathbf{A}))\mathbf{f}_j(\mathbf{x}) = \mathbf{f}_j(\mathbf{b}) - \mathbf{f}_j(\Delta \mathbf{b}) \quad \forall 0 \leq j \leq N-1.$$

The objective function $\|\Delta \mathbf{A}\|^2 + \|\Delta \mathbf{b}\|^2$ can also be expressed solely by its DFT components. Indeed, from Lemma 3.3 we have

$$(8) \quad \|\Delta \mathbf{b}\|^2 = \frac{1}{N} \|\mathbf{f}(\Delta \mathbf{b})\|^2 = \frac{1}{N} \left(\sum_{j=0}^{N-1} \|\mathbf{f}_j(\Delta \mathbf{b})\|^2 \right).$$

Also, by the definition of Frobenius norm we have that $\|\Delta \mathbf{A}\|^2 = \text{Tr}(\Delta \mathbf{A}^* \Delta \mathbf{A})$. Since $\Delta \mathbf{A}$ is block circulant, then by Lemma 3.1 we have that $\Delta \mathbf{A}^* \Delta \mathbf{A}$ is also block circulant, and thus we can write

$$\Delta \mathbf{A}^* \Delta \mathbf{A} = \mathcal{C}(\mathbf{R}_0, \mathbf{R}_1, \dots, \mathbf{R}_{N-1}) = \begin{pmatrix} \mathbf{R}_0 & \mathbf{R}_1 & \cdots & \mathbf{R}_{N-1} \\ \mathbf{R}_{N-1} & \mathbf{R}_0 & \cdots & \mathbf{R}_{N-2} \\ \vdots & \vdots & & \vdots \\ \mathbf{R}_1 & \mathbf{R}_2 & \cdots & \mathbf{R}_0 \end{pmatrix}$$

for some $\mathbf{R}_0, \mathbf{R}_1, \dots, \mathbf{R}_{N-1} \in \mathbb{F}^{n \times n}$, and therefore

$$\|\Delta \mathbf{A}\|^2 = \text{Tr}(\Delta \mathbf{A}^* \Delta \mathbf{A}) = N \text{Tr}(\mathbf{R}_0).$$

Moreover,

$$\begin{aligned}
 \|\Delta \mathbf{A}\|^2 &= N \text{Tr}(\mathbf{R}_0) \stackrel{(4)}{=} N \text{Tr}(\mathbf{F}_0^{-1}(\mathbf{F}(\Delta \mathbf{A}^* \Delta \mathbf{A}))) \\
 &= \sum_{j=0}^{N-1} \text{Tr}(\mathbf{F}_j(\Delta \mathbf{A}^* \Delta \mathbf{A})) \quad (\text{by (3)}) \\
 (9) \quad &= \sum_{j=0}^{N-1} \text{Tr}(\mathbf{F}_j(\Delta \mathbf{A})^* \mathbf{F}_j(\Delta \mathbf{A})) = \sum_{j=0}^{N-1} \|\mathbf{F}_j(\Delta \mathbf{A})\|^2 \quad (\text{by Lemma 3.1, property 2}).
 \end{aligned}$$

We thus obtained that (6) is reduced to

$$\begin{aligned}
 \min_{\Delta \mathbf{A}_0, \dots, \Delta \mathbf{A}_{N-1}, \Delta \mathbf{b}, \mathbf{x}} \quad & \sum_{j=0}^{N-1} (\|\mathbf{F}_j(\Delta \mathbf{A})\|^2 + \frac{1}{N} \|\mathbf{f}_j(\Delta \mathbf{b})\|^2) \\
 \text{subject to} \quad & (\mathbf{F}_j(\mathbf{A}) - \mathbf{F}_j(\Delta \mathbf{A})) \mathbf{f}_j(\mathbf{x}) = \mathbf{f}_j(\mathbf{b}) - \mathbf{f}_j(\Delta \mathbf{b}), \quad 0 \leq j \leq N-1, \\
 & \Delta \mathbf{A} = \mathcal{C}(\Delta \mathbf{A}_0, \Delta \mathbf{A}_1, \dots, \Delta \mathbf{A}_{N-1}).
 \end{aligned}
 \tag{10}$$

Making the change of variables

$$\begin{aligned}
 \mathbf{G}_j &= \mathbf{F}_j(\Delta \mathbf{A}) = \sum_{i=0}^{N-1} \omega^{ij} \Delta \mathbf{A}_i, \quad 0 \leq j \leq N-1, \\
 \mathbf{c}_j &= \mathbf{f}_j(\Delta \mathbf{b}) = \sum_{i=0}^{N-1} \omega^{-ij} \Delta \mathbf{b}_i, \quad 0 \leq j \leq N-1, \\
 \mathbf{z}_j &= \mathbf{f}_j(\mathbf{x}) = \sum_{i=0}^{N-1} \omega^{-ij} \mathbf{x}_i, \quad 0 \leq j \leq N-1,
 \end{aligned}$$

we obtain the following equivalent minimization problem:

$$\begin{aligned}
 \min_{\mathbf{G}_0, \dots, \mathbf{G}_{N-1}, \mathbf{c}_0, \dots, \mathbf{c}_{N-1}, \mathbf{z}_0, \dots, \mathbf{z}_{N-1}} \quad & \sum_{j=0}^{N-1} (\|\mathbf{G}_j\|^2 + \frac{1}{N} \|\mathbf{c}_j\|^2) \\
 \text{subject to} \quad & (\mathbf{F}_j(\mathbf{A}) - \mathbf{G}_j) \mathbf{z}_j = \mathbf{f}_j(\mathbf{b}) - \mathbf{c}_j, \quad 0 \leq j \leq N-1.
 \end{aligned}
 \tag{11}$$

Since (10) is separable in the variables

$$(\mathbf{G}_0, \mathbf{c}_0, \mathbf{z}_0), (\mathbf{G}_1, \mathbf{c}_1, \mathbf{z}_1), \dots, (\mathbf{G}_{N-1}, \mathbf{c}_{N-1}, \mathbf{z}_{N-1})$$

we actually need to solve N small TLS problems and then use the inverse DFT in order to find the values of $(\Delta \mathbf{A}, \Delta \mathbf{b}, \mathbf{x})$.

We summarize the above by presenting the block circulant TLS (BCTLS) algorithm for solving the STLS problem (6). The algorithm is essentially as simple as the classical SVD-based algorithm since its main effort consists of solving N small unstructured TLS problems.

ALGORITHM BCTLS.

Input: $\mathbf{A}, \mathbf{b}$, where $\mathbf{A} = \mathcal{C}(\mathbf{A}_0, \mathbf{A}_1, \dots, \mathbf{A}_{N-1}) \in \mathbb{F}^{Nm \times Nn}$ is a BC matrix and $\mathbf{b} = (\mathbf{b}_0^T, \mathbf{b}_1^T, \dots, \mathbf{b}_{N-1}^T)^T$ such that $\mathbf{A}_0, \mathbf{A}_1, \dots, \mathbf{A}_{N-1} \in \mathbb{F}^{m \times n}$ and $\mathbf{b}_0, \dots, \mathbf{b}_{N-1} \in \mathbb{F}^m$.
Output: $\Delta \mathbf{A}, \Delta \mathbf{b}, \mathbf{x}$, where $\Delta \mathbf{A} \in \mathbb{F}^{Nm \times Nn}$ is a block circulant matrix, $\Delta \mathbf{b} \in \mathbb{F}^{Nm}$, and $\mathbf{x} \in \mathbb{F}^{Nn}$ is the STLS solution.

1. Calculate the N DFT components of $\mathbf{A}$,

$$(12) \quad \mathbf{F}_j(\mathbf{A}) = \sum_{i=0}^{N-1} \omega^{ij} \mathbf{A}_i, \quad 0 \leq j \leq N-1,$$

and the N DFT components of $\mathbf{b}$,

$$\mathbf{f}_j(\mathbf{b}) = \sum_{i=0}^{N-1} \omega^{-ij} \mathbf{b}_i, \quad 0 \leq j \leq N-1.$$

2. For every $0 \leq j \leq N-1$, call the TLS algorithm with input $(\mathbf{F}_j(\mathbf{A}), \mathbf{f}_j(\mathbf{b}), \frac{1}{N})$ and obtain an output $(\mathbf{G}_j, \mathbf{c}_j, \mathbf{z}_j)$.
3. Denote $\mathbf{G} = \mathcal{C}(\mathbf{G}_0, \mathbf{G}_1, \dots, \mathbf{G}_{N-1})$, $\mathbf{c} = (\mathbf{c}_0^T, \mathbf{c}_1^T, \dots, \mathbf{c}_{N-1}^T)^T$ and $\mathbf{z} = (\mathbf{z}_0^T, \mathbf{z}_1^T, \dots, \mathbf{z}_{N-1}^T)^T$. The output of the BCTLS algorithm is computed by applying the inverse DFT of $\mathbf{G}$, $\mathbf{c}$, and $\mathbf{z}$. The obtained solution of the STLS problem (6) is then $\Delta \mathbf{A} = \mathcal{C}(\Delta \mathbf{A}_0, \Delta \mathbf{A}_1, \dots, \Delta \mathbf{A}_{N-1})$, $\Delta \mathbf{b} = (\Delta \mathbf{b}_0^T, \Delta \mathbf{b}_1^T, \dots, \Delta \mathbf{b}_{N-1}^T)^T$, and $\mathbf{x} = (\mathbf{x}_0^T, \mathbf{x}_1^T, \dots, \mathbf{x}_{N-1}^T)^T$, where

$$(13) \quad \Delta \mathbf{A}_j = \mathbf{F}_j^{-1}(\mathbf{G}) = \frac{1}{N} \sum_{i=0}^{N-1} \omega^{-ij} \mathbf{G}_i, \quad 0 \leq j \leq N-1,$$

$$\Delta \mathbf{b}_j = \mathbf{f}_j^{-1}(\mathbf{c}) = \frac{1}{N} \sum_{i=0}^{N-1} \omega^{ij} \mathbf{c}_i, \quad 0 \leq j \leq N-1,$$

$$(14) \quad \mathbf{x}_j = \mathbf{f}_j^{-1}(\mathbf{z}) = \frac{1}{N} \sum_{i=0}^{N-1} \omega^{ij} \mathbf{z}_i, \quad 0 \leq j \leq N-1.$$

Step 2 of Algorithm BCTLS requires N executions of algorithm TLS, and hence the following condition must be satisfied (see step 2 of algorithm TLS):

$$(15) \quad \sigma_n(\mathbf{F}_j(\mathbf{A})) > \sigma_{n+1} \left(\mathbf{F}_j(\mathbf{A}), \frac{1}{\sqrt{N}} \mathbf{f}_j(\mathbf{b}) \right) \quad \forall 0 \leq j \leq N-1.$$

We claim that condition (15) implies that the matrix $\mathbf{A}$ is full column rank, which is the same as $\mathbf{A}^* \mathbf{A}$ being nonsingular. Indeed, condition (15) implies in particular that $\sigma_n(\mathbf{F}_j(\mathbf{A})) > 0$ for every $0 \leq j \leq N-1$, which is the same as saying that $\mathbf{F}_j(\mathbf{A})^* \mathbf{F}_j(\mathbf{A})$ is nonsingular for every $0 \leq j \leq N-1$. By Lemma 3.2 we obtain that $\mathbf{F}_j(\mathbf{A}^* \mathbf{A})$ is nonsingular for every $0 \leq j \leq N-1$. By [13, Theorem 3.1], the eigenvalues of $\mathbf{A}^* \mathbf{A}$ are the Nn eigenvalues of the N matrices $\mathbf{F}_j(\mathbf{A}^* \mathbf{A})$. The latter matrices are nonsingular, and hence have only nonzero eigenvalues. Therefore, all the eigenvalues of $\mathbf{A}^* \mathbf{A}$ are different from zero, and, as a result, $\mathbf{A}^* \mathbf{A}$ is nonsingular. $\square$

From (14) it seems as if $\mathbf{x}$ is a complex vector even in the real case $\mathbb{F} = \mathbb{R}$. However, we claim that in fact it is a real vector. This is proved in the following theorem.

THEOREM 4.1. *Let $\mathbf{A} = \mathcal{C}(\mathbf{A}_0, \mathbf{A}_1, \dots, \mathbf{A}_{N-1})$, $\mathbf{b} = (\mathbf{b}_0^T, \mathbf{b}_1^T, \dots, \mathbf{b}_{N-1}^T)^T$, $\mathbf{A}_0, \dots, \mathbf{A}_{N-1} \in \mathbb{R}^{m \times n}$, $\mathbf{b}_0, \dots, \mathbf{b}_{N-1} \in \mathbb{R}^m$, $\Delta \mathbf{A} \in \mathbb{R}^{Nn \times Nn}$, $\Delta \mathbf{b} \in \mathbb{R}^{Nm}$, and $\mathbf{x} \in \mathbb{R}^{Nn}$. Throughout the proof we use the same notation used in the description of the BCTLS algorithm. The componentwise complex conjugate of a vector or a matrix is denoted by $(\cdot)$. The proof is based on the following four claims:*

- (i) $\mathbf{F}_{N-j}(\mathbf{A}) = \overline{\mathbf{F}_j(\mathbf{A})}$ and $\mathbf{f}_{N-j}(\mathbf{b}) = \overline{\mathbf{f}_j(\mathbf{b})}$ for every $0 \leq j \leq N-1$.
- (ii) $\mathbf{G}_{N-j} = \overline{\mathbf{G}_j}$ and $\mathbf{c}_{N-j} = \overline{\mathbf{c}_j}$ for every $0 \leq j \leq N-1$.
- (iii) $\Delta \mathbf{A}$ is a real-valued matrix, and $\Delta \mathbf{b}$ is a real-valued vector.
- (iv) $\mathbf{x}$ is a real valued vector.

(i). For every $0 \leq j \leq N-1$,

$$\mathbf{F}_{N-j}(\mathbf{A}) = \sum_{i=0}^{N-1} \omega^{i(N-j)} \mathbf{A}_i = \sum_{i=0}^{N-1} \omega^{-ij} \mathbf{A}_i = \sum_{i=0}^{N-1} \overline{\omega^{ij}} \mathbf{A}_i = \sum_{i=0}^{N-1} \overline{\omega^{ij} \mathbf{A}_i} = \overline{\mathbf{F}_j(\mathbf{A})}.$$

The proof that $\mathbf{f}_{N-j}(\mathbf{b}) = \overline{\mathbf{f}_j(\mathbf{b})}$ is almost identical.

(ii). For every $0 \leq j \leq N-1$, let the SVD of $(\mathbf{F}_j(\mathbf{A}), \frac{1}{N}\mathbf{f}_j(\mathbf{b}))$ be given by

$$(16) \quad \left(\mathbf{F}_j(\mathbf{A}), \frac{1}{N}\mathbf{f}_j(\mathbf{b}) \right) = \mathbf{U}^j \Sigma^j (\mathbf{V}^j)^*.$$

Taking the complex conjugate of both sides of (16) and using fact (i), we obtain that the SVD of $(\mathbf{F}_{N-j}(\mathbf{A}), \frac{1}{N}\mathbf{f}_{N-j}(\mathbf{b}))$ is given by

$$(17) \quad \left(\mathbf{F}_{N-j}(\mathbf{A}), \frac{1}{N}\mathbf{f}_{N-j}(\mathbf{b}) \right) = \overline{\mathbf{U}^j} \Sigma^j (\overline{\mathbf{V}^j})^*.$$

Since $(\mathbf{G}_j, \mathbf{c}_j, \mathbf{z}_j)$ is the output of the TLS algorithm with input $(\mathbf{F}_j(\mathbf{A}), \mathbf{f}_j(\mathbf{b}), \frac{1}{N})$, we have that

$$(18) \quad (\mathbf{G}_j, \mathbf{c}_j) = \sigma_{n+1} \mathbf{u}_{n+1}^j (\mathbf{v}_{n+1}^j)^T \mathbf{T}, \quad (\mathbf{G}_{N-j}, \mathbf{c}_{N-j}) = \sigma_{n+1} \mathbf{u}_{n+1}^{N-j} (\mathbf{v}_{n+1}^{N-j})^T \mathbf{T},$$

where

$$\mathbf{T} = (\underbrace{\text{diag } 1, 1, \dots, 1}_{n \text{ times}}, \sqrt{1/N}).$$

$\mathbf{u}_{n+1}^j$ and $\mathbf{v}_{n+1}^j$ are the $(n+1)$ th columns of the matrices $\mathbf{U}^j$ and $\mathbf{V}^j$ respectively. $\mathbf{u}_{n+1}^{N-j}$ and $\mathbf{v}_{n+1}^{N-j}$ are the $(n+1)$ th columns of the matrices $\mathbf{U}^{N-j}$ and $\mathbf{V}^{N-j}$, respectively. From (16) and (17) it follows that

$$(19) \quad \mathbf{u}_{n+1}^{N-j} = \overline{\mathbf{u}_{n+1}^j}, \quad \mathbf{v}_{n+1}^{N-j} = \overline{\mathbf{v}_{n+1}^j}.$$

Substituting (19) into (18) we deduce that $\mathbf{G}_{N-j} = \overline{\mathbf{G}_j}$ and $\mathbf{c}_{N-j} = \overline{\mathbf{c}_j}$.

(iii). First notice that $(\mathbf{G}_0, \mathbf{c}_0, \mathbf{z}_0)$ is the output of the TLS algorithm with input $(\mathbf{F}_0(\mathbf{A}_0), \mathbf{f}_0(\mathbf{b}), \frac{1}{N})$, and therefore $\mathbf{G}_0$ and $\mathbf{c}_0$ are real-valued. If N is odd, then $\Delta \mathbf{A}_j$ can be written as

$$\begin{aligned} \Delta \mathbf{A}_j &= \frac{1}{N} \sum_{i=0}^{N-1} \omega^{-ij} \mathbf{G}_i \\ &= \frac{1}{N} \mathbf{G}_0 + \frac{1}{N} \sum_{i=1}^{\frac{N-1}{2}} (\omega^{-ij} \mathbf{G}_i + \omega^{-(N-i)j} \mathbf{G}_{N-i}) \\ &\stackrel{\text{fact (ii)}}{=} \frac{1}{N} \mathbf{G}_0 + \frac{1}{N} \sum_{i=1}^{\frac{N-1}{2}} (\omega^{-ij} \mathbf{G}_i + \overline{\omega^{-ij} \mathbf{G}_i}), \end{aligned}$$

which is a real-valued matrix. The proof for the case where N is even and for the vector $\Delta \mathbf{b}$ are almost identical.

(iv). Since $\Delta \mathbf{A}$ is a real-valued matrix and $\Delta \mathbf{b}$ is a real-valued matrix and $\mathbf{x}$ is the unique solution of the real system $(\mathbf{A} - \Delta \mathbf{A})\mathbf{x} = \mathbf{b} - \Delta \mathbf{b}$, we conclude that $\mathbf{x}$ must be real-valued. $\square$

There are two kinds of calculations performed by Algorithm BCTLS:

1. Steps 1 and 3 require the calculation of the DFT of a block circulant matrix and a block vector (step 1) and the IDFT of a block circulant matrix and a block vector (step 3). The calculation of the DFT or the IDFT of a block circulant matrix given in (12) and (13) both require $O(N^2 mn)$ operations. However, this can be reduced to $O(N \log(N) mn)$ if one uses fast Fourier transform (FFT) [2]. Indeed, the computation of the sequence $\mathbf{F}_0(\mathbf{A}), \dots, \mathbf{F}_{N-1}(\mathbf{A})$ involves mn DFT calculations of sequences of scalars, one for each component of $\mathbf{A}_0, \dots, \mathbf{A}_{N-1}$. Each of the mn DFT calculations can be done in $O(N \log(N))$ operations by the classical FFT. Similarly, the computational effort required in order to calculate the DFT or the IDFT of a block vector is $O(N \log(N)m)$. Therefore, the overall computational effort of the DFT computations is $O(N \log(N) mn)$.
2. The algorithm requires N SVD calculations in step 2. The SVD of an $m \times n$ matrix can be calculated in $O(m^2 n)$ operations (see, e.g., [6]). Thus, step 2 requires $O(N m^2 n)$ operations.

As a consequence, the overall computational complexity of the algorithm is $O(N m^2 n + N \log(N) mn)$ operations. Note that the computational effort of the unstructured TLS problem with a matrix of the same size $Nm \times Nn$ is $O(N^3 m^2 n)$ operations, which, for large N , is significantly higher.

4.2. Illustration of Algorithm BCTLS. In this section we illustrate the BCTLS algorithm through an example with $N = 3, m = 3$, and $n = 2$. Suppose we have a “correct” linear system

$$\mathbf{A}^{(c)} \mathbf{x}^{(c)} = \mathbf{b}^{(c)},$$

where the “correct” solution is $\mathbf{x}^{(c)} = (1, 1, 1, 1, 1, 1)^T$, $\mathbf{A}^{(c)}$ is the 9×6 matrix defined by $\mathbf{A}^{(c)} = \mathcal{C}(\mathbf{A}_0^{(c)}, \mathbf{A}_1^{(c)}, \mathbf{A}_2^{(c)})$, where

$$\mathbf{A}_0^{(c)} = \begin{pmatrix} 1 & 1 \\ 1 & 1 \\ 1 & 0 \end{pmatrix}, \quad \mathbf{A}_1^{(c)} = \begin{pmatrix} 1 & 1 \\ 0 & 0 \\ 1 & 0 \end{pmatrix}, \quad \mathbf{A}_2^{(c)} = \begin{pmatrix} 1 & 1 \\ 1 & 0 \\ 0 & 1 \end{pmatrix},$$

and $\mathbf{b}^{(c)}$ is the “correct” observed vector, i.e., $\mathbf{b}^{(c)} = \mathbf{A}^{(c)} \mathbf{x} = (6, 3, 3, 6, 3, 3, 6, 3, 3)^T$. Assume now that each component of $\mathbf{A}_0^{(c)}, \mathbf{A}_1^{(c)}, \mathbf{A}_2^{(c)}$, and $\mathbf{b}^{(c)}$ is corrupted by a Gaussian additive noise with zero expectation and standard variation 0.3. In one realization of the noise, this gave rise to the “observed matrix” $\mathbf{A} = \mathcal{C}(\mathbf{A}_0, \mathbf{A}_1, \mathbf{A}_2)$,

$$\mathbf{A}_0 = \begin{pmatrix} 1.529 & 0.584 \\ 0.989 & 0.839 \\ 1.094 & -0.091 \end{pmatrix}, \quad \mathbf{A}_1 = \begin{pmatrix} 1.038 & 0.935 \\ 0.177 & -0.140 \\ 0.681 & -0.148 \end{pmatrix}, \quad \mathbf{A}_2 = \begin{pmatrix} 1.074 & 0.132 \\ 1.287 & 0.224 \\ 0.092 & 1.195 \end{pmatrix},$$

and the “observed vector,”

$$\mathbf{b} = (5.934, 2.925, 2.941, 5.656, 2.989, 3.043, 6.434, 3.114, 3.163)^T.$$

Ignoring the special structure of $\mathbf{A}$ and applying the (unstructured) TLS algorithm, the TLS solution is

$$\mathbf{x}_{TLS} = (0.6832, 1.0906, 0.8109, 1.3365, 0.9744, 1.1405)^T.$$

To solve the STLS problem we need to compute the three DFT components of $\mathbf{A}$ and $\mathbf{b}$:

$$\begin{aligned} \mathbf{F}_0(\mathbf{A}) &= \mathbf{A}_0 + \mathbf{A}_1 + \mathbf{A}_2 = \begin{pmatrix} 3.6424 & 2.6517 \\ 2.4546 & 0.9236 \\ 1.8682 & 0.9557 \end{pmatrix}, \\ \mathbf{F}_1(\mathbf{A}) &= \mathbf{A}_0 + \omega \mathbf{A}_1 + \omega^2 \mathbf{A}_2 = \begin{pmatrix} 0.4728 - 0.0307i & -0.4499 - 0.1702i \\ 0.2566 - 0.9609i & 0.7974 - 0.3166i \\ 0.7077 + 0.5107i & -0.6149 - 1.1646i \end{pmatrix}, \\ \mathbf{F}_2(\mathbf{A}) &= \mathbf{A}_0 + \omega^2 \mathbf{A}_1 + \omega \mathbf{A}_2 = \begin{pmatrix} 0.4728 + 0.0307i & -0.4499 + 0.1702i \\ 0.2566 + 0.9609i & 0.7974 + 0.3166i \\ 0.7077 - 0.5107i & -0.6149 + 1.1646i \end{pmatrix}, \\ \mathbf{f}_0(\mathbf{b}) &= \mathbf{b}_0 + \mathbf{b}_1 + \mathbf{b}_2 = (18.0260, 9.0289, 9.1477)^T, \\ \mathbf{f}_1(\mathbf{b}) &= \mathbf{b}_0 + \omega \mathbf{b}_1 + \omega^2 \mathbf{b}_2 \\ &= (-0.1106 + 0.6740i, -0.1266 + 0.1085i, -0.1621 + 0.1034i)^T, \\ \mathbf{f}_2(\mathbf{b}) &= \mathbf{b}_0 + \omega^2 \mathbf{b}_1 + \omega \mathbf{b}_2 \\ &= (-0.1106 - 0.6740i, -0.1266 - 0.1085i, -0.1621 - 0.1034i)^T. \end{aligned}$$

Applying the TLS algorithm on the three sets of inputs $(\mathbf{F}_j(\mathbf{A}), \mathbf{f}_j(\mathbf{b}), 1/3)$ we derive the three DFT components of $\mathbf{x}$:

$$\mathbf{f}_0(\mathbf{x}) = \begin{pmatrix} 2.5428 \\ 3.4395 \end{pmatrix}, \quad \mathbf{f}_1(\mathbf{x}) = \begin{pmatrix} -0.2096 + 0.1416i \\ -0.1480 - 0.1697i \end{pmatrix}, \quad \mathbf{f}_2(\mathbf{x}) = \begin{pmatrix} -0.2096 - 0.1416i \\ -0.1480 + 0.1697i \end{pmatrix},$$

and the solution of the STLS is given by

$$\mathbf{x}_{STLS} = \mathbf{x} = \mathbf{f}^{-1}(\mathbf{f}(\mathbf{x})) = \frac{1}{3} \begin{pmatrix} \mathbf{f}_0(\mathbf{x}) + \mathbf{f}_1(\mathbf{x}) + \mathbf{f}_2(\mathbf{x}) \\ \mathbf{f}_0(\mathbf{x}) + \omega \mathbf{f}_1(\mathbf{x}) + \omega^2 \mathbf{f}_2(\mathbf{x}) \\ \mathbf{f}_0(\mathbf{x}) + \omega^2 \mathbf{f}_1(\mathbf{x}) + \omega \mathbf{f}_2(\mathbf{x}) \end{pmatrix} = \begin{pmatrix} 0.7079 \\ 1.0478 \\ 0.8357 \\ 1.2938 \\ 0.9993 \\ 1.0978 \end{pmatrix}.$$

It is interesting to note that in this particular case it so happens that

$$(20) \quad |(\mathbf{x}_{STLS})_i - \mathbf{x}_i^{(c)}| < |(\mathbf{x}_{TLS})_i - \mathbf{x}_i^{(c)}| \quad \forall i.$$

Notice also that, as claimed in Theorem 4.1, $\mathbf{x}_{STLS}$ is a real vector.

5. STLS in the case of elementary block circulant systems. In this section we assume that $\mathbf{A}$ is an elementary block circulant matrix, i.e., $\mathbf{A} = \mathcal{M}(\mathbf{A}_0, \mathbf{A}_1)$, $\mathbf{A}_0, \mathbf{A}_1 \in \mathcal{F}^{m \times n}$ ($m > n$) (see section 3), and we wish to find a perturbation matrix $\Delta \mathbf{A}$, which also has an elementary block circulant structure. In this case, the STLS problem becomes

$$(21) \quad \begin{aligned} &\min_{\Delta \mathbf{A}_0, \Delta \mathbf{A}_1, \Delta \mathbf{b}, \mathbf{x}} \quad \|\Delta \mathbf{A}\|^2 + \|\Delta \mathbf{b}\|^2 \\ &\text{subject to} \quad (\mathbf{A} - \Delta \mathbf{A})\mathbf{x} = \mathbf{b} - \Delta \mathbf{b}, \\ &\quad \Delta \mathbf{A} = \mathcal{M}(\Delta \mathbf{A}_0, \Delta \mathbf{A}_1). \end{aligned}$$

Although an EBC matrix is a special case of a BC matrix, we cannot apply the BCTLS algorithm to solve (21) since an EBC matrix possesses additional special structure, $\mathbf{A}_j = \mathbf{A}_k \forall j \neq k (j, k \neq 0)$, which is not guaranteed to be produced by the BCTLS algorithm.

As in the case of the block circulant structure, we will apply the DFT on both sides of the consistency equation $(\mathbf{A} - \Delta\mathbf{A})\mathbf{x} = \mathbf{b} - \Delta\mathbf{b}$ and obtain that the consistency equation is equivalent to N “small” linear systems: the linear system

$$(\mathbf{F}_0(\mathbf{A}) - \mathbf{F}_0(\Delta\mathbf{A}))\mathbf{f}_0(\mathbf{x}) = \mathbf{f}_0(\mathbf{b}) - \mathbf{f}_0(\Delta\mathbf{b})$$

and the $N - 1$ linear systems

$$(\mathbf{F}_1(\mathbf{A}) - \mathbf{F}_1(\Delta\mathbf{A}))\mathbf{f}_j(\mathbf{x}) = \mathbf{f}_j(\mathbf{b}) - \mathbf{f}_j(\Delta\mathbf{b}), \quad j = 1, 2, \dots, N - 1.$$

From (9) and (8) we have that

$$\|\Delta\mathbf{b}\|^2 = \frac{1}{N} \left(\sum_{j=0}^{N-1} \|\mathbf{f}_j(\Delta\mathbf{b})\|^2 \right), \quad \|\Delta\mathbf{A}\|^2 = \|\mathbf{F}_0(\Delta\mathbf{A})\|^2 + (N - 1)\|\mathbf{F}_1(\Delta\mathbf{A})\|^2.$$

Thus we obtain that in the case of EBC structure, the STLS problem (21) is reduced to

$$\begin{aligned} \min_{\Delta\mathbf{A}_0, \Delta\mathbf{A}_1, \Delta\mathbf{b}} \quad & \|\Delta\mathbf{A}_0\|^2 + \frac{1}{N}\|\Delta\mathbf{b}_0\|^2 + (N - 1)\|\Delta\mathbf{A}_1\|^2 + \frac{1}{N} \sum_{i=1}^{N-1} \|\Delta\mathbf{b}_i\|^2 \\ \text{subject to} \quad & (\mathbf{F}_0(\mathbf{A}) - \mathbf{F}_0(\Delta\mathbf{A}))\mathbf{f}_0(\mathbf{x}) = \mathbf{f}_0(\mathbf{b}) - \mathbf{f}_0(\Delta\mathbf{b}) \\ & (\mathbf{F}_1(\mathbf{A}) - \mathbf{F}_1(\Delta\mathbf{A}))\mathbf{f}_j(\mathbf{x}) = \mathbf{f}_j(\mathbf{b}) - \mathbf{f}_j(\Delta\mathbf{b}), \quad 1 \leq j \leq N - 1, \\ & \Delta\mathbf{A} = \mathcal{M}(\Delta\mathbf{A}_0, \Delta\mathbf{A}_1). \end{aligned}$$

Making the change of variables

$$\begin{aligned} \mathbf{G}_0 &= \mathbf{F}_0(\Delta\mathbf{A}) = \Delta\mathbf{A}_0 + (N - 1)\Delta\mathbf{A}_1, \\ \mathbf{G}_1 &= \mathbf{F}_1(\Delta\mathbf{A}) = \Delta\mathbf{A}_0 - \Delta\mathbf{A}_1, \\ \mathbf{c}_j &= \mathbf{f}_j(\Delta\mathbf{b}) = \sum_{i=0}^{N-1} \omega^{-ij} \Delta\mathbf{b}_i, \quad 0 \leq j \leq N - 1, \\ \mathbf{z}_j &= \mathbf{f}_j(\mathbf{x}) = \sum_{i=0}^{N-1} \omega^{-ij} \mathbf{x}_i, \quad 0 \leq j \leq N - 1, \end{aligned}$$

we obtain the following equivalent minimization problem:

$$\begin{aligned} \min_{\mathbf{G}_0, \mathbf{G}_1, \mathbf{c}_0, \dots, \mathbf{c}_{N-1}, \mathbf{z}_0, \dots, \mathbf{z}_{N-1}} \quad & \|\mathbf{G}_0\|^2 + \frac{1}{N}\|\mathbf{c}_0\|^2 + (N - 1)\|\mathbf{G}_1\|^2 + \frac{1}{N} \sum_{i=1}^{N-1} \|\mathbf{c}_i\|^2 \\ \text{subject to} \quad & (\mathbf{F}_0(\mathbf{A}) - \mathbf{G}_0)\mathbf{z}_0 = \mathbf{f}_0(\mathbf{b}) - \mathbf{c}_0 \\ & (\mathbf{F}_1(\mathbf{A}) - \mathbf{G}_1)\mathbf{z}_j = \mathbf{f}_j(\mathbf{b}) - \mathbf{c}_j, \quad 1 \leq j \leq N - 1, \end{aligned} \tag{22}$$

which is separable with respect to the groups of variables $(\mathbf{G}_0, \mathbf{c}_0, \mathbf{z}_0)$ and the variables set

$$\{(\mathbf{G}_1, \mathbf{c}_1, \mathbf{z}_1), \dots, (\mathbf{G}_{N-1}, \mathbf{c}_{N-1}, \mathbf{z}_{N-1})\}.$$

Therefore, the solution of the minimization problem (22) is the sum of the two minimization problems

$$\begin{aligned} \min_{\mathbf{G}_0, \mathbf{c}_0, \mathbf{z}_0} \quad & \|\mathbf{G}_0\|^2 + \frac{1}{N}\|\mathbf{c}_0\|^2 \\ \text{subject to} \quad & (\mathbf{F}_0(\mathbf{A}) - \mathbf{G}_0)\mathbf{z}_0 = \mathbf{f}_0(\mathbf{b}) - \mathbf{c}_0 \end{aligned} \tag{23}$$

and

$$(24) \quad \begin{aligned} & \min_{\mathbf{G}_1, \mathbf{c}_1, \dots, \mathbf{c}_{N-1}, \mathbf{z}_1, \dots, \mathbf{z}_{N-1}} \quad (N-1)\|\mathbf{G}_1\|^2 + \frac{1}{N} \sum_{i=1}^{N-1} \|\mathbf{c}_i\|^2 \\ & \text{subject to} \quad (\mathbf{F}_1(\mathbf{A}) - \mathbf{G}_1)\mathbf{z}_j = \mathbf{f}_j(\mathbf{b}) - \mathbf{c}_j, \quad 1 \leq j \leq N-1. \end{aligned}$$

The minimization problem (23) is a TLS problem, and the second problem (24) is an MTLs problem. This gives rise to the following algorithm for solving the STLS problem for EBC matrices.

ALGORITHM EBCTLS FOR ELEMENTARY BLOCK CIRCULANT MATRICES.

Input: $\mathbf{A}, \mathbf{b}$, where $\mathbf{A} = \mathcal{M}(\mathbf{A}_0, \mathbf{A}_1) \in \mathbb{F}^{Nm \times Nm}$ is an EBC matrix and $\mathbf{b} = (\mathbf{b}_0^T, \mathbf{b}_1^T, \dots, \mathbf{b}_{N-1}^T)^T$ such that $\mathbf{A}_0, \mathbf{A}_1 \in \mathbb{F}^{m \times n}$ and $\mathbf{b}_0, \dots, \mathbf{b}_{N-1} \in \mathbb{F}^m$.

Output: $\Delta\mathbf{A}, \Delta\mathbf{b}, \mathbf{x}$, where $\Delta\mathbf{A} \in \mathbb{F}^{Nm \times Nm}$ is an elementary block circulant matrix, $\Delta\mathbf{b} \in \mathbb{F}^{Nm}$, and $\mathbf{x} \in \mathbb{F}^{Nn}$ is the STLS solution.

1. Calculate the two different DFT components of $\mathbf{A}$,

$$\mathbf{F}_0(\mathbf{A}) = \mathbf{A}_0 + (N-1)\mathbf{A}_1, \quad \mathbf{F}_1(\mathbf{A}) = \mathbf{A}_0 - \mathbf{A}_1,$$

and the N DFT components of $\mathbf{b}$,

$$\mathbf{f}_j(\mathbf{b}) = \sum_{i=0}^{N-1} \omega^{-ij} \mathbf{b}_i, \quad 0 \leq j \leq N-1.$$

2. Call the TLS algorithm with input $(\mathbf{F}_0(\mathbf{A}), \mathbf{f}_0(\mathbf{b}), \frac{1}{N})$ and obtain an output $(\mathbf{G}_0, \mathbf{c}_0, \mathbf{z}_0)$.
3. Call the MTLs algorithm with input $(\mathbf{F}_1(\mathbf{A}), \mathbf{f}_1(\mathbf{b}), \dots, \mathbf{f}_{N-1}(\mathbf{b}), \frac{1}{N(N-1)})$ and obtain an output $\mathbf{G}_1, \mathbf{c}_1, \mathbf{c}_2, \dots, \mathbf{c}_{N-1}$ and $\mathbf{z}_1, \dots, \mathbf{z}_{N-1}$.
4. Denote $\mathbf{G} = \mathcal{M}(\mathbf{G}_0, \mathbf{G}_1)$, $\mathbf{c} = (\mathbf{c}_0^T, \mathbf{c}_1^T, \dots, \mathbf{c}_{N-1}^T)^T$ and $\mathbf{z} = (\mathbf{z}_0^T, \mathbf{z}_1^T, \dots, \mathbf{z}_{N-1}^T)^T$. The output of the EBCTLS algorithm is computed by applying the inverse DFT of $\mathbf{G}, \mathbf{c}$, and $\mathbf{z}$. The optimal solution of the STLS problem (21) is then $\Delta\mathbf{A} = \mathcal{M}(\Delta\mathbf{A}_0, \Delta\mathbf{A}_1)$, $\Delta\mathbf{b} = (\Delta\mathbf{b}_0^T, \Delta\mathbf{b}_1^T, \dots, \Delta\mathbf{b}_{N-1}^T)^T$, and $\mathbf{x} = (\mathbf{x}_0^T, \mathbf{x}_1^T, \dots, \mathbf{x}_{N-1}^T)^T$, where

$$\Delta\mathbf{A}_0 = \mathbf{F}_0^{-1}(\mathbf{G}) = \frac{1}{N}(\mathbf{G}_0 + (N-1)\mathbf{G}_1),$$

$$\Delta\mathbf{A}_1 = \mathbf{F}_1^{-1}(\mathbf{G}) = \frac{1}{N}(\mathbf{G}_0 - \mathbf{G}_1),$$

$$\Delta\mathbf{b}_j = \mathbf{f}_j^{-1}(\mathbf{c}) = \frac{1}{N} \sum_{i=0}^{N-1} \omega^{ij} \mathbf{c}_i, \quad 0 \leq j \leq N-1,$$

$$\mathbf{x}_j = \mathbf{f}_j^{-1}(\mathbf{z}) = \frac{1}{N} \sum_{i=0}^{N-1} \omega^{ij} \mathbf{z}_i, \quad 0 \leq j \leq N-1.$$

1. Steps 2 and 3 of Algorithm BCTLS require the following conditions to be satisfied (see step 2 of the TLS and MTLs algorithms):

$$\sigma_n(\mathbf{F}_0(\mathbf{A})) > \sigma_{n+1} \left(\mathbf{F}_0(\mathbf{A}), \frac{1}{\sqrt{N}} \mathbf{f}_0(\mathbf{b}) \right),$$

$$\sigma_n(\mathbf{F}_1(\mathbf{A})) > \sigma_{n+1} \left(\mathbf{F}_1(\mathbf{A}), \frac{1}{\sqrt{N(N-1)}} \mathbf{f}_1(\mathbf{b}), \dots, \frac{1}{\sqrt{N(N-1)}} \mathbf{f}_{N-1}(\mathbf{b}) \right).$$

2. In the case $\mathbb{F} = \mathbb{R}$, the EBCTLS algorithm generates a real solution $\Delta\mathbf{A}, \Delta\mathbf{b}$, and $\mathbf{x}$. The proof is almost identical to the proof of Theorem 4.1.

6. Computational results. In this section we compare the SVD-based algorithms, BCTLS and EBCTLS, which find the global optimum of the STLS problem with BC and EBC matrices, respectively, to the following three methods.

1. **Least Squares (LS).** Here, we enforce $\Delta \mathbf{A}$ to be zero, and we choose a minimal norm $\Delta \mathbf{b}$. This is of course a very naive algorithm, and it assumes that the nominal value of the matrix is the true value. If the matrix $\mathbf{A}$ has full column rank, then the LS solution is given by $\mathbf{x}_{LS} = (\mathbf{A}^T \mathbf{A})^{-1} \mathbf{A}^T \mathbf{b}$ and $\Delta \mathbf{b} = \mathbf{b} - \mathbf{A} \mathbf{x}_{LS}$ (see, e.g., [1]).
2. **(unstructured) TLS method** disregards the linear structure of the matrix $\mathbf{A}$ and seeks a perturbation matrix $\Delta \mathbf{A}$ and a perturbation vector $\Delta \mathbf{b}$ that minimize $\|\Delta \mathbf{A}\|^2 + \|\Delta \mathbf{b}\|^2$ subject to the consistency equation $\mathbf{b} - \Delta \mathbf{b} \in \text{Range}(\mathbf{A} - \Delta \mathbf{A})$.
3. **STLN (STLS with 2-norm).** This method was introduced and studied in [7]. The STLN method (with 2-norm) is an iterative method for solving STLS problems with arbitrary linear structure. In each iteration of the STLN algorithm, a least squares problem is solved. In our problem, the size of the matrix in the least squares problem is $(Nm + 2mn) \times (Nn + 2mn)$. If $N \ll m, n$, then the complexity of the STLN method is $O(m^3 n^3)$, which is computationally very demanding. The algorithm is essentially a Newton-like method applied to a nonconvex function. There is no theoretical proof of convergence and, even when convergence occurs, there is no guarantee that it converges to a local optimum.

The first example considers a block circulant structure with $N = 2, m = 28, n = 4$. We assume that there is a “correct” system,

$$\mathbf{b}^{(c)} = \mathbf{A}^{(c)} \mathbf{x}^{(c)},$$

where $\mathbf{A}^{(c)}$ is given by

$$\mathbf{A}^{(c)} = \begin{pmatrix} \mathbf{A}_0^{(c)} & \mathbf{A}_1^{(c)} \\ \mathbf{A}_1^{(c)} & \mathbf{A}_0^{(c)} \end{pmatrix}$$

with $\mathbf{A}_0^{(c)}, \mathbf{A}_1^{(c)} \in \mathbb{R}^{28 \times 4}$. Each component of the matrices $\mathbf{A}_0^{(c)}$ and $\mathbf{A}_1^{(c)}$ was chosen to be -1 or 1 with probability $1/2$ independently of the other components. Each component of the correct vector $\mathbf{x}^{(c)}$ was a randomly chosen integer number between -10 and 9 . The actual data $\mathbf{A}, \mathbf{b}$ contains noise and is thus a perturbation of $\mathbf{A}^{(c)}$ and $\mathbf{b}^{(c)}$ (where $\mathbf{A}$ has the same structure as $\mathbf{A}^{(c)}$). In our experiments each component of $\mathbf{A}$ and $\mathbf{b}$ is the corresponding component of $\mathbf{A}^{(c)}$ and $\mathbf{b}^{(c)}$ plus a normal random variable with zero expectation and standard deviation equal to 0.2 . We considered three quantities to describe the results: the relative error of the matrix, the relative error of the solution, and the function value:

$$\begin{aligned} A_{err} &= \|\mathbf{A} - \Delta \mathbf{A} - \mathbf{A}^{(c)}\| / \|\mathbf{A}^{(c)}\|, \\ x_{err} &= \|\mathbf{x} - \mathbf{x}^{(c)}\| / \|\mathbf{x}^{(c)}\|, \\ value &= \|\Delta \mathbf{A}\|^2 + \|\Delta \mathbf{b}\|^2. \end{aligned}$$

The results given in the table below are the average over 200 realizations of the noise affecting the matrix $\mathbf{A}^{(c)}$ and the vector $\mathbf{b}^{(c)}$.

Method	Value
LS	16.4376
TLS	6.2414
STLN	2.7601
BCTLS	2.7601

In this example the STLN algorithm converged to the ℓ_1 optimum. The average number of iterations of the STLN algorithm was 14 and ranged between 11 and 17. It also can be seen that both the BCTLS and the STLN solutions were better than the LS solution in all aspects.

In our second example, we considered an EBC matrix with $N = 3$, $m = 16$, and $n = 4$. Hence, $\mathbf{A}^{(c)}$ is given by

$$\mathbf{A}^{(c)} = \begin{pmatrix} \mathbf{A}_0^{(c)} & \mathbf{A}_1^{(c)} & \mathbf{A}_1^{(c)} \\ \mathbf{A}_1^{(c)} & \mathbf{A}_0^{(c)} & \mathbf{A}_1^{(c)} \\ \mathbf{A}_1^{(c)} & \mathbf{A}_1^{(c)} & \mathbf{A}_0^{(c)} \end{pmatrix},$$

where $\mathbf{A}_0^{(c)}, \mathbf{A}_1^{(c)} \in \mathbb{R}^{16 \times 4}$ and the components of $\mathbf{A}_0^{(c)}$ and $\mathbf{A}_1^{(c)}$ were randomly chosen to be 0 or 1. Each component of the correct vector $\mathbf{x}^{(c)}$ was a randomly chosen integer number between -10 and 9 . The results in the table below are the average over 100 realizations of the noise

Method	Value
LS	523.9572
TLS	34.4647
STLN	12.4633
EBCTLS	6.1789

As can be seen from the above table, the STLN method in this case is suboptimal and does not converge to a global optimum. Moreover, in 17 out of the 100 instances the STLN algorithm did not converge at all, and in all 83 other cases it converged, after hundreds of iterations, but not to a global optimum.

REFERENCES

- [1] G. H. GOLUB AND C. F. VAN LOAN, *Matrix Computations*, 3rd ed., The Johns Hopkins University Press, Baltimore MD, 1996.
- [2] T. H. CORMEN, C. E. LEISERSON, AND R. L. RIVEST, *Introduction to Algorithms*, MIT Press, Cambridge, MA, McGraw-Hill, New York, 1990.
- [3] H. MURAKAMI, *Discrete wavelet transform based on cyclic convolution*, IEEE Trans. Signal Process., 52 (2004), pp. 165–174.
- [4] G. H. GOLUB AND C. F. VAN LOAN, *An analysis of the total least squares problem*, SIAM J. Numer. Anal., 17 (1980), pp. 883–893.
- [5] B. DE MOOR, *Structured total least squares and L_2 approximation problems*, Linear Algebra Appl., 188/189 (1993), pp. 163–207.
- [6] S. VAN HUFFEL AND J. VANDEWALLE, *The Total Least Squares Problem: Computational Aspects and Analysis*, Frontiers App. Math. 9, SIAM, Philadelphia, 1991.
- [7] J. B. ROSEN, H. PARK, AND J. GLICK, *Total least norm formulation and solution for structured problems*, SIAM J. Matrix Anal. Appl., 17 (1996), pp. 110–126.
- [8] B. DE MOOR, *Total least squares for affinely structured matrices and the noisy realization problem*, IEEE Trans. Signal Process., 42 (1994), pp. 3104–3113.
- [9] T. J. ABATZOGLOU, J. M. MENDEL, AND G. A. HARADA, *The constrained total least squares technique and its applications to harmonic suppresolution*, IEEE Trans. Signal Process., 39 (1991), pp. 1070–1087.
- [10] N. MASTRONARDI, P. LEMMERLING, AND S. VAN HUFFEL, *Fast structured total least squares algorithm for solving the basic deconvolution problem*, SIAM J. Matrix Anal. Appl., 22 (2000), pp. 533–553.

- [11] P. LEMMERLING AND S. VAN HUFFEL, *Analysis of the structured total least squares problem for Hankel/Toeplitz matrices*, Numer. Algorithms, 27 (2001), pp. 89–114.
- [12] S. VAN HUFFEL AND J. VANDEWALLE, *Analysis and properties of the generalized total least squares problem $AX \approx B$ when some or all columns in A are subject to error*, SIAM J. Matrix Anal. Appl., 10 (1989), pp. 294–315.
- [13] A. BECK, Y. C. ELDAR, AND A. BEN-TAL, *Minimax mean-squared error estimation of multi-channel signals*. IEEE Trans. Inform. Theory, submitted.
- [14] A. BECK, A. BEN-TAL, AND Y. C. ELDAR, *Robust mean-squared error estimation of multiple signals in linear systems affected by model and noise uncertainties*, Math. Program., to appear.
- [15] A. K. KATSAGGELOS, K. T. LAY, AND N. P. GALATSANOS, *A general framework for frequency domain multi-channel signal processing*, IEEE Trans. Image Process., 2 (1993), pp. 417–420.
- [16] M. SZNAIER, O. CAMPS, AND M. C. MAZZARO, *Finite horizon model reduction of a class of neutrally stable systems with applications to texture synthesis and recognition*, in Proceedings of the 43rd IEEE Conference on Decision and Control, Atlantis, Paradise, Bahamas, 2004.

COMPUTATIONALLY EFFICIENT APPROXIMATIONS OF THE JOINT SPECTRAL RADIUS*

VINCENT D. BLONDEL[†] AND YURII NESTEROV[‡]

Abstract. The joint spectral radius of a set of matrices is a measure of the maximal asymptotic growth rate that can be obtained by forming long products of matrices taken from the set. This quantity appears in a number of application contexts but is notoriously difficult to compute and to approximate. We introduce in this paper a procedure for approximating the joint spectral radius of a finite set of matrices with arbitrary high accuracy. Our approximation procedure is polynomial in the size of the matrices once the number of matrices and the desired accuracy are fixed.

For the special case of matrices with nonnegative entries we give elementary proofs of simple inequalities that we then use to obtain approximations of arbitrary high accuracy. From these inequalities it follows that the spectral radius of matrices with nonnegative entries is given by the simple expression

$$\rho(A_1, \dots, A_m) = \lim_{k \rightarrow \infty} \rho^{1/k}(A_1^{\otimes k} + \dots + A_m^{\otimes k}),$$

where it is somewhat surprising to notice that the right-hand side does not directly involve any mixed product between the matrices. ($A^{\otimes k}$ denotes the k th Kronecker power of A .)

For matrices with arbitrary entries (not necessarily nonnegative), we introduce an approximation procedure based on semidefinite liftings that can be implemented in a recursive way. For two matrices, even the first step of the procedure gives an approximation whose relative accuracy is at least $1/\sqrt{2}$, that is, more than 70%. The subsequent steps improve the accuracy but also increase the dimension of the auxiliary problems from which the approximation can be found.

Our approximation procedures provide approximations of relative accuracy $1 - \epsilon$ in time polynomial in $n^{(\ln m)/\epsilon}$, where m is the number of matrices and n is their size. These bounds are close from optimality since we show that, unless $P=NP$, no approximation algorithm is possible that provides a relative accuracy of $1 - \epsilon$ and runs in time polynomial in n and $1/\epsilon$.

As a by-product of our results we prove that a widely used approximation of the joint spectral radius based on common quadratic Lyapunov functions (or on ellipsoid norms) has relative accuracy $1/\sqrt{m}$, where m is the number of matrices.

Key words. joint spectral radius, generalized spectral radius, approximation algorithm, switched systems, linear inclusion, Lyapunov indicator, Kronecker power

AMS subject classifications. 15A18, 15-04, 15A60, 90C22, 93-04

DOI. 10.1137/040607009

1. Introduction. Let $\|\cdot\|$ be a matrix norm. The $\rho(A)$ of the real matrix A is defined by

$$(1.1) \quad \rho(A) = \lim_{k \rightarrow +\infty} \|A^k\|^{1/k}.$$

*Received by the editors April 19, 2004; accepted for publication (in revised form) by L. Vandenberghe December 23, 2004; published electronically August 31, 2005. This paper presents research results of the Belgian Program on Interuniversity Poles of Attraction initiated by the Belgian State, Prime Minister's Office, Science Policy Programming. The scientific responsibility is assumed by the authors. The research results presented in this paper have been supported in part by a grant "Action de recherche concertée ARC 04/09-315" from the "Direction de la recherche scientifique—Communauté française de Belgique."

<http://www.siam.org/journals/simax/27-1/60700.html>

[†]Division of Applied Mathematics, Université catholique de Louvain, 4 avenue Georges Lemaitre, B-1348 Louvain-la-Neuve, Belgium (blondel@inma.ucl.ac.be).

[‡]Center for Operations Research and Econometrics (CORE), Université catholique de Louvain (UCL), 34 voie du Roman Pays, 1348 Louvain-la-Neuve, Belgium (nesterov@core.ucl.ac.be).

The spectral radius of a matrix does not depend on the chosen matrix norm, and it is also equal to $\rho(A) = \max\{|\lambda| : \lambda \text{ is an eigenvalue of } A\}$ (see, for example, [22, Corollary 5.6.14]). The definition of spectral radius can be extended to $n \times n$ of matrices in a natural way. The $n \times n$ joint spectral radius of a set of matrices is a quantity introduced by Rota and Strang in the early 1960s that measures the maximal asymptotic growth rate that can be obtained by forming long products of matrices; see [29]. The general definition is for arbitrary sets of matrices, but we shall consider here only finite sets. Let $\{A_1, \dots, A_m\}$ be some set of real matrices. To the finite sequence $\sigma = (\sigma_1, \sigma_2, \dots, \sigma_k) \in \{1, \dots, m\}^k$ we associate the corresponding matrix product

$$A_\sigma = A_{\sigma_k} \cdots A_{\sigma_2} A_{\sigma_1}.$$

With this notation, the joint spectral radius is defined by

$$(1.2) \quad \rho(A_1, \dots, A_m) = \limsup_{k \rightarrow +\infty} \max_{\sigma \in \{1, \dots, m\}^k} \|A_\sigma\|^{1/k}.$$

As for the single-matrix case, the joint spectral radius does not depend on the matrix norm used. To see this, remember that any two matrix norms $\|\cdot\|_1$ and $\|\cdot\|_2$ are related by $\alpha\|A\|_1 \leq \|A\|_2 \leq \beta\|A\|_1$ for some $0 < \alpha < \beta$. For any product $\sigma \in \{1, \dots, m\}^k$, one has $\alpha^{1/k}\|A_\sigma\|_1^{1/k} \leq \|A_\sigma\|_2^{1/k} \leq \beta^{1/k}\|A_\sigma\|_1^{1/k}$, and by letting k tend to infinity we conclude that the joint spectral radius is well defined independently of the matrix norm used.

A definition analogous to (1.2) is possible by replacing the norm appearing in the definition by a spectral radius. The quantity defined in this way is the $n \times n$ joint spectral radius introduced in [15]. In [5], the joint and generalized spectral radii of finite (or bounded) sets of matrices are proved to be equal (see also [35, Theorem 1] for an elementary proof), thus reinforcing the status of the joint spectral radius as a legitimate generalization of the spectral radius of a single matrix. In what follows, we shall deal only with the spectral radius defined with a norm, as in (1.2).

Since its introduction in the 1960s the joint spectral radius has appeared in a number of different contexts; see, e.g., [31] or [16] for recent short surveys.¹ Let us illustrate one application in a dynamical system context. Consider the simple discrete-time linear inclusion

$$x_{k+1} \in \{A_1, \dots, A_m\} x_k, \quad x_0 \in \mathbf{R}^n,$$

in which at each step a particular linear transformation is chosen from a finite number of possible choices. The maximal asymptotic rate of growth of the trajectories associated with such a system is given by the joint spectral radius $\rho(A_1, \dots, A_m)$. (In the discrete linear inclusion literature, the logarithm of the joint spectral radius is sometimes called $\lambda(A_1, \dots, A_m)$; see, for example, [2].) In particular, all possible trajectories will converge to the origin if and only if $\rho(A_1, \dots, A_m) < 1$. The joint spectral radius can thus be associated with the stability properties of time-varying linear systems in the worst case over all possible time variations. It also occurs in the context of “asynchronous” [33] or “desynchronized” [23] systems. Besides systems-theoretic interpretations, the concept is pervasive in many areas of applied mathematics such as in wavelets [15], iterated function systems, random walks, fractals, numerical solutions to ordinary differential equations [18], discrete-event systems [10], and coding theory [27].

¹Google returns 625 entries upon entry of the query “joint spectral radius.”

Despite its natural interpretation, the joint spectral radius is difficult to compute. Questions related to its computability and to the existence of efficient approximation algorithms have been posed more than a decade ago (see [34] and [24]). In principle, the spectral radius can be approximated to any desired accuracy by computing converging sequences of upper and lower bounds. The following bounds, proved in [24],

$$(1.3) \quad \max_{\sigma \in \{1, \dots, m\}^k} \rho(A_\sigma)^{1/k} \leq \rho(A_1, \dots, A_m) \leq \max_{\sigma \in \{1, \dots, m\}^k} \|A_\sigma\|^{1/k},$$

can be evaluated for increasing values of k and lead to arbitrary accurate approximations of ρ (see, e.g., [16] or [17]). Such approximation algorithms can in turn be used in procedures that decide, after finitely many steps, whether $\rho > 1$ or $\rho < 1$. Such procedures are given, e.g., by Brayton and Tong [13] in a system theory context and by Barabanov [2] in the context of discrete linear inclusions. These procedures may, however, not terminate when ρ happens to be equal to 1, and the existence of algorithms for computing arbitrarily precise approximations of ρ does therefore not rule out the possibility that the decision problem “ $\rho < 1$?” is undecidable. It is so far unknown whether this is the case or not (see [24] for a discussion of this issue and for a description of its connection with the finiteness conjecture that has since then been proved to be false; see [11] and [6]). A negative result in this direction is given in [8] where it is proved that the related problem “ $\rho \leq 1$?” is algorithmically undecidable.

An approach based on optimal control techniques for approximating the joint spectral radius is explained in [37] and in more detail in Appendix D of [14]. Approximations that are directly based on the inequalities (1.3) are expensive to compute. In [26], the exponential number of products that appear in the direct and naive computation of the bounds in (1.3) is reduced by avoiding duplicate computation of cyclic permutations; the total number of product to consider remains, however, exponential. In addition to this, there are to this date no known theoretical guarantees for the uniform rate of convergence of the bounds appearing in (1.3). Convergence rates are supplied in [36] and [37], but the bounds appearing there depend on the set of matrices considered. Approximations of arbitrary degree of accuracy can be computed but at a price that may happen to be prohibitive.

There are in fact intrinsic limitations for the rate at which the joint spectral radius can be approximated. Let us say that the value ξ approximates the value ρ with relative accuracy $\mu \in [0, 1]$ (or 100μ %) if $\mu \xi \leq \rho \leq \xi$. By using a small adaptation of a proof appearing in [32] we show that, unless $P=NP$, there is no algorithm that can compute the joint spectral radius of two matrices with relative accuracy $1 - \epsilon$ in time polynomial in the size of the matrices and in $1/\epsilon$ (see later for more precise definitions). Despite this negative result it is still conceivable, as pointed out in [9], that for any desired accuracy, there exists a polynomial time algorithm that computes the joint spectral radius of the matrices with that accuracy. We prove in this paper that this is indeed the case. In Theorem 5, we prove that the joint spectral radius of m matrices of size n can be approximated with relative accuracy $1/\sqrt{m}$ by computing the spectral radius of a single matrix whose size is less than n^2 . This procedure can be applied in a recursive way, and in general we show how a relative accuracy of $(1/\sqrt{m})^{1/k}$ can be obtained by computing the spectral radius of a single matrix of size less than n^{2k} . As an illustration, the spectral radius of two matrices of size n can be computed with an accuracy of 70% by computing the spectral radius of a single matrix of size n^2 , and an accuracy of 95% can be obtained for the joint spectral radius of three matrices by computing the spectral radius of a single matrix of size n^{11} . More generally, for any number of matrices in the set and desired relative

accuracy we construct in section 4 a single matrix whose spectral radius approximates the joint spectral radius with the desired accuracy. The approximation procedure runs in polynomial time once the desired accuracy and the number of matrices in the set are fixed. More precisely, our approximation procedures provide approximations of relative accuracy $1 - \epsilon$ in time polynomial in $n^{(\ln m)/\epsilon}$, where m is the number of matrices and n is their size. Notice that $n^{(\ln m)/\epsilon} = e^{(\ln n \ln m)/\epsilon} = m^{(\ln n)/\epsilon}$, and so our approximation procedure also runs in time polynomial in the number of matrices once the desired accuracy and the size of the matrices are fixed. These bounds are close from optimality since we prove that, unless $P=NP$, no approximation polynomial time algorithm in n and $1/\epsilon$ is possible.

We now briefly describe how our results are obtained and how the paper is organized. We first notice in section 2 that for matrices with nonnegative entries the joint spectral radius satisfies

$$(1.4) \quad \frac{1}{m} \rho(A_1 + \cdots + A_m) \leq \rho(A_1, \dots, A_m) \leq \rho(A_1 + \cdots + A_m).$$

(The left-hand side inequality is valid for arbitrary matrices; there is no need to assume that the matrices have nonnegative entries.) Matrices with nonnegative entries are exactly those matrices A that are such that $A \mathbf{R}_+^n \subseteq \mathbf{R}_+^n$, and we prove in Theorem 1 that the inequalities (1.4) are satisfied not only for sets of matrices with nonnegative entries but also for sets of matrices that leave a proper cone invariant, i.e., matrices A_i that are such that $A_i K \subseteq K$ for some proper cone K and all i . Thus for these matrices the inequalities (1.4) provide a relative accuracy $1/m$. This accuracy can be improved by considering Kronecker powers of matrices. In section 3 we give an elementary proof that the joint spectral radius of the k th Kronecker powers of the matrices in a set is equal to the k th power of the joint spectral radius of the set (Theorem 3). Combining this with the relation (1.4) we prove that the approximation

$$\rho^{1/k}(A_1^{\otimes k} + \cdots + A_m^{\otimes k})$$

has relative accuracy $1/m^{1/k}$. Some of the arguments used to derive this result are valid only for matrices that have a common proper invariant cone. In section 4 we show how similar arguments can be used for arbitrary matrices. We introduce a semidefinite lifting procedure that transforms a linear operator acting $\mathbf{R}^n$ into a linear operator acting on the space of symmetric $n \times n$ matrices. It appears that under this transformation the joint spectral radius is simply squared. Moreover, the lifting has the interesting feature that the operator defined in this way leaves the cone of semidefinite matrices invariant. This observation leads to a simple approximation procedure (Theorem 5) for arbitrary matrices. Implementation issues and numerical examples are provided in section 5. Our approximation procedure provides a relative accuracy $1 - \epsilon$ in time polynomial in $n^{(\ln m)/\epsilon}$. We prove in section 6 (Theorem 6) that the same accuracy cannot be obtained in time polynomial in n and $1/\epsilon$ unless $P=NP$. The semidefinite lifting introduced in section 4 is a fundamental and powerful tool for stability analysis. As an illustration of this, we prove in section 7 how to apply our results to analyze the quality of the so-called ellipsoid approximation of the joint spectral radius. We prove that the ellipsoid approximation (a notion that is formally defined in [6] but that is implicitly present in a number of earlier contributions on hybrid and time-varying systems; see, e.g., [21] and [1], as well as some of the references in [25]) is an approximation of guaranteed accuracy $1/\sqrt{m}$; a proof of this corollary can also be extracted from section 3 of [21]. It turns out that the ellipsoid approximation is also an approximation of a guaranteed accuracy $1/\sqrt{n}$ (see, e.g., [6] and [1]), and so

our bound is an improvement of this bound for the case where the dimension of the matrices is larger than the cardinality of the set. Finally, in a last section, we discuss some of our results.

2. Approximation for matrices leaving a cone invariant. In this section, we consider sets of matrices that leave a proper cone invariant, and we show with elementary arguments how joint spectral radius approximations of guaranteed accuracy can easily be computed for this case. We start with a proof that the spectral radius of a convex combination of matrices is always less than or equal to the joint spectral radius of the matrices. This result is valid for all sets of matrices (there is no need to assume that the matrices have nonnegative entries) and is proved in [7] using the main result of [23]. Here we present a direct and elementary justification.

LEMMA 1. Let $\{A_i : i = 1, \dots, m\}$ be a set of matrices in $\mathbf{R}^{n \times n}$ such that $\alpha_i \geq 0$ and $\sum_{i=1}^n \alpha_i = 1$.

$$(2.1) \quad \rho\left(\sum_{i=1}^m \alpha_i A_i\right) \leq \rho(A_1, \dots, A_m).$$

Proof. Let us fix some $\alpha_i \geq 0$ with $\sum_i \alpha_i = 1$ and an integer $k \geq 1$. Then

$$\left\|\left(\sum_{i=1}^m \alpha_i A_i\right)^k\right\| = \left\|\sum_{\sigma \in \{1, \dots, m\}^k} \alpha_\sigma A_\sigma\right\| \leq \sum_{\sigma \in \{1, \dots, m\}^k} \alpha_\sigma \|A_\sigma\| \leq \max_{\sigma \in \{1, \dots, m\}^k} \|A_\sigma\|,$$

but then also

$$\lim_{k \rightarrow \infty} \left\|\left(\sum_{i=1}^m \alpha_i A_i\right)^k\right\|^{1/k} = \limsup_{k \rightarrow \infty} \left\|\left(\sum_{i=1}^m \alpha_i A_i\right)^k\right\|^{1/k} \leq \limsup_{k \rightarrow \infty} \max_{\sigma \in \{1, \dots, m\}^k} \|A_\sigma\|^{1/k},$$

and the result then follows from the definitions of the spectral radius (1.1) and of the joint spectral radius (1.2). $\square$

An immediate consequence of this lemma is given by the following corollary.

COROLLARY 1.

$$\frac{1}{m} \rho\left(\sum_{i=1}^m A_i\right) \leq \rho(A_1, \dots, A_m). \quad \square$$

The example $A_1 = A, A_2 = -A$ clearly shows that we cannot hope in general to have $\rho(A_1, \dots, A_m) \leq \rho(\sum_{i=1}^m A_i)$. This inequality is nevertheless satisfied when the matrices A_i leave a proper cone invariant. A subset $K \subseteq \mathbf{R}^n$ is a proper cone if it is closed, convex, has nonempty interior, and contains no straight line. For example, the set of vectors with nonnegative entries $\mathbf{R}_+^n$ is a proper cone, but $\mathbf{R}^n$ itself is not (see [30] for more background on cones and proper cones). We shall say that the matrices A_i leave a proper cone $K \subseteq \mathbf{R}^n$ invariant if there exists a proper cone $K \subseteq \mathbf{R}^n$ such that $A_i K \subseteq K$ for all i . For example, matrices with nonnegative entries leave the proper cone $\mathbf{R}_+^n$ invariant.

LEMMA 2. Let $K \subseteq \mathbf{R}^n$ be a proper cone. If $A_i K \subseteq K$ for all i , then $\|A+B\|_K \leq \|A\|_K + \|B\|_K$. Most usual matrix norms satisfy this property when $K = \mathbf{R}_+^n$; we provide a construction for arbitrary proper cones K .

Let $\|\cdot\|$ be some arbitrary vector norm. The dual vector norm $\|\cdot\|_*$ is defined by $\|w\|_* = \max_{\|v\|=1} w^T v$. Assume that K is a proper cone. Define the dual of K by

$$K^* = \{w \in \mathbf{R}^n : w^T v \geq 0 \ \forall v \in K\}.$$

The dual of a proper cone is again a proper cone; see [30]. Let us now consider the quantity

$$\|A\|_K = \max_{\substack{v \in K, w \in K^* \\ \|v\| = \|w\|_* = 1}} w^T Av.$$

It is easy to verify that because K and its dual are proper cones, the quantity $\|\cdot\|_K$ is indeed a matrix norm. Moreover, by the definition of the norm, matrices A and B that satisfy $AK \subseteq K$ and $BK \subseteq K$ are such that $\|A\|_K \leq \|A + B\|_K$. $\square$

We may now state the result of this section.

THEOREM 1. *Let $\{A_i : i = 1, \dots, m\}$ be a family of matrices such that*

$$(2.2) \quad \frac{1}{m} \rho \left(\sum_{i=1}^m A_i \right) \leq \rho(A_1, \dots, A_m) \leq \rho \left(\sum_{i=1}^m A_i \right).$$

The lower bound in (2.2) is already established in Corollary 1. Assume that the matrices in $\{A_i : i = 1, \dots, m\}$ leave the proper cone K invariant, and define the matrix norm $\|\cdot\|_K$ as in Lemma 2. Since $A_i K \subseteq K$, we also have $A_\sigma K \subseteq K$ for all $\sigma \in \{1, \dots, m\}^k$, but then also

$$\max_{\sigma \in \{1, \dots, m\}^k} \|A_\sigma\|_K \leq \left\| \sum_{\sigma \in \{1, \dots, m\}^k} A_\sigma \right\|_K = \left\| \left(\sum_{i=1}^m A_i \right)^k \right\|_K,$$

and therefore

$$\begin{aligned} \rho(A_1, \dots, A_m) &= \limsup_{k \rightarrow \infty} \max_{\sigma \in \{1, \dots, m\}^k} \|A_\sigma\|_K^{1/k} \\ &\leq \limsup_{k \rightarrow \infty} \left\| \left(\sum_{i=1}^m A_i \right)^k \right\|_K^{1/k} = \rho \left(\sum_{i=1}^m A_i \right). \quad \square \end{aligned}$$

It is interesting to notice that the conclusion of Theorem 1 does not directly involve the cone K . The existence of a proper cone satisfying the hypotheses suffices to conclude, and the exact nature of the cone is irrelevant.

A matrix with nonnegative entries leaves the cone $\mathbf{R}_+^n$ invariant, and so we have the following corollary.

COROLLARY 2. *Let $\{A_i : i = 1, \dots, m\}$ be a family of matrices such that*

$$\frac{1}{m} \rho \left(\sum_{i=1}^m A_i \right) \leq \rho(A_1, \dots, A_m) \leq \rho \left(\sum_{i=1}^m A_i \right). \quad \square$$

3. Kronecker lifting. We now describe a way to exploit Theorem 1 for obtaining approximations of arbitrary high accuracy. Our approximations involve Kronecker powers of matrices. The Kronecker product of two matrices $A \in R^{p_1 \times q_1}$ and $B \in R^{p_2 \times q_2}$ is the $p_1 p_2 \times q_1 q_2$ matrix defined by

$$A \otimes B = \begin{pmatrix} A_{1,1}B & \dots & A_{1,q_1}B \\ \vdots & \ddots & \vdots \\ A_{p_1,1}B & \dots & A_{p_1,q_1}B \end{pmatrix}.$$

We also define the Kronecker power of a matrix:

$$A^{\otimes k} = \underbrace{A \otimes A \dots A \otimes A}_{k \text{ times}}.$$

There is no need for parentheses in this expression since the Kronecker product of two matrices is an associative operation. Let us prove some elementary properties.

LEMMA 3.

1. $(A \otimes B)^T = A^T \otimes B^T$.
2. $(A_1 \otimes B_1)(A_2 \otimes B_2) = (A_1 A_2) \otimes (B_1 B_2)$.
3. $(A^{\otimes k})^T = (A^T)^{\otimes k}$.
4. $A^{\otimes k} B^{\otimes k} = (AB)^{\otimes k}$.
5. $\|A^{\otimes k}\| = \|A\|^k$.

The first statement directly follows from the definition of the Kronecker product. The second statement is quite standard; see, e.g., section 4.2 in [22]. The third and fourth statement follow, respectively, from repeated application of the first and second statement. Finally, for proving the last statement, note that by using (3) and (4) we obtain

$$\|A^{\otimes k}\|^2 = \max_{\|x\|=1} x^T (A^{\otimes k})^T A^{\otimes k} x = \max_{\|x\|=1} x^T (A^T)^{\otimes k} A^{\otimes k} x = \max_{\|x\|=1} x^T (A^T A)^{\otimes k} x.$$

The matrix $(A^T A)^{\otimes k}$ is symmetric, and so the last expression is also equal to the largest magnitude of the eigenvalues of $(A^T A)^{\otimes k}$. For a matrix B the eigenvalues of the matrix $B^{\otimes k}$ are given by all possible products of k eigenvalues of B (see Theorem 4.2.12 in [22]), and we therefore conclude $\max_{\|x\|=1} x^T (A^T A)^{\otimes k} x = \|A\|^{2k}$. $\square$

We can now prove a useful identity for the spectral radius of Kronecker powers of matrices.

THEOREM 2. Let $\{A_i : i = 1, \dots, m\}$ be a set of matrices with $\|A_i\| \geq 1$ for all i .

$$\rho(A_1^{\otimes l}, \dots, A_m^{\otimes l}) = \rho^l(A_1, \dots, A_m).$$

Let $\sigma \in \{1, \dots, m\}^k$. By using the equalities (3) and (5) of Lemma 3 we obtain

$$\|A_{\sigma_k}^{\otimes l} \dots A_{\sigma_1}^{\otimes l}\| = \|A_{\sigma}^{\otimes l}\| = \|A_{\sigma}\|^l.$$

The result then follows from the definition of the joint spectral radius (1.2). $\square$

We now would like to combine Theorem 2 with Theorem 1 in order to obtain approximations of arbitrary high accuracy for matrices leaving a cone invariant. In order to do this we need to consider Kronecker products of cones. For two sets $Q_1 \subseteq R^n$ and $Q_2 \subseteq R^m$ let us denote

$$Q_1 \otimes Q_2 = \{z = u \otimes v : u \in Q_1, v \in Q_2\}.$$

(In this definition, the vectors u and v are treated as column matrices.) If K_1 and K_2 are proper cones, then $K_1 \otimes K_2$ does not need to be a proper cone. Indeed, consider, for example, the proper cones $K_1 = K_2 = \mathbf{R}_+^2$ for which it is easy to see that $K_1 \otimes K_2$ has empty interior in $\mathbf{R}^4$. We do, however, have the following result.

LEMMA 4. *If $K_1 \subset \mathbf{R}^n$ and $K_2 \subset \mathbf{R}^m$ are proper cones, then $K = \text{Conv}(K_1 \otimes K_2) \subset \mathbf{R}^{nm}$ is a proper cone. The cone K is closed and convex by definition. Assume that $\text{int } K = \emptyset$. Then there exists $A \in \mathbf{R}^{nm} \setminus \{0\}$ such that*

$$\langle A, z \rangle = 0 \quad \forall z \in K \supset K_1 \otimes K_2.$$

Note that the function $f(u, v) = \langle A, u \otimes v \rangle$ is a bilinear form defined on $K_1 \times K_2$. Since $\text{int}(K_1 \times K_2) \neq \emptyset$, f can vanish on this set only if $A = 0$. That is a contradiction.

Let us prove now that K contains no straight line. Assume that this is not the case. Then $\text{int } K^* = \emptyset$, which implies the existence of some $B \in \mathbf{R}^{nm} \setminus \{0\}$ such that

$$(3.1) \quad \langle B, w \rangle = 0 \quad \forall w \in K^*.$$

Since $\langle u \otimes v, x \otimes y \rangle = \langle u, x \rangle \cdot \langle v, y \rangle$, we have $K_1^* \otimes K_2^* \subseteq K^*$. Since K_1^* and K_2^* have nonempty interior, relation (3.1) is impossible by the first part of the proof. $\square$

We are ready to state the main result of this section.

THEOREM 3. *Let $\{A_i : i = 1, \dots, m\}$ be a family of matrices in $\mathbf{R}^{n \times n}$ such that*

$$\frac{1}{m^{1/k}} \rho^{1/k}(A_1^{\otimes k} + \dots + A_m^{\otimes k}) \leq \rho(A_1, \dots, A_m) \leq \rho^{1/k}(A_1^{\otimes k} + \dots + A_m^{\otimes k}).$$

then the following limit exists and is equal to the spectral radius of the family:

$$(3.2) \quad \rho(A_1, \dots, A_m) = \lim_{k \rightarrow \infty} \rho^{1/k}(A_1^{\otimes k} + \dots + A_m^{\otimes k}).$$

Assume that the matrices leave the proper cone K invariant. If $AK \subseteq K$, then $(AK)^{\otimes k} \subseteq K^{\otimes k}$ and, by Lemma 3, $A^{\otimes k} K^{\otimes k} \subseteq K^{\otimes k}$. But then also $A^{\otimes k} \text{Conv } K^{\otimes k} \subseteq \text{Conv } K^{\otimes k}$. By Lemma 4 the cone $\text{Conv } K^{\otimes k}$ is a proper cone. Thus the matrices $A_i^{\otimes k}$ leave a proper cone invariant, and by Theorem 1 we have

$$\frac{1}{m} \rho(A_1^{\otimes k} + \dots + A_m^{\otimes k}) \leq \rho(A_1^{\otimes k}, \dots, A_m^{\otimes k}) \leq \rho(A_1^{\otimes k} + \dots + A_m^{\otimes k}).$$

In order to conclude it suffices to use the fact, proved in Theorem 2, that

$$\rho(A_1^{\otimes k}, \dots, A_m^{\otimes k}) = \rho^k(A_1, \dots, A_m). \quad \square$$

It is somewhat surprising to notice that the right-hand side in (3.2) does not directly involve any mixed product between the matrices.

An approximation of relative accuracy $(1/m)^{1/k}$ can thus be obtained by computing the spectral radius of a single matrix of dimension n^k . For pairs of matrices, some of the relative accuracies and the corresponding matrix sizes are as follows:

Relative accuracy	0.707	0.840	0.917	0.957
Matrix size	n^2	n^4	n^8	n^{16}

4. Semidefinite lifting. The result in Theorem 3 provides an easy way to evaluate with arbitrary accuracy the joint spectral radius of matrices that leave a proper cone invariant. Unfortunately, matrices do not always leave a proper cone invariant. We show in this section how an invariant cone can always be created by semidefinite lifting.

Let $A \in \mathbf{R}^{n \times n}$ and consider the following linear operator:

$$(4.1) \quad X \rightarrow AXA^T : \mathbf{R}^{n \times n} \rightarrow \mathbf{R}^{n \times n}.$$

A matrix representation for this linear operator can be obtained by using the matrix-to-vector operator that develops a matrix into a vector by taking its columns one by one. This operator, denoted vec , satisfies the elementary property $\text{vec}(CXD) = (D^T \otimes C) \text{vec}(X)$ (see Lemma 4.3.1 in [22]). We therefore have $\text{vec}(AXA^T) = (A \otimes A) \text{vec}X$, and a matrix representation of the linear operator (4.1) is thus simply given by $A \otimes A$.

Consider now the space of symmetric matrices $\mathbf{S}$. This space is a subspace of $\mathbf{R}^{n \times n}$ of dimension $n(n+1)/2$, and the operator

$$(4.2) \quad X \rightarrow AXA^T : \mathbf{S} \rightarrow \mathbf{S}$$

is again a linear operator. Matrix representations of this operator are of course different from those of the operator (4.1) since, in particular, the spaces on which these operators operate have different dimensions. (We describe in the next section how to construct a matrix representation for the operator on symmetric matrices.) In the next theorem we prove that even though the operators $X \rightarrow AXA^T$ on $\mathbf{S}$ and on $\mathbf{R}^{n \times n}$ are different, their joint spectral radius are equal.

THEOREM 4. Let $\{A_i \in \mathbf{R}^{n \times n} : i = 1, \dots, m\}$ be a family of matrices and let $M_A = \frac{1}{m} \sum_{i=1}^m A_i \otimes A_i$. Then

$$\rho(M_{A_1}, \dots, M_{A_m}) = \rho(A_1 \otimes A_1, \dots, A_m \otimes A_m) = \rho^2(A_1, \dots, A_m).$$

The second equality is already proved in Theorem 2; we need only to prove the first equality. Let $\|\cdot\|_F$ denote the Frobenius matrix norm (i.e., the square root of the sum of squares of all entries), and consider the resulting induced operator norms for $X \rightarrow AXA^T$ on $\mathbf{R}^{n \times n}$ and $\mathbf{S}$:

$$\sup_{X \in \mathbf{R}^{n \times n}, \|X\|_F=1} \|AXA^T\|_F \quad \text{and} \quad \sup_{X \in \mathbf{S}, \|X\|_F=1} \|AXA^T\|_F.$$

We claim that these two operator norms are equal and that the supremum is achieved for some symmetric matrix of rank one. Indeed, using the matrix-to-vector operator vec and denoting by $\|\cdot\|$ the usual vector Euclidean norm we get

$$\begin{aligned} \sup_{X \in \mathbf{R}^{n \times n}, \|X\|_F=1} \|AXA^T\|_F^2 &= \sup_{x \in \mathbf{R}^{n^2}, \|x\|=1} \|(A \otimes A)x\|^2 \\ &= \sup_{x \in \mathbf{R}^{n^2}, x^T x=1} x^T (A \otimes A)^T (A \otimes A) x \\ &= \sup_{x \in \mathbf{R}^{n^2}, x^T x=1} x^T (A^T A \otimes A^T A) x. \end{aligned}$$

For deriving the last equality, we have used point 2 in Lemma 3. The supremum in the last expression is achieved by any eigenvector x associated with an eigenvalue

of largest magnitude of $A^T A \otimes A^T A$. Let $(\lambda_i, v_i), i = 1, \dots, n$, be the set of eigenvalues/eigenvectors pairs associated with the symmetric matrix $A^T A$, and let λ_1 be such that $|\lambda_1| \geq \lambda_i$ for all i . The eigenvalues/eigenvectors pairs of the matrix $A^T A \otimes A^T A$ are then given by $(\lambda_i \lambda_j, v_i \otimes v_j)$ for $i, j = 1, \dots, n$. The vector $v_1 \otimes v_1$ is thus an eigenvector of $A^T A \otimes A^T A$ of largest eigenvalue magnitude. Since $\text{vec}(v_k v_k^T) = v_k \otimes v_k$ we see that the supremum of $\|AXA^T\|_F$ is achieved for the symmetric rank one matrix $v_1 v_1^T$. From this we conclude that

$$\sup_{X \in \mathbf{R}^{n \times n}, \|X\|_F=1} \|AXA^T\|_F = \sup_{X \in \mathbf{S}, \|X\|_F=1} \|AXA^T\|_F,$$

and thus also

$$\sup_{\|x\|=1} \|(A \otimes A)x\| = \sup_{\|x\|=1} \|(M_A)x\|.$$

Notice that $M_{A_\sigma} = (M_{A_i})_\sigma$, and so

$$\|(M_{A_i})_\sigma\| = \|M_{A_\sigma}\| = \|A_\sigma \otimes A_\sigma\| = \|(A_i \otimes A_i)_\sigma\|;$$

it then suffices to apply the definition of the joint spectral radius to conclude the proof. $\square$

A matrix A is *symmetric positive semidefinite* (which we denote by $A \succeq 0$) if $v^T A v \geq 0$ for all $v \in \mathbf{R}^n$. The set of symmetric positive semidefinite matrices, denoted $\mathbf{S}_+$, is a cone. It is not a proper cone of $\mathbf{R}^{n \times n}$ because it has empty interior in $\mathbf{R}^{n \times n}$, but it is a proper cone of the set of symmetric matrices $\mathbf{S}$. If $X \in \mathbf{S}_+$, then we clearly have $AXA^T \in \mathbf{S}_+$, and so the operator $X \rightarrow AXA^T$ leaves the proper cone $\mathbf{S}_+$ invariant. Combining this observation with Theorems 3 and 4 we deduce the following theorem.

THEOREM 5. *Let $\{A_i \in \mathbf{R}^{n \times n} : i = 1, \dots, m\}$ be a finite set of symmetric positive semidefinite matrices. Then*

$$(4.3) \quad \frac{1}{\sqrt{m}} \rho^{1/2}(M_{A_1} + \dots + M_{A_m}) \leq \rho(A_1, \dots, A_m) \leq \rho^{1/2}(M_{A_1} + \dots + M_{A_m}).$$

Note that for $m = 2$ the bounds (4.3) are not so bad. Indeed, $1/\sqrt{2} > 0.7$, and so the relative accuracy of the approximation is at least 70%. The main interest of this result resides, however, in the possibility of applying it recursively. Starting from an initial matrix $A \in \mathbf{R}^{n \times n}$, we define a sequence of operators acting on spaces of increasing dimensions. For $l = 1$ we define

$$A^{[1]}(x) \equiv Ax, \quad x \in E_1 \equiv \mathbf{R}^n,$$

and for $l \geq 2$ we define the operators recursively:

$$A^{[l]}(x) \equiv M_{A^{[l-1]}}(x), \quad x \in E_l \equiv \mathbf{S}(E_{l-1}).$$

Assume for the simplicity of the presentation that $m = 2$, and let $A_1, A_2 \in \mathbf{R}^{n \times n}$. From Theorem 4 we know that

$$\rho(A_1^{[l]}, A_2^{[l]}) = \rho^{2^l}(A_1, A_2).$$

On the other hand, for any $l \geq 1$ the operators $A_1^{[l]}$ and $A_2^{[l]}$ leave the cone $\mathbf{S}_+(E_{l-1})$ invariant. Therefore, in view of Theorem 1, we have

$$\frac{1}{2} \rho(A_1^{[l]} + A_2^{[l]}) \leq \rho(A_1^{[l]}, A_2^{[l]}) \leq \rho(A_1^{[l]} + A_2^{[l]}).$$

Combining these last two expressions, we get the following bounds:

$$\left(\frac{1}{2}\right)^{1/2^l} \left[\rho \left(A_1^{[l]} + A_2^{[l]} \right) \right]^{1/2^l} \leq \rho(A_1, A_2) \leq \left[\rho \left(A_1^{[l]} + A_2^{[l]} \right) \right]^{1/2^l}.$$

Note that

$$1 - \frac{\ln 2}{2^l} \leq \left(\frac{1}{2}\right)^{1/2^l} \leq 1,$$

and thus the improvement in the quality of our approximation is quite fast. Unfortunately, the dimensions of the spaces E_l are also growing fast; we have

$$n_{l+1} = \frac{1}{2}n_l(n_l + 1)$$

(where $n_i = \dim E_i$), and therefore asymptotically we have $n_l = O((n/2)^{2^l})$. Let us display, for a pair of matrices, the relative accuracy of our approximation as a function of the resulting dimension.

Steps	Accuracy	$n = 2$	$n = 10$	$n = 100$
1	0.707	3	55	5050
2	0.840	6	1540	*
3	0.917	21	118570	*
4	0.957	231	*	*
5	0.978	26796	*	*

We use the symbol * to mark the cases for which the dimension of the auxiliary problem goes beyond the abilities of modern computers.

5. Numerical implementation and examples. The recursive semidefinite lifting for obtaining approximations of the joint spectral radius of increasing accuracy may be difficult to implement because the matrix M_A is not easy to express in terms of the matrix A . Consider, for example, the case of 2×2 matrices. Let

$$A = \begin{pmatrix} a_{11} & a_{12} \\ a_{21} & a_{22} \end{pmatrix};$$

then one can easily compute

$$M_A = \begin{pmatrix} a_{11}^2 & 2a_{11}a_{12} & a_{12}^2 \\ a_{11}a_{21} & a_{11}a_{22} + a_{12}a_{21} & a_{12}a_{22} \\ a_{21}^2 & 2a_{21}a_{22} & a_{22}^2 \end{pmatrix}.$$

In general, the matrix M_A can be expressed as follows:

$$(5.1) \quad M_A = (Q^T Q)^{-1} Q^T P^{-1} (A \otimes A) P Q,$$

where P is a particular permutation matrix of size n^2 and Q is given by

$$Q = \begin{pmatrix} I_n & 0 \\ 0 & I_{\frac{n(n-1)}{2}} \\ 0 & I_{\frac{n(n-1)}{2}} \end{pmatrix}.$$

(The matrix I_k is the identity matrix of size k .) One difficulty with expression (5.1) is that the permutation matrix P is tedious to construct; it essentially relates the column decomposition of a matrix with its diagonal decomposition. The construction

of the permutation matrix is somewhat cumbersome, but it can be automated. A way to bypass repeated computation of the matrix for larger and larger dimensions can be achieved by using the semidefinite lifting only once and then applying only Kronecker liftings. The semidefinite lifting can be performed either at the beginning or at the end of the process, leading to the bounds

$$(5.2) \quad \frac{1}{m} \rho(M_{A_1}^{\otimes l} + \cdots + M_{A_m}^{\otimes l}) \leq \rho^{2l}(A_1, \dots, A_m) \leq \rho(M_{A_1}^{\otimes l} + \cdots + M_{A_m}^{\otimes l})$$

and

$$(5.3) \quad \frac{1}{m} \rho(M_{A_1^{\otimes l}} + \cdots + M_{A_m^{\otimes l}}) \leq \rho^{2l}(A_1, \dots, A_m) \leq \rho(M_{A_1^{\otimes l}} + \cdots + M_{A_m^{\otimes l}}).$$

The validity of these inequalities results from the combination of Theorem 1, Theorem 2, Lemma 4, and Theorem 4. Expressions (5.2) and (5.3) both provide a relative accuracy of $(1/m)^{1/(2l)}$ but involve matrices of different size. The matrices $M_{A_i}^{\otimes l}$ have size $(n(n+1)/2)^l$, whereas the matrices $M_{A_i^{\otimes l}}$ have size $n^l(n^l+1)/2 > (n(n+1)/2)^l$. In addition to this, the matrices $M_{A_i}^{\otimes l}$ are easier to compute than $M_{A_i^{\otimes l}}$ since they necessitate the evaluation of semidefinite liftings of smaller matrices. The approximation based on (5.2) is therefore to be preferred because it is easier to evaluate and leads to matrices of smaller dimension.

Of course, other combinations of Kronecker and semidefinite liftings are also possible. Whenever a k th Kronecker power is used, the current relative accuracy is k th rooted, and whenever a semidefinite lifting is used, the relative accuracy is square rooted. Moreover, at least one semidefinite lifting is needed if the original matrices do not leave a proper cone invariant.

For the numerical implementation of these approximations, notice also that the spectral radius of $M_{A_1} + \cdots + M_{A_m}$ is the spectral radius of the linear operator

$$B : X \rightarrow A_1 X A_1^T + \cdots + A_m X A_m^T : \mathbf{S} \rightarrow \mathbf{S},$$

and it can therefore be found by standard linear algebra techniques or by solving the following linear inequality optimization problem,

$$(5.4) \quad \inf_{X, \tau} \{ \tau : \tau X \succeq A_1 X A_1^T + \cdots + A_m X A_m^T, X \succ 0 \},$$

for which efficient convex optimization algorithms are available (see, e.g., [12]) that are implemented in standard commercial softwares such as the MATLAB “LMI Control Toolbox.”

6. Computational complexity analysis. The joint spectral radius can be approximated to arbitrary accuracy. It is proved in [32] that, unless $P=NP$, approximating algorithms of relative accuracy $1 - \epsilon$ for pairs of matrices cannot run in time polynomial in the size of the matrices and in $\ln(1/\epsilon)$. It was later noticed (see note 9 on page 1260 of [9]) that a careful examination of the proof in [32] shows that, unless $P=NP$, there are no approximation algorithms of relative accuracy $1 - \epsilon$ that run in time polynomial in the size of the matrices and in $1/\epsilon$. This is, however, not proved in [9]. In this section we prove this result by extracting the essential part of the proof in [32].

We proceed by reduction from the classical NP-complete satisfiability problem 3-SAT. This problem is defined as follows. Consider a set $\{x_1, \dots, x_n\}$ of Boolean variables. A literal is either a Boolean variable, x_i , or its negation, $\bar{x}_i$. A clause is a disjunction of three literals.

is a disjunction of three literals (e.g., $x_3 \vee \bar{x}_5 \vee \bar{x}_6$). The 3-SAT problem is the problem of determining, for a given collection of three-literal clauses, if there exists a truth assignment for the variables that simultaneously satisfies all clauses. This problem is known to be NP-complete; finding a polynomial time algorithm for the problem is equivalent to proving that $P=NP$.

The complexity result proved in [32] is based on the construction of a pair of matrices which we briefly outline. (The complete construction can easily be extracted from [32].) From a given instance of 3-SAT with n variables and p clauses, two square matrices A_1, A_2 are constructed in polynomial time. The matrices have their entries in $\{0, 1\}$, are of size $(n+1)(p+1)$, and have a joint spectral radius that satisfies

$$\begin{aligned} \rho(A_1, A_2) &\geq p^{1/(n+2)} && \text{if the instance is satisfiable,} \\ \rho(A_1, A_2) &\leq (p-1)^{1/(n+2)} && \text{if the instance is not satisfiable.} \end{aligned}$$

Any algorithm of relative accuracy $(1 - 1/p)^{1/(n+2)}$ allows us to make the distinction between these two cases. Moreover, since

$$\left(1 - \frac{1}{p}\right)^{1/(n+2)} \leq 1 - \frac{1}{p(n+2)} \leq 1,$$

it is clear that an approximation algorithm of relative accuracy $1 - 1/p(n+2)$ also allows us to decide 3-SAT. Since the size of the matrices in the construction are given by $(n+1)(p+1)$, we deduce the following theorem.

THEOREM 6.

Let A_1, A_2 be two square matrices with entries in $\{0, 1\}$ of size $(n+1)(p+1)$ such that $\rho(A_1, A_2) \geq 1 - \epsilon$.

From Theorem 3 we know that the spectral radius of two matrices with nonnegative entries is given by $\rho(A_1, A_2) = \lim_{k \rightarrow \infty} \rho^{1/k}(A_1^{\otimes k} + A_2^{\otimes k})$, and so the quantity $\lim_{k \rightarrow \infty} \rho^{1/k}(A_1^{\otimes k} + A_2^{\otimes k})$ is NP-hard to approximate in the sense given in Theorem 6. Theorem 3 provides also a rate of convergence for the approximations $\rho^{1/k}(A_1^{\otimes k} + A_2^{\otimes k})$. From this we may prove the following complexity result.

THEOREM 7. Let A_1, A_2 be two square matrices with entries in $\{0, 1\}$ of size $(n+1)(p+1)$ and $k \geq 0$.

If $\rho(A_1, A_2) \geq 1 - \epsilon$, then

$$\rho(A_1^{\otimes k} + A_2^{\otimes k}) < 1$$

if and only if

We proceed by reduction from 3-SAT. From a given instance of 3-SAT with n variables and p clauses we use the construction given in [32] to construct two square matrices B_1, B_2 that have their entries in $\{0, 1\}$, are of size $(n+1)(p+1)$, and have a joint spectral radius that satisfies

$$\begin{aligned} \rho(B_1, B_2) &\geq p^{1/(n+2)} && \text{if the instance is satisfiable,} \\ \rho(B_1, B_2) &\leq (p-1)^{1/(n+2)} && \text{if the instance is not satisfiable.} \end{aligned}$$

We then choose $\alpha \in \mathbf{R}$ with $0 < \alpha < 1$, $r \in \mathbf{Q}$, and $k \geq 0$ such that $\alpha < (1/2)^{1/k}$ and

$$(6.1) \quad (p-1)^{\frac{1}{n+2}} < \frac{1}{\alpha}(p-1)^{\frac{1}{n+2}} < r < \alpha q^{\frac{1}{n+2}} < p^{\frac{1}{n+2}}.$$

Consider now the matrices $A_1 = (1/r)B_1$ and $A_2 = (1/r)B_2$. We claim that $\rho(A_1^{\otimes k} + A_2^{\otimes k}) < 1$ if and only if the instance of 3-SAT is not satisfiable. Indeed, assume first that the instance of 3-SAT is not satisfiable. Then

$$\rho(B_1, B_2) \leq (p-1)^{1/(n+2)}.$$

Since $\rho(A_1^{\otimes k} + A_2^{\otimes k})$ is an approximation of relative accuracy $1/2^{2k}$ of $\rho(A_1, A_2)$, we have

$$\rho(A_1^{\otimes k} + A_2^{\otimes k}) \leq 2^{2k} \rho(A_1, A_2) \leq \frac{2^{2k}}{r} \rho(B_1, B_2) \leq \frac{2^{2k}}{r} (p-1)^{1/(n+2)},$$

and by using the inequalities (6.1) we conclude that $\rho(A_1^{\otimes k} + A_2^{\otimes k}) < 1$. In an analogous way one can see that if the instance of 3-SAT is satisfiable, then $\rho(A_1^{\otimes k} + A_2^{\otimes k}) > 1$, and the proof is therefore complete. $\square$

7. Ellipsoid approximation. Let us now compare our approximation of the joint spectral radius with another approximation appearing in the literature. The following approximation, called ellipsoid approximation, is introduced in [1] (see also [7]):

$$(7.1) \quad \hat{\rho}(A_1, \dots, A_m) = \left[\inf_{X, \tau} \{ \tau : \tau X \succeq A_i X A_i^T, i = 1, \dots, m, X \succ 0 \} \right]^{1/2}.$$

This expression corresponds to the best ellipsoid norm for the set of matrices. In [7] the following inequalities are proved:

$$\frac{1}{\sqrt{n}} \hat{\rho}(A_1, \dots, A_m) \leq \rho(A_1, \dots, A_m) \leq \hat{\rho}(A_1, \dots, A_m).$$

Thus, the ellipsoid approximation has a relative accuracy of $1/\sqrt{n}$. This accuracy decreases with the size of the matrices but does not depend on the number of matrices in the set. Using our results we can prove that the relative accuracy of the ellipsoid approximation is in fact bounded by $1/\sqrt{m}$, where m is the number of matrices.

THEOREM 8. $\rho(A_1, \dots, A_m) \leq \hat{\rho}(A_1, \dots, A_m) \leq \frac{1}{\sqrt{m}} \rho(A_1, \dots, A_m)$.

$$\frac{1}{\sqrt{m}} \hat{\rho}(A_1, \dots, A_m) \leq \rho(A_1, \dots, A_m) \leq \hat{\rho}(A_1, \dots, A_m).$$

It is easy to see that $\rho(A_1, \dots, A_m) \leq \hat{\rho}(A_1, \dots, A_m)$. For proving the first inequality, note that the spectral radius of the linear operator

$$B : X \rightarrow \sum_{i=1}^m A_i X A_i^T : \mathbf{S} \rightarrow \mathbf{S}$$

can be represented as follows:

$$(7.2) \quad \rho(B) = \inf_{X, \tau} \left\{ \tau : \tau X \succeq \sum_{i=1}^m A_i X A_i^T, X \succ 0 \right\}.$$

If a pair (X, τ) is feasible for the optimization problem in (7.2), then it is also feasible for the optimization problem in (7.1). Therefore

$$\hat{\rho}(A_1, \dots, A_m) \leq \rho^{1/2}(B).$$

Finally, by Theorem 5 we have

$$\frac{1}{\sqrt{m}} \rho^{1/2}(B) \leq \rho(A_1, \dots, A_m),$$

and so we get the following bounds:

$$(7.3) \quad \frac{1}{\sqrt{m}}\hat{\rho}(A_1, \dots, A_m) \leq \frac{1}{\sqrt{m}}\rho^{1/2}(B) \leq \rho(A_1, \dots, A_m) \leq \hat{\rho}(A_1, \dots, A_m) \leq \rho^{1/2}(B).$$

Thus the ellipsoid estimate $\hat{\rho}(A_1, \dots, A_m)$ has relative quality $1/\sqrt{m}$. $\square$

By using Theorem 2 one can obtain a statement analogous to that of Theorem 3.

COROLLARY 3. $\{A_i : i = 1, \dots, m\}$ $\hat{\rho}^{1/k}(A_1^k, \dots, A_m^k) \leq \rho(A_1, \dots, A_m) \leq \hat{\rho}^{1/k}(A_1^k, \dots, A_m^k)$ (7.1)

$$(7.4) \quad \frac{1}{m^{1/(2k)}} \hat{\rho}^{1/k}(A_1^k, \dots, A_m^k) \leq \rho(A_1, \dots, A_m) \leq \hat{\rho}^{1/k}(A_1^k, \dots, A_m^k).$$

For $k = 1$, this result is also proved in [1]. For arbitrary k , a proof can be reconstructed from the result proved in [21]. Notice, however, that approximation algorithms that are directly based on (7.4) are not efficient because they require solving linear matrix inequalities of very large dimension. Approximations based on (5.2) are easier to obtain because there exist many efficient numerical algorithms for approximating the spectral radius of a matrix.

8. Discussion. The results presented in this paper introduce the possibility of computing approximations for the joint spectral radius with worst-case theoretical guarantees. Of course, the computational complexity of these estimates needs further examination. Indeed, the spectral structure of the matrix $A^{\otimes k}$ is very simple, and this simplicity must be inherited somehow by the matrix $A_1^{\otimes k} + A_2^{\otimes k}$. It therefore appears to be an interesting problem to investigate the possibility of constructing efficient procedures for finding the spectral radius of this sum; of course there are theoretical limitations on what can be achieved since we have shown that the problem of deciding $\rho(A_1^{\otimes k} + A_2^{\otimes k}) \leq 1$ is NP-hard.

Another interesting issue is that of determining if worst-case theoretical guarantees can be provided for the approximation

$$\max_{\sigma \in \{1, \dots, m\}^k} \rho(A_\sigma)^{1/k} \leq \rho(A_1, \dots, A_m).$$

In many numerical examples, this approximation does in fact perform at least as well.

When the paper was ready for publication, the authors discovered a very interesting theory developed by Protasov [28]. This theory also admits an interpretation in terms of Kronecker powers of matrices. However, it is focused mainly on geometrical properties and objects (invariant sets, p -sums of convex sets, etc.). Our approach is different in several aspects. First, it is simpler and it directly focused on computational aspects. Second, for some important cases (matrices with nonnegative elements) we have finer results. Finally, for our purposes we do not need additional assumptions (irreducibility).

REFERENCES

- [1] T. ANDO AND M.-H. SHIH, *Simultaneous contractibility*, SIAM J. Matrix Anal. Appl., 19 (1998), pp. 487–498.
- [2] N. E. BARABANOV, *On the Lyapunov exponent of discrete inclusions. I*, Avtomat. i Telemekh., no. 2, 1988, pp. 40–46; translation in *Automat. Remote Control*, 49 (1988), pp. 152–157.
- [3] N. E. BARABANOV, *On the Lyapunov exponent of discrete inclusions. II*, Avtomat. i Telemekh., no. 3, 1988, pp. 24–29; translation in *Automat. Remote Control*, 49 (1988), pp. 283–287.
- [4] N. E. BARABANOV, *On the Lyapunov exponent of discrete inclusions. III*, Avtomat. i Telemekh., no. 5, 1988, pp. 17–24; translation in *Automat. Remote Control*, 49 (1988), pp. 558–565.

- [5] M. A. BERGER AND Y. WANG, *Bounded semigroups of matrices*, Linear Algebra Appl., 166 (1992), pp. 21–27.
- [6] V. D. BLONDEL, J. THEYS, AND A. A. VLADIMIROV, *An elementary counterexample to the finiteness conjecture*, SIAM J. Matrix Anal. Appl., 24 (2003), pp. 963–970.
- [7] V. D. BLONDEL, Y. NESTEROV, AND J. THEYS, *On the accuracy of the ellipsoid norm approximation of the joint spectral radius*, Linear Algebra Appl., 394 (2005), pp. 91–107.
- [8] V. D. BLONDEL AND J. N. TSITSIKLIS, *The boundedness of all products of a pair of matrices is undecidable*, Systems Control Lett., 41 (2000), pp. 135–140.
- [9] V. D. BLONDEL AND J. N. TSITSIKLIS, *A survey of computational complexity results in systems and control*, Automat. J. IFAC, 36 (2000), pp. 1249–1274.
- [10] V. D. BLONDEL, S. GAUBERT, AND J. N. TSITSIKLIS, *Approximating the spectral radius of sets of matrices in the max-algebra is NP-hard*, IEEE Trans. Automat. Control, 45 (2000), pp. 1762–1765.
- [11] T. BOUSCH AND J. MAIRESSE, *Asymptotic height optimization for topological IFS, Tetris heaps and the finiteness conjecture*, J. Amer. Math. Soc., 15 (2002), pp. 77–111.
- [12] S. BOYD, L. EL GHAOU, E. FERON, AND V. BALAKRISHNAN, *Linear Matrix Inequalities in System and Control Theory*, SIAM Stud. Appl. Math. 15, SIAM, Philadelphia, 1994.
- [13] R. BRAYTON AND C. TONG, *Constructive stability and asymptotic stability of dynamical systems*, IEEE Trans. Circuits and Systems, 27 (1980), pp. 1121–1130.
- [14] F. COLONIUS AND W. KLIEMANN, *The Dynamics of Control*, Birkhäuser Boston, Boston, MA, 2000.
- [15] I. DAUBECHIES AND J. C. LAGARIAS, *Sets of matrices all infinite products of which converge*, Linear Algebra Appl., 161 (1992), pp. 227–263.
- [16] I. DAUBECHIES AND J. C. LAGARIAS, *Corrigendum/addendum to: Sets of matrices all infinite products of which converge*, Linear Algebra Appl., 327 (2001), pp. 69–83.
- [17] G. GRIPENBERG, *Computing the joint spectral radius*, Linear Algebra Appl., 234 (1996), pp. 43–60.
- [18] N. GUGLIELMI AND M. ZENNARO, *On the zero-stability of variable stepsize multistep methods: The spectral radius approach*, Numer. Math., 88 (2001), pp. 445–458.
- [19] L. GURVITS, *Stability of discrete linear inclusions*, Linear Algebra Appl., 231 (1995), pp. 47–85.
- [20] L. GURVITS, *Stability and observability of discrete linear inclusion – Finite automata approach. Book of abstracts*, in Proceedings of the IEEE International Symposium on the Mathematical Theory of Networks and Systems, Kobe, Japan, 1991, pp. 166–167.
- [21] L. GURVITS, *Stabilities and controllabilities of switched systems (with applications to the quantum systems)*, in Proceedings of the Fifteenth International Symposium on Mathematical Theory of Networks and Systems, University of Notre Dame, Notre Dame, IN, 2002.
- [22] R. A. HORN AND C. R. JOHNSON, *Topics in Matrix Analysis*, Cambridge University Press, Cambridge, UK, 1991.
- [23] V. S. KOZYAKIN, *Algebraic unsolvability of problem of absolute stability of desynchronized systems*, Automat. Remote Control, 51 (1990), pp. 754–759.
- [24] J. C. LAGARIAS AND Y. WANG, *The finiteness conjecture for the generalized spectral radius of a set of matrices*, Linear Algebra Appl., 214 (1995), pp. 17–42.
- [25] D. LIBERZON, J. P. HESPANHA, AND A. S. MORSE, *Stability of switched systems: A Lie-algebraic condition*, Systems Control Lett., 37 (1999), pp. 117–122.
- [26] M. MAESUMI, *An efficient lower bound for the generalized spectral radius*, Linear Algebra Appl., 240 (1996), pp. 1–7.
- [27] B. E. MOISION, A. ORLITSKY, AND P. H. SIEGEL, *On codes that avoid specified differences*, IEEE Trans. Inform. Theory, 47 (2001), pp. 433–442.
- [28] V. YU. PROTASOV, *The generalized joint spectral radius. A geometric approach*, Izv. Math., 61 (1997), pp. 995–1030.
- [29] G.-C. ROTA AND W. G. STRANG, *A note on the joint spectral radius*, Indag. Math., 22 (1960), pp. 379–381.
- [30] R. T. ROCKAFELLAR, *Convex Analysis*, Princeton University Press, Princeton, NJ, 1970.
- [31] G. STRANG, *The Joint Spectral Radius*, commentary by Gilbert Strang on paper number 5 in “Collected Works of Gian-Carlo Rota,” 2001.
- [32] J. N. TSITSIKLIS AND V. D. BLONDEL, *The Lyapunov exponent and joint spectral radius of pairs of matrices are hard – when not impossible – to compute and to approximate*, Math. Control Signals Systems, 10 (1997), pp. 31–40.
- [33] J. N. TSITSIKLIS, *On the stability of asynchronous iterative processes*, Math. Systems Theory, 20 (1987), pp. 137–153.
- [34] J. N. TSITSIKLIS, *The Stability of the Products of a Finite Set of Matrices*, in Open Problems in Communication and Computation, T. M. Cover and B. Gopinath, eds., Springer-Verlag,

- New York, 1987, pp. 161–163.
- [35] A. A. VLADIMIROV, L. ELSNER, AND W.-J. BEYN, *Stability and paracontractivity of discrete linear inclusions*, Linear Algebra Appl., 312 (2000), pp. 125–134.
 - [36] F. WIRTH, *The generalized spectral radius and extremal norms*, Linear Algebra Appl., 342 (2002), pp. 17–40.
 - [37] F. WIRTH, *On the calculation of real time-varying stability radii*, Internat. J. Robust Nonlinear Control, 8 (1998), pp. 1043–1058.

THE SINGULAR VALUE BEHAVIOR OF THE FINITE SECTIONS OF BLOCK TOEPLITZ OPERATORS*

A. ROGOZHIN†

Abstract. In this paper we discuss the asymptotic distribution of the singular values of the finite sections for a block Toeplitz operator $T(a)$ with continuous matrix-valued generating function a . Here the k -splitting property of the singular values of the finite sections $T_n(a) = P_n T(a) P_n$ plays an important role. We show that the singular values $s_{k+1}(T_n(a))$ tend to $\min(\|T(a)^+\|^{-1}, \|T(\tilde{a})^+\|^{-1})$ as $n \rightarrow \infty$, where k stands for the splitting number. Moreover, we estimate the speed of this convergence for a smooth generating function a .

Key words. Toeplitz operators, singular values, finite sections

AMS subject classifications. 15A18, 47B35, 65F20

DOI. 10.1137/S0895479804441973

1. Introduction. Let C denote the C^* -algebra of all continuous functions defined on the unit circle $\mathbb{T} := \{z \in \mathbb{C} : |z| = 1\}$ and let $C_{N \times N}$ be the C^* -algebra of all $N \times N$ matrices with entries from C . In this paper we analyze the behavior of the singular values of the finite sections $T_n(a)$ approximating the Toeplitz operator $T(a) \in \mathcal{L}(\ell_N^2)$, where $a \in C_{N \times N}$ and the operator $T(a)$ is supposed to be Fredholm. Our research is essentially based on new works of Böttcher, Roch, and Silbermann concerning the distribution of the singular values of the matrices $T_n(a)$ (see [1], [2], [4], [5]). In particular, we use that the set $\Sigma(T_n(a))$ of singular values of the finite sections of a Fredholm Toeplitz operator splits into two parts: one tending to zero as $n \rightarrow \infty$ and one being bounded below by a positive constant for all n . More precisely, the result reads as follows (see, e.g., [4]): If $T(a)$ is Fredholm, $a \in C_{N \times N}$, then exactly k singular values of $(T_n(a))$ tend to zero as $n \rightarrow \infty$ and $\liminf_{n \rightarrow \infty} s_{k+1}(T_n(a)) > 0$, where

$$k = \dim \ker T(a) + \dim \ker T(\tilde{a}), \quad \tilde{a}(t) = a(1/t), t \in \mathbb{T}.$$

On the one hand, this result gives a detailed information about the singular value behavior of $T_n(a)$. On the other hand, we get a way to determine the kernel dimension of $T(a)$, if we would be able to compute the part of $\Sigma(T_n(a))$ tending to zero (see [5]). In view of this observation, it makes sense to consider the following subjects: the convergence speed of $s_k(T_n(a))$ to zero and the behavior of $s_{k+1}(T_n(a))$. The first problem was already investigated in [6]. It was proved that the singular values $s_k(T_n(a))$ tend to zero with high speed if the generating function a is smooth. For instance, if a is rational, then $s_k(T_n(a)) = O(e^{-\rho n})$, with some $\rho > 0$.

In the present paper we deal with the second subject. We show (see Theorem 7.2) that the sequence $(s_{k+1}(T_n(a)))$ of the $(k+1)$ th singular values of finite sections $T_n(a)$ converges as $n \rightarrow \infty$ and the limit is equal to $\min(\|T(a)^+\|^{-1}, \|T(\tilde{a})^+\|^{-1})$, where $T(a)^+$ and $T(\tilde{a})^+$ are the Moore–Penrose inverses of the Fredholm Toeplitz operators

*Received by the editors March 8, 2004; accepted for publication (in revised form) by R. Nabben December 15, 2004; published electronically August 31, 2005. This work was supported by the Deutsche Forschungsgemeinschaft, DFG project SI 474/110-1.
<http://www.siam.org/journals/simax/27-1/44197.html>

†Department of Mathematics, Chemnitz University of Technology, D09107 Chemnitz, Germany (Alexander.Rogozhin@mathematik.tu-chemnitz.de).

$T(a)$ and $T(\tilde{a})$, respectively. Further, we estimate the speed of this convergence by the smoothness of the matrix function a (see Theorem 7.3). Besides, for the case where the Toeplitz operators $T(a)$ and $T(\tilde{a})$ are invertible, we obtain estimates of the speed with which $\|T_n^{-1}(a)\| = 1/s_1(T_n(a))$ converges to $\max(\|T(a)^{-1}\|, \|T(\tilde{a})^{-1}\|)$ (see Corollary 5.3).

2. Toeplitz operators and the algebra generated by their finite sections.

Let ℓ_N^2 denote the Hilbert space of all sequences $x : \mathbb{Z}_+ \rightarrow \mathbb{C}^N$, $\mathbb{Z}_+ = \{i \in \mathbb{Z} : i \geq 0\}$, such that

$$\|x\|_{\ell_N^2} := \left(\sum_{i=0}^{\infty} \|x_i\|_{\mathbb{C}^N}^2 \right)^{1/2} < \infty,$$

where $\|x_i\|_{\mathbb{C}^N}^2 = \|(x_i^1, x_i^2, \dots, x_i^N)\|_{\mathbb{C}^N}^2 = |x_i^1|^2 + |x_i^2|^2 + \dots + |x_i^N|^2$.

Let $a_k, k \in \mathbb{Z}$, be the $N \times N$ matrix $((a_{ij})_k)_{i,j=1}^N$ formed by the Fourier coefficients of a function $a = (a_{ij})_{i,j=1}^N \in C_{N \times N}$. The block Toeplitz operator $T(a) : \ell_N^2 \rightarrow \ell_N^2$ is defined by the matrix representation

$$T(a) = (a_{i-j})_{i,j=0}^{\infty} = \begin{pmatrix} a_0 & a_{-1} & a_{-2} & \cdots \\ a_1 & a_0 & a_{-1} & \cdots \\ a_2 & a_1 & a_0 & \cdots \\ \cdots & \cdots & \cdots & \cdots \end{pmatrix}.$$

For all generating functions $a \in C_{N \times N}$, the Toeplitz operator $T(a)$ is a linear bounded operator on ℓ_N^2 , i.e., $T(a) \in \mathcal{L}(\ell_N^2)$, and $\|T(a)\|_{\mathcal{L}(\ell_N^2)} = \|a\|_{\infty}$. Here $\mathcal{L}(H)$ stands for the C^* -algebra of all linear bounded operators acting on a Hilbert space H . Further, it is well known that the Toeplitz operator $T(a) \in \mathcal{L}(\ell_N^2), a \in C_{N \times N}$, is Fredholm if and only if $\det a(t) \neq 0$ for all $t \in \mathbb{T}$. More detailed information about the Toeplitz operators can be found for instance in the book [3].

Now, for $n \in \mathbb{N}$, we introduce the projectors P_n and Q_n on the Hilbert space ℓ_N^2 by

$$\begin{aligned} P_n x &= P_n(x_0, x_1, \dots, x_{n-1}, x_n, x_{n+1}, \dots) := (x_0, x_1, \dots, x_{n-1}, 0, 0, \dots), \\ Q_n x &:= (I - P_n)x = (0, 0, \dots, 0, x_n, x_{n+1}, \dots). \end{aligned}$$

The finite sections $T_n(a)$ of the Toeplitz operator $T(a)$ are defined by the truncated $nN \times nN$ matrices

$$T_n(a) := P_n T(a) P_n = (a_{i-j})_{i,j=0}^{n-1}, \quad n \in \mathbb{N}.$$

Further, let $W_n, n \in \mathbb{N}$, be the operators on ℓ_N^2 acting by the rule

$$W_n x = W_n(x_0, x_1, \dots, x_{n-1}, x_n, x_{n+1}, \dots) := (x_{n-1}, x_{n-2}, \dots, x_0, 0, 0, \dots).$$

Obviously we have the relations

$$W_n^2 = P_n, \quad W_n P_n = P_n W_n = W_n, \quad W_n T_n(a) W_n = T_n(\tilde{a}),$$

where $\tilde{a}(t) = a(1/t), t \in \mathbb{T}$.

Instead of the finite sections of the pure Toeplitz operators we will consider more general matrices, namely, matrices of the form

$$(2.1) \quad A_n = T_n(a) + P_n K P_n + W_n L W_n + C_n,$$

where $a \in C_{N \times N}$, where the sequence (C_n) of $nN \times nN$ matrices tends to zero in norm, i.e., $\|C_n P_n\|_{\mathcal{L}(\ell_N^2)} \rightarrow 0$, and where the operators $K, L \in \mathcal{K}(\ell_N^2)$. Here $\mathcal{K}(H)$ stands for the closed two-sided ideal of all compact operators acting on a Hilbert space H . Let us denote by $\mathbf{S}(C)$ the set of all operator sequences $(A_n)_{n \in \mathbb{N}}$ with $A_n \in \mathcal{L}(\text{im } P_n)$ of the form (2.1).

Notice that if $(A_n) \in \mathbf{S}(C)$, i.e., if $A_n = T_n(a) + P_n K P_n + W_n L W_n + C_n$, then there exist the strong limits

$$\begin{aligned} \text{s-lim } A_n &= T(a) + K =: \mathcal{W}_1(A_n), \\ \text{s-lim } W_n A_n W_n &= T(\tilde{a}) + L =: \mathcal{W}_2(A_n). \end{aligned}$$

Consideration of the matrices of the form (2.1) is motivated by the following important result.

THEOREM 2.1 (see, e.g., [1, Theorem 2.6]). *If $(A_n)_{n \in \mathbb{N}}$ is a sequence of matrices of the form (2.1) such that*

$$\lambda(A_n) + (B_n) = (\lambda A_n + B_n), \quad (A_n)(B_n) = (A_n B_n), \quad (A_n)^* = (A_n^*),$$

then the following conditions are equivalent:

$$\|(A_n)\|_{\mathbf{S}(C)} = \sup \left\{ \|A_n P_n\|_{\mathcal{L}(\ell_N^2)} : n = 1, 2, \dots \right\},$$

where $\mathbf{S}(C)$ is the C^* -algebra generated by the sequence $(A_n)_{n \in \mathbb{N}}$.

Next we turn to the singular values of the matrices of the form (2.1). The singular values of a linear bounded operator A acting on a Hilbert space H are the nonnegative square roots of the numbers in the spectrum of the operator A^*A . The set of the singular values of A is denoted by $\Sigma(A)$. If A_n is an $nN \times nN$ matrix, then $A_n^* A_n$ has nN eigenvalues and we can order the singular values of A_n so that

$$0 \leq s_1(A_n) \leq s_2(A_n) \leq \dots \leq s_{nN}(A_n) = \|A_n P_n\|_{\mathcal{L}(\ell_N^2)}.$$

DEFINITION 2.2. *A sequence $(A_n)_{n \in \mathbb{N}}$ of $nN \times nN$ matrices is called k -regular, where $k \in \mathbb{N}$, if $\|A_n P_n\|_{\mathcal{L}(\ell_N^2)} \rightarrow 0$ ($c_n \geq 0$) and $d > 0$ such that $\Sigma(A_n) \subset [0, c_n] \cup [d, \infty)$, $n \in \mathbb{N}$, and $k \leq \dim \ker A_n$ for all $n \in \mathbb{N}$.*

Now we are ready to formulate the result on the singular value behavior of sequences (A_n) from the C^* -algebra $\mathbf{S}(C)$.

THEOREM 2.3 (see, e.g., [4]). *If $(A_n)_{n \in \mathbb{N}}$ is a k -regular sequence from the C^* -algebra $\mathbf{S}(C)$ then the following conditions are equivalent:*

$$k = \dim \ker \mathcal{W}_1(A_n) + \dim \ker \mathcal{W}_2(A_n).$$

Finally, we mention several well-known relations (see, e.g., [3]) which will be used in the next sections.

$$(2.2) \quad \inf_{K \in \mathcal{K}(\ell_N^2)} \|T(a) + K\|_{\mathcal{L}(\ell_N^2)} = \|T(a)\|_{\mathcal{L}(\ell_N^2)} = \|a\|_\infty,$$

$$(2.3) \quad T(ab) = T(a)T(b) + H(a)H(\tilde{b}),$$

$$(2.4) \quad T_n(ab) = T_n(a)T_n(b) + P_n H(a)H(\tilde{b})P_n + W_n H(\tilde{a})H(b)W_n,$$

$$(2.5) \quad P_n T(a)Q_n T(b)P_n = W_n H(\tilde{a})H(b)W_n,$$

where the matrix functions $a, b \in C_{N \times N}$ and where the block Hankel operator $H(a) : \ell_N^2 \rightarrow \ell_N^2$ is defined by the following matrix representation:

$$H(a) = (a_{i+j+1})_{i,j=0}^\infty = \begin{pmatrix} a_1 & a_2 & a_3 & \cdots \\ a_2 & a_3 & a_4 & \cdots \\ a_3 & a_4 & a_5 & \cdots \\ \cdots & \cdots & \cdots & \cdots \end{pmatrix}.$$

Notice that, for all $a \in C_{N \times N}$, the Hankel operator is bounded ($\|H(a)\|_{\mathcal{L}(\ell_N^2)} \leq \|a\|_\infty$) and compact, i.e., $H(a) \in \mathcal{K}(\ell_N^2)$.

3. Asymptotic inverses. In what follows we use the same notation $\|\cdot\|$ for the norms of operators acting on ℓ_N^2 as for those acting on $\text{im } P_n$, i.e., we write $\|A\|$ and $\|A_n\|$ instead of $\|A\|_{\mathcal{L}(\ell_N^2)}$ and $\|A_n P_n\|_{\mathcal{L}(\ell_N^2)}$.

The next theorem gives a criteria for the invertibility of matrices of the form (2.1) for large n .

THEOREM 3.1 (see, e.g., [3]). *Let $(A_n)_{n \in \mathbb{N}}$ be a sequence of $nN \times nN$ matrices, $(A_n) \in \mathbf{S}(C)$.*

- (a) *If $n_0 \in \mathbb{N}$ and $\sup_{n \geq n_0} \|A_n^{-1}\| < \infty$, then $(A_n) \in \mathbf{S}(C)$.*
- (b) *If $\mathcal{W}_1(A_n) \rightarrow \mathcal{W}_1(A)$ and $\mathcal{W}_2(A_n) \rightarrow \mathcal{W}_2(A)$, then $(A_n) \in \mathbf{S}(C)$.*

Now we consider the structure of the inverses. Let $[\alpha]$ be the integral part of $\alpha \in (0, \infty)$.

THEOREM 3.2. *Let $(A_n) = (T_n(a) + P_n K P_n + W_n L W_n + C_n) \in \mathbf{S}(C)$. Then $\mathcal{W}_1(A_n) = T(a) + K$ and $\mathcal{W}_2(A_n) = T(\tilde{a}) + L$ for all n .*

$$A_n^{-1} = T_n(d) + P_n X P_n + W_n Y W_n + D_n,$$

$$d = a^{-1}, \quad X = [\mathcal{W}_1(A_n)]^{-1} - T(a^{-1}), \quad Y = [\mathcal{W}_2(A_n)]^{-1} - T(\tilde{a}^{-1}),$$

$$X = [\mathcal{W}_1(A_n)]^{-1} - T(a^{-1}), \quad Y = [\mathcal{W}_2(A_n)]^{-1} - T(\tilde{a}^{-1}),$$

$$(3.1) \quad \|D_n\| = O(\alpha_n)$$

Put

$$\alpha_n = \max (\|C_n\|, \|K Q_{[n/2]}\|, \|L Q_{[n/2]}\|, \|Q_{[n/2]} X\|, \|Q_{[n/2]} Y\|) \rightarrow 0.$$

By virtue of [1, Theorem 2.11], it remains to prove only the estimate (3.1). Put

$$B_n = T_n(d) + P_n X P_n + W_n Y W_n.$$

Then $D_n = A_n^{-1} - B_n = A_n^{-1}(P_n - A_n B_n)$, and since $\|A_n^{-1}\|$ is $O(1)$ by the previous theorem, it suffices to check that $\|P_n - A_n B_n\|$ is $O(\alpha_n)$.

Throughout the proof we will denote by (Z_n) all sequences with $\|Z_n\| = O(\alpha_n)$. We have

$$\begin{aligned} A_n B_n &= (T_n(a) + P_n K P_n + W_n L W_n + C_n)(T_n(d) + P_n X P_n + W_n Y W_n) \\ &= P_n(T(a) + K)P_n(T(d) + X)P_n + P_n(T(a) + K)W_n Y W_n \\ &\quad + W_n L W_n(T(d) + X)P_n + W_n L P_n Y W_n + Z_n. \end{aligned}$$

Now we treat separately each term of the sum. Using the equality $P_n + Q_n = I$ and relation (2.5) we obtain

$$\begin{aligned}
& P_n(T(a) + K)P_n(T(d) + X)P_n \\
&= P_n(T(a) + K)(T(a) + K)^{-1}P_n - P_n(T(a) + K)Q_n(T(d) + X)P_n \\
&= P_n - P_nT(a)Q_nT(d)P_n - P_nKQ_n(T(d) + X)P_n - P_nT(a)Q_nXP_n \\
&= P_n - W_nH(\tilde{a})H(d)W_n + Z_n.
\end{aligned}$$

Since $W_nP_{[n/2]} = Q_{[n/2]}W_nP_{[n/2]}$, we get

$$\begin{aligned}
& P_n(T(a) + K)W_nYW_n \\
&= W_nW_nT(a)W_nYW_n + P_nKW_n(P_{[n/2]} + Q_{[n/2]})YW_n \\
&= W_nT(\tilde{a})P_nYW_n + P_nKQ_{[n/2]}W_nP_{[n/2]}YW_n + P_nKW_nQ_{[n/2]}YW_n \\
&= W_nT(\tilde{a})P_nYW_n + Z_n.
\end{aligned}$$

Analogously,

$$\begin{aligned}
& W_nLW_n(T(d) + X)P_n \\
&= W_nLW_nT(d)W_nW_n + W_nLW_n(P_{[n/2]} + Q_{[n/2]})XP_n \\
&= W_nLP_nT(\tilde{d})W_n + Z_n.
\end{aligned}$$

Summing all terms and applying (2.3) we obtain

$$\begin{aligned}
A_nB_n &= P_n - W_nH(\tilde{a})H(d)W_n + W_nT(\tilde{a})P_nYW_n \\
&\quad + W_nLP_nT(\tilde{d})W_n + W_nLP_nYW_n + Z_n \\
&= P_n + W_n(T(\tilde{a}) + L)P_nYW_n + W_nLP_nT(\tilde{d})W_n \\
&\quad - W_nH(\tilde{a})H(d)W_n + Z_n \\
&= P_n + W_n(T(\tilde{a}) + L)[(T(\tilde{a}) + L)^{-1} - T(\tilde{d})]W_n \\
&\quad + W_nLP_nT(\tilde{d})W_n - W_nH(\tilde{a})H(d)W_n + Z_n \\
&= P_n + P_n - W_nT(\tilde{a})T(\tilde{d})W_n - W_nLT(\tilde{d})W_n \\
&\quad + W_nLP_nT(\tilde{d})W_n - W_nH(\tilde{a})H(d)W_n + Z_n \\
&= P_n + P_n - W_nT(\tilde{a}\tilde{d})W_n + W_nH(\tilde{a})H(d)W_n \\
&\quad - W_nLQ_nT(\tilde{d})W_n - W_nH(\tilde{a})H(d)W_n + Z_n \\
&= P_n + Z_n.
\end{aligned}$$

Hence

$$\|P_n - A_nB_n\| = O(\alpha_n).$$

Finally, the convergence of α_n to zero follows from the compactness of the operators K , L , X , and Y and from the strong convergence of the orthogonal projectors $Q_{[n/2]}$ to zero. $\square$

Note that this theorem provides an asymptotic inverse of a sequence $(A_n) \in \mathbf{S}(C)$, i.e., a sequence (B_n) of $nN \times nN$ matrices B_n such that $\|A_n^{-1} - B_n\| \rightarrow 0$ as $n \rightarrow \infty$. Now, for a continuous function $a \in C_{N \times N}$, we put

$$E_n(a) := \inf_{p \in \mathcal{R}_{N \times N}^n} \|a - p\|_\infty, \quad n \in \mathbb{Z}_+,$$

where $\mathcal{R}_{N \times N}^n$ is the set of all trigonometric polynomials p on $\mathbb{T}$ of the form

$$p(t) = \sum_{k=-n}^n p_k t^k, \quad p_k \in \mathbb{C}^{N \times N}.$$

It is well known (see, e.g., [10, Chapter 3.13]) that, for any $a \in C_{N \times N}$ and $n \in \mathbb{Z}_+$, there is a polynomial $p_n(a) \in \mathcal{R}_{N \times N}^n$ such that $E_n(a) = \|a - p_n(a)\|_\infty$.

Further, let $C^\alpha, \alpha > 0$, denote the set of all continuous functions on $\mathbb{T}$ which are $[\alpha]$ times continuously differentiable and whose $[\alpha]$ th derivative satisfies a Hölder condition with the exponent $\{\alpha\} = \alpha - [\alpha]$ ($0 \leq \{\alpha\} < 1$), and let $C_{N \times N}^\alpha$ be the set of all $N \times N$ matrices with entries from C^α . Notice that if $a \in C_{N \times N}^\alpha$, then $E_n(a) = O(n^{-\alpha}), n \in \mathbb{N}$ (see, e.g., [10, Chapter 3.13]).

The following result will be used in the next sections to estimate the convergence speed of the singular values.

COROLLARY 3.3. *Let $a, b \in C_{N \times N}$, $\lambda \in \mathbb{C}$, $A_n = T_n(a)T_n(b) + \lambda P_n$. Then*

$$(T(a)T(b) + \lambda I)^{-1} - (T(\tilde{a})T(\tilde{b}) + \lambda I)^{-1} = O(n^{-\alpha}),$$

$$A_n^{-1} = T_n(d) + P_n X P_n + W_n Y W_n + D_n,$$

$$d = (ab + \lambda E)^{-1}, \quad \|D_n\| \rightarrow 0, \quad X = (T(a)T(b) + \lambda I)^{-1} - T(d), \quad Y = (T(\tilde{a})T(\tilde{b}) + \lambda I)^{-1} - T(\tilde{d}).$$

Applying (2.4), we rewrite the matrices A_n in form (2.1):

$$\begin{aligned} A_n &= T_n(a)T_n(b) + \lambda P_n \\ &= T_n(ab) - P_n H(a)H(\tilde{b})P_n - W_n H(\tilde{a})H(b)W_n + \lambda P_n \\ &= T_n(ab + \lambda E) - P_n H(a)H(\tilde{b})P_n - W_n H(\tilde{a})H(b)W_n. \end{aligned}$$

Thus we have $K = -H(a)H(\tilde{b})$, $L = -H(\tilde{a})H(b)$, $C_n = 0$, and in view of Theorem 3.2 it remains to estimate the numbers α_n from the degree of smoothness of the functions a and b .

First, we assume that the functions $a, b \in C_{N \times N}^\alpha$. For any continuous function $c \in C_{N \times N}$, we have

$$\begin{aligned} (3.2) \quad \|H(c)Q_{[n/2]}\| &= \|H(c - p_{[n/2]}(c) + p_{[n/2]}(c))Q_{[n/2]}\| \\ &= \|H(c - p_{[n/2]}(c))Q_{[n/2]}\| \leq \|c - p_{[n/2]}(c)\|_\infty = E_{[n/2]}(c). \end{aligned}$$

This implies that $\|KQ_{[n/2]}\|, \|LQ_{[n/2]}\| = O(n^{-\alpha})$, as $E_{[n/2]}(b) = E_{[n/2]}(\tilde{b}) = O(n^{-\alpha})$. Now we consider the operators X and Y . Taking into account relation (2.3) we get

$$\begin{aligned} X &= (T(a)T(b) + \lambda I)^{-1} - T(d) = [I - T(d)(T(a)T(b) + \lambda I)](T(a)T(b) + \lambda I)^{-1} \\ &= [I - T(d)T(a)T(b) - \lambda T(d)](T(a)T(b) + \lambda I)^{-1} \\ &= [I - T(da)T(b) + H(d)H(\tilde{a})T(b) - \lambda T(d)](T(a)T(b) + \lambda I)^{-1} \\ &= [I - T(d(ab + \lambda E)) + H(da)H(\tilde{b}) + H(d)H(\tilde{a})T(b)](T(a)T(b) + \lambda I)^{-1} \\ &= [H(da)H(\tilde{b}) + H(d)H(\tilde{a})T(b)](T(a)T(b) + \lambda I)^{-1}, \end{aligned}$$

and, analogously,

$$Y = [H(\tilde{d}\tilde{a})H(b) + H(\tilde{d})H(a)T(\tilde{b})](T(\tilde{a})T(\tilde{b}) + \lambda I)^{-1}.$$

Further, similarly to (3.2), one can show that $\|Q_{[n/2]}H(c)\| \leq E_{[n/2]}(c)$ for any continuous function $c \in C_{N \times N}$. Hence $\|Q_{[n/2]}X\|, \|Q_{[n/2]}Y\| = O(n^{-\alpha})$, as the matrix functions d and da belong to $C_{N \times N}^\alpha$ too. Thus

$$\alpha_n = \max(\|KQ_{[n/2]}\|, \|LQ_{[n/2]}\|, \|Q_{[n/2]}X\|, \|Q_{[n/2]}Y\|) = O(n^{-\alpha}).$$

The case of rational functions a and b can be treated in just the same way. We remark only that if a function $c \in C_{N \times N}$ is rational, then $E_n(c) = O(e^{-\mu n})$ with some $\mu > 0$, since the Fourier coefficients of rational functions decay exponentially. $\square$

4. Norms of the finite sections of perturbed Toeplitz band matrices.

Let a be a trigonometric polynomial of the form

$$a(t) = \sum_{k=-r}^r a_k t^k, \quad a_k \in \mathbb{C}^{N \times N}, r \in \mathbb{N},$$

and let X and Y be infinite matrices all entries of which are zero outside the upper left $rN \times rN$ block, that is,

$$P_r X P_r = X, \quad P_r Y P_r = Y.$$

In this section we consider the norms of the band matrices

$$A_n = T_n(a) + P_n X P_n + W_n Y W_n.$$

Note that, for scalar valued functions a , i.e., for the particular case $N = 1$, all results of this section were already obtained in [1]. Here we generalize these results to the case of matrix valued functions a .

Obviously, the band matrices A_n are of the form (2.1), i.e., the sequence (A_n) belongs to the C^* -algebra $\mathbf{S}(C)$. Therefore we can use the following theorem.

THEOREM 4.1 (see, e.g., [1, Theorem 2.13]). $\lim_{n \rightarrow \infty} \|A_n\| = \max(\mathcal{W}_1(A_n), \mathcal{W}_2(A_n)).$

$$\lim_{n \rightarrow \infty} \|A_n\| = \max(\mathcal{W}_1(A_n), \mathcal{W}_2(A_n)).$$

This theorem shows that the norms $\|A_n\|$ of the band matrices A_n converge to

$$M := \max(\|T(a) + X\|, \|T(\tilde{a}) + Y\|).$$

The goal of the section is to estimate the speed of this convergence. We start with an auxiliary result.

LEMMA 4.2. $1 \leq r \leq l \leq n/3.$

$$1 \leq r \leq l \leq n/3.$$

Let $x \in \text{im } P_n$. Then $\|x\|_{\ell_N^2}^2 = \sum_{j=0}^{n-1} \|x_j\|_{\mathbb{C}^N}^2 = 1$, and m is a positive integer.

$$l+r \leq m \leq 3l-r, \quad \|P_{m+r}Q_{m-r}x\|_{\ell_N^2}^2 \leq \left[\frac{l}{r}\right]^{-1}.$$

Put $d = [l/r]$. We obtain $l + 2dr \leq 3l$ and

$$1 = \|x\|_{\ell_N^2}^2 \geq \|P_{l+2dr}Q_l x\|_{\ell_N^2}^2 = \sum_{k=0}^{d-1} \|P_{l+2(k+1)r}Q_{l+2kr}x\|_{\ell_N^2}^2.$$

As there are d terms on the right-hand side, at least one of them does not exceed $1/d$. Hence, there exist a $k_0, 0 \leq k_0 \leq d-1$, such that $\|P_{l+2(k_0+1)r}Q_{l+2k_0r}x\|_{\ell_N^2}^2 \leq 1/d$. Thus the assertion follows with $m = l + 2k_0r + r$. $\square$

Now we denote by M_0 the norm of the Toeplitz operator $T(a)$. Note that in view of (2.2) we always have

$$M \geq \|T(a) + X\| \geq \inf_{K \in \mathcal{K}(\ell_N^2)} \|T(a) + K\| = \|T(a)\| = M_0.$$

The following two results provide a lower estimate for the norm of the band matrix A_n .

LEMMA 4.3. Put $n \geq 20r$.

$$M_0 \left(1 - \frac{20r}{n}\right) \leq \|T_n(a)\|.$$

Put $\varepsilon_n = M_0 - \|T_n(a)\|$. Clearly, $\varepsilon_n \geq 0$ and ε_n converges monotonously to zero. We have to show that

$$\varepsilon_n \leq \frac{20r}{n} M_0.$$

Choose $x \in \text{im } P_n$ so that

$$(4.1) \quad \|x\|_{\ell_N^2} = 1 \quad \text{and} \quad \|T_n(a)x\|_{\ell_N^2} = \|T_n(a)\| = M_0 - \varepsilon_n.$$

Set $l = [n/4]$. By Lemma 4.2, there exists a natural number m such that

$$l + r \leq m \leq 3l - r \quad \text{and} \quad \|P_{m+r}Q_{m-r}x\|_{\ell_N^2}^2 \leq \left[\frac{l}{r}\right]^{-1}.$$

Since $T_n(a)$ is a band matrix with at most $(2r+1)N$ nonzero diagonals, we get

$$\begin{aligned} \|T_n(a)x\|_{\ell_N^2}^2 &= \|P_n(P_m + Q_m)T(a)P_n x\|_{\ell_N^2}^2 \\ &= \|P_m T(a)P_n x\|_{\ell_N^2}^2 + \|P_n Q_m T(a)P_n x\|_{\ell_N^2}^2 \\ &= \|P_m T(a)P_{m+r}x\|_{\ell_N^2}^2 + \|P_n Q_m T(a)Q_{m-r}P_n x\|_{\ell_N^2}^2 \\ &= \|P_m P_{m+r} T(a)P_{m+r}x\|_{\ell_N^2}^2 + \|Q_m P_n Q_{m-r} T(a)Q_{m-r}P_n x\|_{\ell_N^2}^2 \\ &\leq \|T_{m+r}(a)\|^2 \|P_{m+r}x\|_{\ell_N^2}^2 + \|T_{n-m+r}(a)\|^2 \|P_n Q_{m-r}x\|_{\ell_N^2}^2 \\ &= (M_0 - \varepsilon_{m+r})^2 \|P_{m+r}x\|_{\ell_N^2}^2 + (M_0 - \varepsilon_{n-m+r})^2 \|P_n Q_{m-r}x\|_{\ell_N^2}^2. \end{aligned}$$

Further, taking into account that $m+r \leq n-l$ and $n-m+r \leq n-l$ we obtain

$$\begin{aligned} \|T_n(a)x\|_{\ell_N^2}^2 &\leq (M_0 - \varepsilon_{n-l})^2 (\|P_{m+r}x\|_{\ell_N^2}^2 + \|P_n Q_{m-r}x\|_{\ell_N^2}^2) \\ &= (M_0 - \varepsilon_{n-l})^2 (1 + \|P_{m+r}Q_{m-r}x\|_{\ell_N^2}^2) \leq (M_0 - \varepsilon_{n-l})^2 (1 + [l/r]^{-1}). \end{aligned}$$

Thus, by (4.1),

$$(M_0 - \varepsilon_n)^2 \leq (M_0 - \varepsilon_{n-l})^2 + M_0^2[l/r]^{-1}$$

or, equivalently,

$$(4.2) \quad \left(1 - \frac{\varepsilon_n}{M_0}\right)^2 \leq \left(1 - \frac{\varepsilon_{n-l}}{M_0}\right)^2 + \left[\frac{l}{r}\right]^{-1}.$$

Recall that $l = [n/4]$, and put $n_1 = [4n/3]$. Straightforward checks show that

$$n_1 - \left[\frac{n_1}{4}\right] = n \quad \text{and} \quad \left[\frac{[n_1/4]}{r}\right] \geq \frac{n}{4r}.$$

Replacing n by n_1 in (4.2) we arrive at the inequality

$$(4.3) \quad \left(1 - \frac{\varepsilon_{n_1}}{M_0}\right)^2 \leq \left(1 - \frac{\varepsilon_n}{M_0}\right)^2 + \frac{4r}{n}.$$

Now replace the n in (4.3) consequently by $n_1 = [4n/3]$, $n_2 = [4n_1/3]$, ..., $n_j = [4n_{j-1}/3]$ to get

$$\left(1 - \frac{\varepsilon_{n_j}}{M_0}\right)^2 \leq \left(1 - \frac{\varepsilon_n}{M_0}\right)^2 + \frac{4r}{n} + \sum_{i=1}^{j-1} \frac{4r}{n_i}.$$

Passing to the limit $j \rightarrow \infty$ we obtain

$$1 - \left(1 - \frac{\varepsilon_n}{M_0}\right)^2 \leq \frac{4r}{n} + \sum_{i=1}^{\infty} \frac{4r}{n_i}.$$

As $n/n_1 \leq 4/5$ and $n_i/n_{i+1} \leq 4/5$, for all $i \in \mathbb{N}$, it follows that

$$1 - \left(1 - \frac{\varepsilon_n}{M_0}\right)^2 \leq \frac{4r}{n} + \frac{4r}{n} \sum_{i=1}^{\infty} \left(\frac{4}{5}\right)^i = \frac{20r}{n}.$$

Finally, since $s \geq s^2$, $0 \leq s \leq 1$, we obtain that

$$\frac{\varepsilon_n}{M_0} = 1 - \left(1 - \frac{\varepsilon_n}{M_0}\right) \leq 1 - \left(1 - \frac{\varepsilon_n}{M_0}\right)^2 \leq \frac{20r}{n}. \quad \square$$

COROLLARY 4.4. $\dots$, $n \geq 41r$, $\dots$

$$M_0 \left(1 - \frac{41r}{n}\right) \leq \|A_n\|.$$

$\dots$ By the previous lemma, there is an $x \in \text{im } P_{[n/2]}$ such that $\|x\|_{\ell_N^2} = 1$ and

$$M_0 \left(1 - \frac{20r}{[n/2]}\right) \leq \|T_{[n/2]}(a)x\|_{\ell_N^2}.$$

Let $l = [n/4]$, and define $y \in \text{im } P_n$ by

$$y = (\underbrace{0, \dots, 0}_{lN}, x, \underbrace{0, \dots, 0}_{l'N}).$$

This implies that $l' \geq l \geq r$ and $A_n y = T_n(a)y$, whence

$$\begin{aligned} \|A_n\|^2 &\geq \|A_n x\|_{\ell_N^2}^2 = \|T_n(a)y\|_{\ell_N^2}^2 \geq \|T_{[n/2]}(a)x\|_{\ell_N^2}^2 \\ &\geq M_0^2 \left(1 - \frac{20r}{[n/2]}\right)^2 \geq M_0^2 \left(1 - \frac{40r}{n-1}\right)^2 \geq M_0^2 \left(1 - \frac{41r}{n}\right)^2. \end{aligned}$$

The last estimate follows from the assumption that $n \geq 41r \geq 41$. $\square$

Now we present an upper estimate for the norm of the band matrix A_n .

LEMMA 4.5. $\dots$ $n \geq 20r$, $\dots$

$$\|A_n\| \leq M \left(1 + \frac{3r}{n}\right).$$

$\dots$ Set $l = [n/4]$, and choose $x \in \text{im } P_n$ so that $\|x\|_{\ell_N^2} = 1$ and $\|A_n x\|_{\ell_N^2} = \|A_n\|$. By Lemma 4.2, there exists a natural number m such that

$$l + r \leq m \leq 3l - r \quad \text{and} \quad \|P_{m+r} Q_{m-r} x\|_{\ell_N^2}^2 \leq \left[\frac{l}{r}\right]^{-1}.$$

Proceeding as in the proof of Lemma 4.3, we get

$$\begin{aligned} \|A_n x\|_{\ell_N^2}^2 &= \|P_m A_n P_{m+r} x\|_{\ell_N^2}^2 + \|Q_m A_n Q_{m-r} x\|_{\ell_N^2}^2 \\ &= \|P_m [(T(a) + X)P_n + W_n P_r Y P_r W_n] P_{m+r} x\|_{\ell_N^2}^2 \\ &\quad + \|Q_m [W_n (T(\tilde{a}) + Y)W_n + P_r X P_r] Q_{m-r} x\|_{\ell_N^2}^2 \\ &= \|[P_m (T(a) + X)P_n + P_m Q_{n-r} W_n P_r Y P_r W_n] P_{m+r} x\|_{\ell_N^2}^2 \\ &\quad + \|[Q_m W_n (T(\tilde{a}) + Y)W_n + Q_m P_r X P_r] Q_{m-r} x\|_{\ell_N^2}^2. \end{aligned}$$

Since $n - r > 3l - r \geq m$ and $m \geq l + r > r$, we obtain

$$\|A_n x\|_{\ell_N^2}^2 \leq \|T(a) + X\|^2 \|P_{m+r} x\|_{\ell_N^2}^2 + \|T(\tilde{a}) + Y\|^2 \|Q_{m-r} x\|_{\ell_N^2}^2 \leq M^2 \left(1 + \left[\frac{l}{r}\right]^{-1}\right).$$

As $[l/r] \geq n/(5r)$, it results that

$$\|A_n\|^2 \leq M^2 \left(1 + \frac{5r}{n}\right) \leq M^2 \left(1 + \frac{3r}{n}\right)^2. \quad \square$$

If $M = M_0$, this lemma together with Corollary 4.4 implies the following estimate for the convergence speed of $\|A_n\|$ to M .

THEOREM 4.6 (cf. [1, Theorem 4.3]). $\dots$ $n \geq 41r$, $\dots$ $M = M_0$, $\dots$

$$(4.4) \quad \left| \|A_n\| - M \right| \leq \frac{41Mr}{n}.$$

Now we turn to the case $M > M_0$. Put $A = T(a) + X$, and recall that $\|A\| \geq \|T(a)\| = M_0$, by (2.2).

LEMMA 4.7. $\dots$ $\|A\| > M_0$, $\dots$ $x \in \ell_N^2$, $\dots$

$$(4.5) \quad \|x\|_{\ell_N^2} = 1, \quad \|Ax\|_{\ell_N^2} = \|A\|, \quad \dots \quad \|Q_{2r+l} x\|_{\ell_N^2} \leq \left(\frac{M_0}{\|A\|}\right)^{l/(2r)}$$

$\dots$ $l \in \mathbb{Z}_+$.

Let $N := \|A\|$, and pick a natural number $m \geq r$. There is a sequence $x_k \in \ell_N^2$ such that

$$(4.6) \quad \|x_k\|_{\ell_N^2} = 1, \quad \|Ax_k\|_{\ell_N^2}^2 = N^2 - \delta_k,$$

where δ_k goes monotonously to zero. Again we see as in the proof of Lemma 4.3 that

$$Ax_k = P_m Ax_k + Q_m(T(a) + P_r X P_r)x_k = P_m A P_{m+r} x_k + Q_m T(a) Q_{m-r} x_k,$$

whence

$$N^2 - \delta_k \leq N^2(1 - \|Q_{m+r} x_k\|_{\ell_N^2}^2) + M_0^2 \|Q_{m-r} x_k\|_{\ell_N^2}^2.$$

On defining

$$O_n^k := \|Q_n x_k\|_{\ell_N^2}^2, \quad \gamma_k := \frac{\delta_k}{N^2}, \quad q := \frac{M_0^2}{N^2},$$

we can write the last inequality in the form

$$(4.7) \quad O_{m+r}^k \leq q O_{m-r}^k + \gamma_k, \quad m \geq r.$$

Since $O_n^k \leq 1$, we conclude from (4.7) that $O_{2r+v}^k \leq q + \gamma_k$, $v \in \mathbb{Z}_+$. This and (4.7) imply, for all $j, v \in \mathbb{Z}_+$,

$$\begin{aligned} O_{2jr+2r+v}^k &\leq q O_{2(j-1)r+2r+v}^k + \gamma_k \leq \dots \leq q^j O_{2r+v}^k + (q^{j-1} + \dots + 1)\gamma_k \\ &\leq q^{j+1} + (q^j + \dots + 1)\gamma_k \leq q^{j+1} + \frac{\gamma_k}{1-q}. \end{aligned}$$

Let $s = q^{1/(2r)}$ and $l = 2jr + v$. Then we obtain, for $j \geq 0$, $0 \leq v < 2r$,

$$O_{2r+l}^k = O_{2jr+2r+v}^k \leq s^{2jr+2r} + \frac{\gamma_k}{1-q} \leq s^l + \frac{\gamma_k}{1-q}.$$

The last estimate follows from the inequality $s \leq 1$. It results that

$$(4.8) \quad O_{2r+l}^k \leq s^l + \frac{\gamma_k}{1-q} \quad \forall l \in \mathbb{Z}_+.$$

We now show that $\{x_k\}_{k=1}^\infty$ contains a Cauchy subsequence. Given $\varepsilon > 0$, there is a natural number $n \geq 2r$ such that $s^{n-2r} < \varepsilon^2/8$. On passing to a subsequence if necessary, we can assume that the sequence $\{P_n x_k\}_{k=1}^\infty$ is a Cauchy sequence in $\text{im } P_n$. Hence there is a natural number p such that

$$\|P_n x_{k_1} - P_n x_{k_2}\|_{\ell_N^2}^2 < \frac{\varepsilon^2}{4} \quad \text{and} \quad \frac{\gamma_{k_1} + \gamma_{k_2}}{1-q} < \frac{\varepsilon^2}{8}$$

whenever $k_1, k_2 \geq p$. From (4.8) we obtain

$$\begin{aligned} \|x_{k_1} - x_{k_2}\|_{\ell_N^2}^2 &= \|P_n(x_{k_1} - x_{k_2})\|_{\ell_N^2}^2 + \|Q_n(x_{k_1} - x_{k_2})\|_{\ell_N^2}^2 \\ &\leq \varepsilon^2/4 + 2\|Q_n x_{k_1}\|_{\ell_N^2}^2 + 2\|Q_n x_{k_2}\|_{\ell_N^2}^2 = \varepsilon^2/4 + 2O_n^{k_1} + 2O_n^{k_2} \\ &\leq \varepsilon^2/4 + 4s^{n-2r} + 2\frac{\gamma_{k_1} + \gamma_{k_2}}{1-q} \leq \varepsilon^2/4 + \varepsilon^2/2 + \varepsilon^2/4 = \varepsilon^2. \end{aligned}$$

This shows that $\{x_k\}_{k=1}^\infty$ contains a Cauchy subsequence in ℓ_N^2 . Let $x \in \ell_N^2$ be the limit of the Cauchy subsequence of $\{x_k\}_{k=1}^\infty$. Passing to the limit $k \rightarrow \infty$ in (4.6) and in (4.8), we obtain

$$\|x\|_{\ell_N^2} = 1, \quad \|Ax\|_{\ell_N^2} = \|A\|, \quad \text{and} \quad \|Q_{2r+l}x\|_{\ell_N^2}^2 \leq s^l = \left(\frac{M_0}{N}\right)^{(2l)/(2r)}, \quad l \in \mathbb{Z}_+. \quad \square$$

The next two lemmas provide lower and upper estimates for the norm of the band matrix A_n in the case $M > M_0$.

LEMMA 4.8. *Let $n \geq 5r$, $M > M_0$, and*

$$M(1 - c\rho^n) \leq \|A_n\|,$$

$$c = (M/M_0)^{(4r+1)/(4r)}, \quad \rho = (M_0/M)^{1/(4r)}.$$

For the sake of definiteness, we suppose that $M = \|T(a) + X\|$ (because $\|A_n\| = \|W_n A_n W_n\|$, the case $M = \|T(\tilde{a}) + Y\|$ can be reduced to the case we consider). Let x satisfy (4.5). Put $m = [n/2]$ and $y_n = P_m x$. Then

$$\begin{aligned} \|A_n y_n\|_{\ell_N^2} &= \|P_n(T(a) + X)P_m x + W_n P_r Y P_r W_n P_m x\|_{\ell_N^2} \\ &= \|(T(a) + X)P_m x - Q_n(T(a) + X)P_m x\|_{\ell_N^2} \\ &= \|(T(a) + X)x - (T(a) + X)Q_m x - Q_n(T(a) + X)Q_{n-r}P_m x\|_{\ell_N^2} \\ &= \|(T(a) + X)x - (T(a) + X)Q_m x\|_{\ell_N^2} \geq M - M\|Q_m x\|_{\ell_N^2} \\ &\geq M - M(M_0/M)^{(m-2r)/(2r)} \geq M - M(M_0/M)^{(n-1-4r)/(4r)}. \end{aligned}$$

Since $\|y_n\|_{\ell_N^2} \leq 1$, we finally obtain that

$$\|A_n\| \geq \|A_n y_n\|_{\ell_N^2} \geq M - M c \rho^n. \quad \square$$

LEMMA 4.9. *Let $n \geq 8r$, $M > M_0$, and*

$$\|A_n\| \leq M(1 + c\rho^n),$$

$$c = (M/M_0)^{(4r+1)/(4r)}, \quad \rho = (M_0/M)^{1/(4r)}.$$

Since $\|A_n\| \rightarrow M$, there are vectors $x_n \in \text{im } P_n$ such that

$$(4.9) \quad \|x_n\|_{\ell_N^2} = 1, \quad \|A_n x_n\|_{\ell_N^2} = \|A_n\|, \quad \|A_n x_n\|_{\ell_N^2}^2 = M^2 + \varepsilon_n,$$

where $\varepsilon_n \rightarrow 0$ as $n \rightarrow \infty$. Again set $l = [n/4]$, and note that $l > r$. For $l < m < 2l$, we get

$$\begin{aligned} \|A_n x_n\|_{\ell_N^2}^2 &= \|P_m A_n P_{m+r} x_n\|_{\ell_N^2}^2 + \|Q_m P_{n-m} A_n x_n\|_{\ell_N^2}^2 + \|Q_{n-m} A_n Q_{n-m-r} x_n\|_{\ell_N^2}^2 \\ &= \|P_m(T(a) + X)P_{m+r} x_n\|_{\ell_N^2}^2 + \|Q_m P_{n-m} T(a) Q_{m-r} P_{n-m+r} x_n\|_{\ell_N^2}^2 \\ &\quad + \|Q_{n-m} W_n(T(\tilde{a}) + Y)Q_{n-m-r} x_n\|_{\ell_N^2}^2 \\ &\leq M^2(\|P_{m+r} x_n\|_{\ell_N^2}^2 + \|Q_{n-m-r} x_n\|_{\ell_N^2}^2) + M_0^2 \|Q_{m-r} P_{n-m+r} x_n\|_{\ell_N^2}^2. \end{aligned}$$

Letting $O_k^n := \|Q_k P_{n-k} x_n\|_{\ell_N^2}^2$, we therefore obtain

$$M^2 + \varepsilon_n \leq M^2(1 - O_{m+r}^n) + M_0^2 O_{m-r}^n$$

or, equivalently,

$$O_{m+r}^n \leq q O_{m-r}^n - \gamma_n,$$

with $q := (M_0/M)^2$ and $\gamma_n := \varepsilon_n/M^2$. This is of the form (4.7). Consequently, we have

$$O_{2r+j}^n \leq \begin{cases} s^j - \gamma_n, & \gamma_n \geq 0, \\ s^j - \frac{\gamma_n}{1-q}, & \gamma_n < 0, \end{cases} \quad \text{with} \quad s := q^{1/(2r)},$$

provided O_{2r+j}^n is well defined, i.e., for all j satisfying $2r + j \leq n - (2r + j) - 1$ or, equivalently, for all $j \leq [(n - 4r - 1)/2]$. As $O_{2r+j}^n \geq 0$ we get, for $\gamma_n \geq 0$ ($\varepsilon_n \geq 0$),

$$(4.10) \quad \gamma_n \leq s^{[(n-4r-1)/2]} \leq q^{(n-4r-2)/(4r)}.$$

If $\varepsilon_n < 0$, then $\gamma_n < 0$, and hence (4.10) is also satisfied. Finally, we obtain that

$$\begin{aligned} \|A_n\|^2 &= M^2 + M^2\gamma_n \leq M^2(1 + q^{(n-4r-3)/(8r)}q^{(n-4r-1)/(8r)}) \\ &\leq M^2(1 + q^{(n-4r-1)/(8r)})^2 = M^2(1 + (M_0/M)^{(n-4r-1)/(4r)})^2. \quad \square \end{aligned}$$

Lemmas 4.8 and 4.9 imply the following estimate for the convergence speed of $\|A_n\|$ to M in the case $M > M_0$.

THEOREM 4.10 (cf. [1, Theorem 4.4]). *Let $n \geq 8r$ and $M > M_0$. Then*

$$(4.11) \quad |\|A_n\| - M| \leq cM\rho^n,$$

where $c = (M/M_0)^{(4r+1)/(4r)}$, $\rho = (M_0/M)^{1/(4r)}$.

Thus, we have obtained that if $M > M_0$, the norms $\|A_n\|$ converge much faster to M . Note there are indeed band matrices A_n for which we do not have the exponential convergence (see [1]). Consequently, the gap between estimates (4.4) and (4.11) is essential and cannot be removed.

5. Norms of the inverses. Now we return to the inverses of the matrices A_n of form (2.1). From Theorem 3.2 we know that if the operators $\mathcal{W}_1(A_n)$ and $\mathcal{W}_2(A_n)$ are invertible then, for all sufficiently large n ,

$$A_n^{-1} = T_n(a^{-1}) + P_n X P_n + W_n Y W_n + D_n,$$

where the operators X and Y are compact and $\|D_n\| \rightarrow 0$ as $n \rightarrow \infty$. Further, Theorem 4.1 implies that the norms $\|A_n^{-1}\|$ converge to

$$M := \max(\|T(a^{-1}) + X\|, \|T(\widetilde{a^{-1}}) + Y\|) = \max(\|[\mathcal{W}_1(A_n)]^{-1}\|, \|[\mathcal{W}_2(A_n)]^{-1}\|).$$

Using the results of the previous section we can estimate the speed of this convergence.

THEOREM 5.1. *Let $A_n = (T_n(a) + P_n K P_n + W_n L W_n + C_n) \in \mathbf{S}(C)$ with $r(n) \rightarrow \infty$, $r(n)/n \rightarrow 0$ as $n \rightarrow \infty$. Then*

$$|\|A_n^{-1}\| - M| = \begin{cases} O(e^{-\delta n/r(n)}) + O(\beta_n) & \text{if } M > \|T(a^{-1})\|, \\ O\left(\frac{r(n)}{n}\right) + O(\beta_n) & \text{if } M = \|T(a^{-1})\|, \end{cases}$$

where $\delta > 0$ and

$$\beta_n = \max(E_{r(n)}(a^{-1}), \|C_n\|, \|KQ_{[n/2]}\|, \|LQ_{[n/2]}\|, \|XQ_{r(n)}\|, \|YQ_{r(n)}\|, \|Q_{r(n)}X\|, \|Q_{r(n)}Y\|) \rightarrow 0.$$

Put $X_n = P_{r(n)}XP_{r(n)}$ and $Y_n = P_{r(n)}YP_{r(n)}$, and let $d_n \in \mathcal{R}_{N \times N}^{r(n)}$ be a trigonometric polynomial so that $E_{r(n)}(a^{-1}) = \|a^{-1} - d_n\|_\infty$. We denote by B_n the band matrix

$$B_n = T_n(d_n) + P_n X_n P_n + W_n Y_n W_n.$$

Since $n/r(n) \rightarrow \infty$ as $n \rightarrow \infty$, we have $\|D_n\| = O(\beta_n)$ by Theorem 3.2. Hence,

$$\begin{aligned} \|A_n^{-1} - B_n\| &\leq \|T_n(d - d_n)\| + \|X - X_n\| + \|Y - Y_n\| + \|D_n\| \leq \|d - d_n\|_\infty \\ &\quad + \|Q_{r(n)}X\| + \|XQ_{r(n)}\| + \|Q_{r(n)}Y\| + \|YQ_{r(n)}\| + \|D_n\| = O(\beta_n). \end{aligned}$$

Further, let

$$\begin{aligned} M(n) &:= \max(\|T(d_n) + X_n\|, \|T(\widetilde{d_n}) + Y_n\|), \\ M_0 &= \|T(a^{-1})\|, \quad M_0(n) = \|T(d_n)\|. \end{aligned}$$

We have that the number $|M - M(n)|$ is at most

$$\begin{aligned} &\max(\| \|T(a^{-1}) + X\| - \|T(d_n) + X_n\| \|, \| \|T(\widetilde{a^{-1}}) + Y\| - \|T(\widetilde{d_n}) + Y_n\| \|) \\ &\leq \max(\| \|T(a^{-1}) + X - T(d_n) - X_n\| \|, \| \|T(\widetilde{a^{-1}}) + Y - T(\widetilde{d_n}) - Y_n\| \|) = O(\beta_n), \end{aligned}$$

and

$$|M_0 - M_0(n)| = \| \|T(a^{-1})\| - \|T(d_n)\| \| \leq \|T(a^{-1} - d_n)\| = O(\beta_n).$$

In summary, we obtain that

$$\begin{aligned} \| \|A_n^{-1}\| - M \| &\leq \| \|A_n^{-1}\| - \|B_n\| \| + \| \|B_n\| - M(n) \| + |M(n) - M| \\ &\leq \| \|B_n\| - M(n) \| + \|A_n^{-1} - B_n\| + |M(n) - M| = \| \|B_n\| - M(n) \| + O(\beta_n) \end{aligned}$$

and that

$$(5.1) \quad M(n) = M + O(\beta_n), \quad M_0(n) = M_0 + O(\beta_n).$$

Now suppose $M > M_0$. From the assumption $r(n)/n \rightarrow 0$ and from (5.1), it follows that, for all sufficiently large n , $n \geq 8r(n)$ and $M(n) > M_0(n)$. Then we can use Theorem 4.10 to get

$$\| \|B_n\| - M(n) \| \leq M(n) \left(\frac{M(n)}{M_0(n)} \right)^{4r(n)+1/4r(n)} \left(\frac{M_0(n)}{M(n)} \right)^{n/4r(n)}.$$

Choose an $\varepsilon > 0$ such that $0 < (M_0 + \varepsilon)/(M - \varepsilon) < 1$. If n is large enough, we obtain

$$\| \|B_n\| - M(n) \| \leq (M + \varepsilon) \left(\frac{M + \varepsilon}{M_0 - \varepsilon} \right)^2 \left(\frac{M_0 + \varepsilon}{M - \varepsilon} \right)^{\frac{n}{4r(n)}} = O\left(e^{-\delta n/r(n)}\right),$$

where

$$\delta = -\log \left(\frac{M_0 + \varepsilon}{M - \varepsilon} \right)^{1/4} > 0.$$

This yields the estimate

$$|\|A_n^{-1}\| - M| = O(e^{-\delta n/r(n)}) + O(\beta_n).$$

Now we assume that $M = M_0$. By Corollary 4.4 and Lemma 4.5, we get that

$$M(n) - \|B_n\| \leq |M(n) - M_0(n)| + 41M_0(n)\frac{r(n)}{n} = O(\beta_n) + O(r(n)/n),$$

and

$$\|B_n\| - M(n) \leq 3M(n)\frac{r(n)}{n} = O(r(n)/n)$$

for all sufficiently large n . Therefore we obtain

$$|\|A_n^{-1}\| - M| = O(r(n)/n) + O(\beta_n).$$

Finally, the convergence of β_n to zero follows from the compactness of the operators K , L , X , and Y and from the strong convergence of the orthogonal projectors Q_n to zero, since $\|C_n\| \rightarrow 0$ and, obviously, $E_n(a^{-1}) \rightarrow 0$. $\square$

To estimate the convergence speed of the singular values we will use matrices of the form $T_n(a)T_n(b) + \lambda P_n$, where $a, b \in C_{N \times N}$ and $\lambda \in \mathbb{C}$. Assuming that the operators $T(a)T(b) + \lambda I$ and $T(\tilde{a})T(\tilde{b}) + \lambda I$ are invertible we put $N_0 := \|T((ab + \lambda E)^{-1})\|$ and $N := \max(\|(T(a)T(b) + \lambda I)^{-1}\|, \|(T(\tilde{a})T(\tilde{b}) + \lambda I)^{-1}\|)$. From Theorem 5.1 we get that $\|(T_n(a)T_n(b) + \lambda P_n)^{-1}\|$ converges to N , and, moreover, we can estimate the speed of this convergence by the smoothness of the functions a and b .

COROLLARY 5.2. *Let $a, b \in C_{N \times N}^\alpha$, $\alpha > 0$, $\lambda \in \mathbb{C}$, $N_0 = \|T((ab + \lambda E)^{-1})\|$, $N = \max(\|(T(a)T(b) + \lambda I)^{-1}\|, \|(T(\tilde{a})T(\tilde{b}) + \lambda I)^{-1}\|)$. Then*

$$(5.2) \quad |\|A_n^{-1}\| - N| = \begin{cases} O\left(\left(\frac{\ln n}{n}\right)^\alpha\right) & \text{if } N > N_0, \\ O\left(\left(\frac{1}{n}\right)^{\frac{\alpha}{1+\alpha}}\right) & \text{if } N = N_0. \end{cases}$$

Let $a, b \in C_{N \times N}^\alpha$, $\alpha > 0$, $\lambda \in \mathbb{C}$, $N_0 = \|T((ab + \lambda E)^{-1})\|$, $N = \max(\|(T(a)T(b) + \lambda I)^{-1}\|, \|(T(\tilde{a})T(\tilde{b}) + \lambda I)^{-1}\|)$.

$$(5.3) \quad |\|A_n^{-1}\| - N| = \begin{cases} O\left(e^{-\gamma\sqrt{n}}\right) & \text{if } N > N_0, \\ O\left(\frac{\ln n}{n}\right) & \text{if } N = N_0, \end{cases}$$

where $\gamma > 0$.

In the proof of Corollary 4.4 we have shown that the matrices A_n and A_n^{-1} can be written in form (2.1), i.e.,

$$\begin{aligned} A_n &= T_n(ab + \lambda E) + P_n K P_n + W_n L W_n, \\ A_n^{-1} &= T_n((ab + \lambda E)^{-1}) + P_n X P_n + W_n Y W_n + D_n. \end{aligned}$$

Moreover, we have obtained there that

$$\|D_n\|, \|KQ_{[n/2]}\|, \|LQ_{[n/2]}\|, \|Q_{[n/2]}X\|, \|Q_{[n/2]}Y\| = O(n^{-\alpha}).$$

Completely analogously, it can be checked that

$$\|XQ_{r(n)}\|, \|YQ_{r(n)}\|, \|Q_{r(n)}X\|, \|Q_{r(n)}Y\| = O(r(n)^{-\alpha}).$$

As $C_n = 0$ and $E_{r(n)}((ab + \lambda E)^{-1}) = O(r(n)^{-\alpha})$, this yields that $\beta_n = O(r(n)^{-\alpha})$. Now we suppose that $N > N_0$. From Theorem 5.1 we obtain the estimate

$$|\|A_n^{-1}\| - N| = O(e^{-\delta n/r(n)}) + O(r(n)^{-\alpha}),$$

with some $\delta > 0$. Putting $r(n) = [(\delta n)/(\alpha \ln n)]$ we arrive at the estimate

$$|\|A_n^{-1}\| - N| = O\left(\left(\frac{1}{n}\right)^\alpha\right) + O\left(\left(\frac{\ln n}{n}\right)^\alpha\right) = O\left(\left(\frac{\ln n}{n}\right)^\alpha\right).$$

Now we suppose that $N = N_0$. Putting $r(n) = [n^{1/(1+\alpha)}]$ we get by Theorem 5.1 that

$$|\|A_n^{-1}\| - N| = O\left(\frac{r(n)}{n}\right) + O(r(n)^{-\alpha}) = O(n^{-\frac{\alpha}{1+\alpha}}).$$

If the functions a and b are rational, then it can be shown in just the same way that $\beta_n = O(e^{-\delta r(n)})$ with some $\delta > 0$. Thus the estimate (5.3) follows immediately from Theorem 5.1 by taking $r(n) = [\ln n/\delta]$ for the case $N = N_0$ and $r(n) = [\sqrt{n}]$ for the case $N > N_0$. $\square$

In particular, this result provides an estimate for the speed with which $\|T_n^{-1}(a)\|$ converges to $N = \max(\|T^{-1}(a)\|, \|T^{-1}(\tilde{a})\|)$ if the operators $T(a)$ and $T(\tilde{a})$ are invertible and if the function $a \in C_{N \times N}$ is smooth.

COROLLARY 5.3. *Let $a \in C_{N \times N}^\alpha$, $\alpha > 0$. Then*

$$|\|T_n^{-1}(a)\| - N| = \begin{cases} O\left(\left(\frac{\ln n}{n}\right)^\alpha\right) & \text{if } N > \|T(a^{-1})\|, \\ O\left(\left(\frac{1}{n}\right)^{\frac{\alpha}{1+\alpha}}\right) & \text{if } N = \|T(a^{-1})\|. \end{cases}$$

where $\tilde{a} = a^{-1}$ if $N > \|T(a^{-1})\|$ and $\tilde{a} = a$ if $N = \|T(a^{-1})\|$.

$$|\|T_n^{-1}(a)\| - N| = \begin{cases} O(e^{-\gamma\sqrt{n}}) & \text{if } N > \|T(a^{-1})\|, \\ O\left(\frac{\ln n}{n}\right) & \text{if } N = \|T(a^{-1})\|, \end{cases}$$

where $\gamma > 0$.

6. The relationship between M and M_0 . Given a sequence $(A_n) \in \mathbf{S}(C)$, it is in general difficult to decide whether $M = \max(\|[\mathcal{W}_1(A_n)]^{-1}\|, \|[\mathcal{W}_2(A_n)]^{-1}\|)$ is greater than $M_0 = \|T(a^{-1})\|$ or equal to $\|T(a^{-1})\|$. In this section we present an answer of this question.

First, we recall some basic facts concerning Moore–Penrose invertibility of linear operators. Let H be a Hilbert space. An operator $A \in \mathcal{L}(H)$ is called Moore–Penrose invertible if there is an operator $A^+ \in \mathcal{L}(H)$ such that

$$AA^+A = A, \quad A^+AA^+ = A^+, \quad (AA^+)^* = AA^+, \quad (A^+A)^* = A^+A.$$

If such an operator A^+ exists, then it is uniquely determined. Clearly, if A is invertible, then it is Moore–Penrose invertible, and the operator A^+ coincides with the inverse of A . It is well known that an operator is Moore–Penrose invertible if and only if its range is closed. In particular, if A is Fredholm, then it is Moore–Penrose invertible.

As the following theorem shows, the Moore–Penrose invertibility of an operator $A \in \mathcal{L}(H)$ is closely related to the set $\Sigma(A)$ of the singular values of A .

THEOREM 6.1 (see [2, Theorems 4.21 and 4.23]). Let $A \in \mathcal{L}(H)$, $d > 0$, $\Sigma(A) \subset \{0\} \cup [d, \infty)$. Then

$$\|A^+\|_{\mathcal{L}(H)} = \max(1/s : s \in \Sigma(A) \setminus \{0\}).$$

Further, we recall an auxiliary result which can be found in every textbook on linear algebra.

LEMMA 6.2 (singular value decomposition). Let $C \in \mathbb{C}^{k \times k}$, $U, V \in \mathbb{C}^{k \times k}$ unitary, $\Sigma = \text{diag}(s_1, \dots, s_k)$, $0 \leq s_1 \leq \dots \leq s_k$, $C = U\Sigma V^*$.

Note that, for all sequences $(A_n) \in \mathbf{S}(C)$, the operators $\mathcal{W}_1(A_n)$ and $\mathcal{W}_2(A_n)$ are the sum of a Toeplitz operator and a compact operator.

LEMMA 6.3. Let $a \in C_{N \times N}$, $K \in \mathcal{L}(\ell_N^2)$, $A = T(a) + K$, $\|A^+\| > \|T(a^{-1})\|$, $d = 1/\|A^+\|$. Taking into account that $T^*(a) = T(a^*)$, where $a^*(t) = (a_{ji}(t))_{i,j=1}^N$, we get, for any $\lambda \in \mathbb{C}$,

$$\begin{aligned} A^*A - \lambda I &= (T(a) + K)^*(T(a) + K) - \lambda I \\ &= T(a^*)T(a) - \lambda I + K^*T(a) + T(a^*)K + K^*K \\ &= T(a^*a) - \lambda I + K^*T(a) + T(a^*)K + K^*K - H(a^*)H(\tilde{a}) \\ &= T(a^*a - \lambda E) + K_1, \end{aligned}$$

where the operator K_1 is compact. This implies that the operator $A^*A - \lambda I$ is Fredholm if and only if the Toeplitz operator $T(a^*a - \lambda E)$ is Fredholm, i.e., if and only if $\det(a^*(t)a(t) - \lambda E) \neq 0$ for all $t \in \mathbb{T}$.

Further, for each fixed $t \in \mathbb{T}$, let $U(t)\Sigma(t)V(t)$ be the singular value decomposition of the $N \times N$ matrix $a(t)$, where $\Sigma(t) = \text{diag}(s_1(t), \dots, s_N(t))$. We obtain that

$$\begin{aligned} (6.1) \quad \det(a^*(t)a(t) - \lambda E) &= \det(V^*(t)\Sigma^2(t)V(t) - \lambda E) \\ &= \det(V^*(t)(\Sigma^2(t) - \lambda E)V(t)) = \det(\Sigma^2(t) - \lambda E) = \prod_{i=1}^N (s_i^2(t) - \lambda). \end{aligned}$$

As the operator A is Fredholm, we have by Theorem 6.1 that $\Sigma(A) \subset \{0\} \cup [d, \infty)$ with $d = 1/\|A^+\|$. Hence the operator $A^*A - \lambda I$ is Fredholm or, equivalently, $\det(a^*(t)a(t) - \lambda E) \neq 0$ for all $\lambda \in [0, d^2)$. This and (6.1) yield that $s_1(t) \geq d$ for all $t \in \mathbb{T}$.

In summary, we obtain that

$$\begin{aligned} \|T(a^{-1})\| &= \|a^{-1}\|_\infty = \max_{t \in \mathbb{T}} 1/s_1(t) < \|A^+\| = 1/d \\ &\Leftrightarrow d_1 = \min_{t \in \mathbb{T}} s_1(t) > d \Leftrightarrow \det(a^*(t)a(t) - d^2 E) \neq 0 \quad \forall t \in \mathbb{T} \\ &\Leftrightarrow \text{the operator } A^*A - d^2 I \text{ is Fredholm.} \quad \square \end{aligned}$$

In what follows let $\text{sp}A$ stand for the spectrum of an operator A .

LEMMA 6.4. Let $a \in C_{N \times N}$, $K \in \mathcal{L}(\ell_N^2)$, $A = T(a) + K$, $\|A^+\| > \|T(a^{-1})\|$, $d = 1/\|A^+\|$, $\Sigma(A) \subset \{0\} \cup \{d\} \cup [d_*, \infty)$, $d_* > d$.

From the previous lemma it follows that if $\|A^+\| > \|T(a^{-1})\|$, then the operator $A^*A - d^2I$ is Fredholm. Using Theorem 6.1 we obtain that

$$\Sigma(A^*A - d^2I) = (\text{sp}(A^*A - d^2I)^*(A^*A - d^2I))^{1/2} \subset \{0\} \cup [d_1, \infty),$$

with some $d_1 > 0$. Since the operator $(A^*A - d^2I)$ is self-adjoint, we get that

$$\text{sp}(A^*A - d^2I) \subset \{0\} \cup (-\infty, -d_1] \cup [d_1, \infty).$$

Finally, taking into account that $\Sigma(A) \subset \{0\} \cup [d, \infty)$ we obtain that

$$\Sigma(A) = (\text{sp}A^*A)^{1/2} \subset \{0\} \cup \{d\} \cup [\sqrt{d^2 + d_1}, \infty). \quad \square$$

As the following simple example shows, the reverse implication is not fulfilled; i.e., there are Fredholm operators $A = T(a) + K$ such that $\Sigma(A) \subset \{0\} \cup \{d\} \cup [d_*, \infty)$, $d_* > d > 0$, and $\|T(a^{-1})\| = \|A^+\| = 1/d$.

Let $K = 0$, and let a be the 2×2 diagonal matrix $\text{diag}(1, 2)$. Thus

$$\begin{aligned} A &= T(a) = \text{diag}(1, 2, 1, 2, 1, 2, \dots), \\ A^{-1} &= T(a^{-1}) = \text{diag}(1, 1/2, 1, 1/2, 1, 1/2, \dots), \\ \Sigma(A) &= \{1\} \cup \{2\}, \quad \text{and} \quad \|A^{-1}\| = \|T(a^{-1})\| = 1. \end{aligned}$$

From Theorem 2.3 (or from the proof of Lemma 6.3) it follows that, for any sequence $(A_n) \in \mathbf{S}(C)$ and each $\lambda \in \mathbb{C}$, the operators $(\mathcal{W}_1^*(A_n)\mathcal{W}_1(A_n) - \lambda I)$ and $(\mathcal{W}_2^*(A_n)\mathcal{W}_2(A_n) - \lambda I)$ are Fredholm simultaneously. Thus Lemmas 6.3 and 6.4 imply the next result.

THEOREM 6.5. Let $(A_n) \in \mathbf{S}(C)$, $\mathcal{W}_1(A_n) \rightarrow \mathcal{W}_1$, $\mathcal{W}_2(A_n) \rightarrow \mathcal{W}_2$, $M = \max(\|\mathcal{W}_1(A_n)\|^{-1}, \|\mathcal{W}_2(A_n)\|^{-1})$, $M > \|T(a^{-1})\|$, $d = 1/M$, $M > \|T(a^{-1})\|$, $d > 0$, $\Sigma(\mathcal{W}_1(A_n)) \cup \Sigma(\mathcal{W}_2(A_n)) \subset \{0\} \cup \{d\} \cup [d_*, \infty)$.

7. Convergence of the singular values of $T_n(a)$. Let (E_n) be a sequence of nonempty subsets of $\mathbb{C}$. We define the uniform limiting set $\liminf(E_n)$ as the set of all $\lambda \in \mathbb{C}$ for which there are $\lambda_n \in E_n$ such that $\lambda_n \rightarrow \lambda$, and the partial limiting set $\limsup(E_n)$ is defined as the set of all $\lambda \in \mathbb{C}$ for which there are $\lambda_{n_k} \in E_{n_k}$ such that $n_k \rightarrow \infty$ and $\lambda_{n_k} \rightarrow \lambda$. Clearly, we always have $\liminf(E_n) \subset \limsup(E_n)$.

LEMMA 7.1 (see [1, Theorem 2.19]). Let $(A_n) \in \mathbf{S}(C)$,

$$\liminf(\Sigma(A_n)) = \limsup(\Sigma(A_n)) = \Sigma(\mathcal{W}_1(A_n)) \cup \Sigma(\mathcal{W}_2(A_n)).$$

Using this lemma and Theorem 2.3 we can show that if for a sequence $(A_n) \in \mathbf{S}(C)$ the operator $\mathcal{W}_1(A_n)$ is Fredholm, then the sequence $(s_{k+1}(A_n))$ of the $(k+1)$ th singular values of matrices A_n converges to $\min(\|\mathcal{W}_1^+(A_n)\|^{-1}, \|\mathcal{W}_2^+(A_n)\|^{-1})$.

THEOREM 7.2. Let $(A_n) \in \mathbf{S}(C)$, $\mathcal{W}_1(A_n) \rightarrow \mathcal{W}_1$, $\mathcal{W}_2(A_n) \rightarrow \mathcal{W}_2$, $k = \dim \ker \mathcal{W}_1(A_n) + \dim \ker \mathcal{W}_2(A_n)$.

$$\lim_{n \rightarrow \infty} s_{k+1}(A_n) = \min(\|\mathcal{W}_1^+(A_n)\|^{-1}, \|\mathcal{W}_2^+(A_n)\|^{-1}).$$

Put $d := \min(\|\mathcal{W}_1^+(A_n)\|^{-1}, \|\mathcal{W}_2^+(A_n)\|^{-1})$. Then by Theorem 6.1 we get

$$d \in \Sigma(\mathcal{W}_1(A_n)) \cup \Sigma(\mathcal{W}_2(A_n)) \subset \{0\} \cup [d, \infty).$$

From Lemma 7.1 it follows that

$$d_0 := \liminf_{n \rightarrow \infty} s_{k+1}(A_n) \in \limsup(\Sigma(A_n)) \subset \{0\} \cup [d, \infty).$$

As $d_0 > 0$ by Theorem 2.3, this implies that $d_0 \geq d$. Again from Lemma 7.1 we obtain that $d \in \liminf(\Sigma(A_n))$. Hence there is a sequence (λ_n) such that $\lambda_n \in \Sigma(A_n)$ and $\lambda_n \rightarrow d$. Taking into account Theorem 2.3 we obtain that $s_k(A_n) < d/2$ and $|\lambda_n - d| < d/2$ for all sufficiently large n . This yields that $\lambda_n \geq s_{k+1}(A_n)$ for all sufficiently large n . Thus

$$\limsup_{n \rightarrow \infty} s_{k+1}(A_n) \leq \lim_{n \rightarrow \infty} \lambda_n = d \leq d_0 = \liminf_{n \rightarrow \infty} s_{k+1}(A_n).$$

This completes the proof. $\square$

Now we estimate the speed with which the $(k+1)$ th singular values of the finite sections $T_n(a)$ converge to $d := \min(\|T(a)^+\|^{-1}, \|T(\tilde{a})^+\|^{-1})$.

THEOREM 7.3. *Let $a \in C_{N \times N}^\alpha$, $\alpha > 0$, $d < \|T(a^{-1})\|^{-1}$, $k = \dim \ker T(a) + \dim \ker T(\tilde{a})$.*

$$(7.1) \quad |d - s_{k+1}(T_n(a))| = \begin{cases} O\left(\left(\frac{\ln n}{n}\right)^\alpha\right) & \text{if } d < \|T(a^{-1})\|^{-1}, \\ O\left(\left(\frac{1}{n}\right)^{\frac{\alpha}{1+\alpha}}\right) & \text{if } d = \|T(a^{-1})\|^{-1}. \end{cases}$$

Let $a \in C_{N \times N}^\gamma$, $\gamma > 0$, $d < \|T(a^{-1})\|^{-1}$.

$$(7.2) \quad |d - s_{k+1}(T_n(a))| = \begin{cases} O\left(e^{-\gamma\sqrt{n}}\right) & \text{if } d < \|T(a^{-1})\|^{-1}, \\ O\left(\frac{\ln n}{n}\right) & \text{if } d = \|T(a^{-1})\|^{-1}. \end{cases}$$

From Theorems 2.3 and 7.2 we get that there is a number n_0 such that for all $n > n_0$

$$s_k^2(T_n(a)) < \frac{d^2}{4} \quad \text{and} \quad |s_{k+1}^2(T_n(a)) - d^2| < \frac{d^2}{5}.$$

Put $\lambda = 3d^2/4$, and recall that $T^*(a) = T(a^*)$. Since $\Sigma(T(a)) \cup \Sigma(T(\tilde{a})) \subset \{0\} \cup [d, \infty)$ we obtain that

$$\text{sp}(T(a^*)T(a) - \lambda I) \cup \text{sp}(T(\tilde{a}^*)T(\tilde{a}) - \lambda I) \subset \left\{-\frac{3}{4}d^2\right\} \cup \left[\frac{1}{4}d^2, \infty\right).$$

Thus we get that the operators $(T(a^*)T(a) - \lambda I)$ and $(T(\tilde{a}^*)T(\tilde{a}) - \lambda I)$ are invertible. Moreover, since these operators are self-adjoint, it follows that

$$N := \max(\|(T(a^*)T(a) - \lambda I)^{-1}\|, \|(T(\tilde{a}^*)T(\tilde{a}) - \lambda I)^{-1}\|) = 4/d^2.$$

Further, we have, for any $n > n_0$,

$$\begin{aligned} \text{sp}(T_n(a^*)T_n(a) - \lambda P_n) &\subset \left[-\frac{3}{4}d^2, s_k^2(T_n(a)) - \frac{3}{4}d^2\right] \cup \left[s_{k+1}^2(T_n(a)) - \frac{3}{4}d^2, \infty\right) \\ &\subset \left[-\frac{3}{4}d^2, -\frac{1}{2}d^2\right) \cup \left(\frac{1}{20}d^2, \infty\right). \end{aligned}$$

This implies the invertibility of the operator $(T_n(a^*)T_n(a) - \lambda P_n)$. Taking into account that the operators $(T_n(a^*)T_n(a) - \lambda P_n)$ are self-adjoint and that, for all $n > n_0$,

$$|s_{k+1}^2(T_n(a)) - 3d^2/4| < 9d^2/20 < |s_k^2(T_n(a)) - 3d^2/4|,$$

we obtain that, for all $n > n_0$,

$$\|(T_n(a^*)T_n(a) - \lambda P_n)^{-1}\| = (s_{k+1}^2(T_n(a)) - 3d^2/4)^{-1}.$$

Hence we arrive at the inequality

$$\begin{aligned} \left| \|(T_n(a^*)T_n(a) - \lambda P_n)^{-1}\| - N \right| &= \left| (s_{k+1}^2(T_n(a)) - 3d^2/4)^{-1} - 4/d^2 \right| \\ &= d^{-2} (s_{k+1}^2(T_n(a)) - 3d^2/4)^{-1} |d^2 - 4s_{k+1}^2(T_n(a)) - 3d^2| \\ &\geq 10d^{-4} |d^2 - s_{k+1}^2(T_n(a))| \geq 10d^{-3} |d - s_{k+1}(T_n(a))|. \end{aligned}$$

This yields that

$$|d - s_{k+1}(T_n(a))| = O(|\|(T_n(a^*)T_n(a) - \lambda P_n)^{-1}\| - N|).$$

Further, for each fixed $t \in \mathbb{T}$, let $U(t)\Sigma(t)V(t)$ be the singular value decomposition of the $N \times N$ matrix $a(t)$, where $\Sigma(t) = \text{diag}(s_1(t), \dots, s_N(t))$. Thus we obtain that

$$\|T(a^{-1})\|^{-1} = \|a^{-1}\|_{\infty}^{-1} = \left(\max_{t \in \mathbb{T}} 1/s_1(t) \right)^{-1} = \min_{t \in \mathbb{T}} s_1(t),$$

and

$$\|T((a^*a - \lambda E)^{-1})\| = \|(a^*a - \lambda E)^{-1}\|_{\infty} = \max_{t \in \mathbb{T}} (s_1^2(t) - \lambda)^{-1} = \left(\min_{t \in \mathbb{T}} s_1^2(t) - \lambda \right)^{-1}.$$

Consequently, the condition $d < \|T(a^{-1})\|^{-1}$ implies the strict inequality

$$\|T((a^*a - \lambda E)^{-1})\| < (d^2 - \lambda)^{-1} = 4/d^2 = N.$$

Now the estimates (7.1) and (7.2) follow immediately from Corollary 5.2. $\square$

Note that in view of Lemma 6.3 the condition $d < \|T(a^{-1})\|^{-1}$ is equivalent to the Fredholmness of the operator $T(a^*)T(a) - d^2I$.

Moreover, we remark that although the condition $d < \|T(a^{-1})\|^{-1}$ is difficult to verify, one can always use the worse estimates. In particular, if a is rational, then $s_{k+1}(T_n(a))$ converges to d at least with the speed $O(\ln n/n)$.

REFERENCES

- [1] A. BÖTTCHER AND S. GRUDSKY, *Toeplitz Matrices, Asymptotical Linear Algebra and Functional Analysis*, Hindustan Book Agency, New Delhi, 2000 and Birkhäuser Verlag, Basel, 2000.
- [2] A. BÖTTCHER AND B. SILBERMANN, *Introduction to Large Truncated Toeplitz Matrices*, Springer-Verlag, New York, 1999.
- [3] A. BÖTTCHER AND B. SILBERMANN, *Analysis of Toeplitz Operators*, Akademie-Verlag, Berlin, 1989 and Springer-Verlag, Berlin, Heidelberg, New York, 1990.
- [4] S. ROCH AND B. SILBERMANN, *Index calculus for approximation methods and singular value decomposition*, J. Math. Anal. Appl., 225 (1998), pp. 401–426.
- [5] B. SILBERMANN, *Modified finite sections for Toeplitz operators and their singular values*, SIAM J. Matrix Anal. Appl., 24 (2003), pp. 678–692.

- [6] B. SILBERMANN, *How to compute the partial indices of a regular and smooth matrix-valued function?*, in Factorization, Singular Operators and Related Problems, S. Samko, A. Lebre, and A. F. dos Santos, eds., Kluwer Academic Publishers, Dordrecht, Boston, London, 2003, pp. 291–300.
- [7] E. E. TYRTYSHNIKOV, *New theorems on the distribution of eigenvalues and singular values of multilevel Toeplitz matrices*, Dokl. Akad. Nauk, 333 (1993), pp. 300–303 (in Russian).
- [8] E. E. TYRTYSHNIKOV, *A unifying approach to some old and new theorems on distribution and clustering*, Linear Algebra Appl., 232 (1996), pp. 1–43.
- [9] H. WIDOM, *On the singular values of Toeplitz matrices*, Z. Anal. Anwendungen, 8 (1989), pp. 221–229.
- [10] A. ZYGMUND, *Trigonometric series*, Vol. I, Cambridge University Press, New York, 1959.

ON THE HERMITIAN-GENERALIZED HAMILTONIAN SOLUTIONS OF LINEAR MATRIX EQUATIONS*

ZHONG-ZHI ZHANG[†], XI-YAN HU[‡], AND LEI ZHANG[‡]

Abstract. By means of the properties of Hermitian-generalized Hamiltonian matrices, we give some necessary and sufficient conditions for the solvability of the matrix equation $AX = B$ in Hermitian-generalized Hamiltonian matrix set $HHC^{m \times n}$. In the case where $AX = B$ is solvable in $HHC^{n \times n}$, we derive the general representation of the solutions. We also give the expression of the solution for the corresponding optimal approximation problem.

Key words. generalized Hamiltonian matrix, Hermitian-generalized Hamiltonian matrix, matrix equation, matrix norm, optimal approximation

AMS subject classifications. 15A24, 15A30

DOI. 10.1137/S0895479801396725

1. Introduction. Let us first introduce some notations and concepts that will be used in the paper. Let $R^{n \times m}$ and $C^{n \times m}$ be the sets of all real and complex $n \times m$ matrices, respectively. It is clear that $C^{n \times m}$ is a complete inner product space. We abbreviate $C^{n \times 1}$ as C^n . Let $HC^{n \times n}$ stand for the set of all $n \times n$ Hermitian matrices and $AHC^{n \times n}$ the set of all $n \times n$ anti-Hermitian matrices. We denote the conjugate transformation and the Moore–Penrose generalized inverse of a matrix A by A^H and A^+ , respectively, and the identity matrix of order n by I_n . We use $\langle A, B \rangle = \text{tr}(B^H A)$ to define the inner product of matrices A and B in $C^{n \times m}$. Without specification, the matrix norm used in the paper is the so-called Frobenius norm defined by $\|A\| = \sqrt{\langle A, A \rangle} = (\text{tr}(A^H A))^{1/2}$. Let $OASR^{n \times n}$ be the set of all $n \times n$ real orthogonal antisymmetric matrices, i.e.,

$$OASR^{n \times n} = \{J | J^T J = J J^T = I_n, J = -J^T, J \in R^{n \times n}\}.$$

It is clear that $J^2 = -I_n \forall J \in OASR^{n \times n}$. Consequently, n must be an even integer.

DEFINITION 1.1. Let $J \in OASR^{n \times n}$.

- (i) $A \in C^{n \times n}$, $A^H J = J A$, $J A J = A^H$
 $A \in HTC^{n \times n}$
- (ii) $A \in C^{n \times n}$, $A^H J = -J A$, $J A J = -A^H$
 $A \in SHTC^{n \times n}$

It is not difficult to see that $HTC^{n \times n}$ and $SHTC^{n \times n}$ depend upon matrix J . If $J = \begin{pmatrix} 0 & I_k \\ -I_k & 0 \end{pmatrix}$, then $HTC^{n \times n}$ and $SHTC^{n \times n}$ reduce to the sets of well-known Hamiltonian and skew Hamiltonian matrices, respectively.

*Received by the editors October 17, 2001; accepted for publication (in revised form) by V. Mehrmann May 17, 2004; published electronically August 31, 2005. This research was supported by NSF of China grant 10171031 and by the postdoctoral project of Central South University.

<http://www.siam.org/journals/simax/27-1/39672.html>

[†]College of Mathematics and Computational Science, Central South University, Changsha, Hunan 410083, People's Republic of China and Department of Mathematics, Dongguan University of Technology, Dongguan, Guangdong 523000, People's Republic of China (zhangzz@dgut.edu.cn). This work was done while the author was a Postdoctoral Fellow of the Department of Mathematics, Central South University, Changsha, Hunan 410083, People's Republic of China.

[‡]College of Mathematics and Econometrics, Hunan University, Changsha, Hunan 410082, People's Republic of China (xyhu@mail.hunu.edu.cn).

Throughout the paper, we always assume that the matrix J is fixed.

DEFINITION 1.2. $J \in OASR^{n \times n}$.

- (i) $A \in C^{n \times n}$, $A^H = A$, $JAJ = A^H$
- (ii) $A \in C^{n \times n}$, $A^H = A$, $JAJ = -A^H$

We denote $HHC^{n \times n}$ and $HSNC^{n \times n}$ the sets of all $n \times n$ Hermitian-generalized Hamiltonian matrices and Hermitian-generalized skew Hamiltonian matrices, respectively, i.e.,

$$\begin{aligned} HHC^{n \times n} &= \{A | A = A^H, JAJ = A^H, A \in C^{n \times n}\}, \\ HSHC^{n \times n} &= \{A | A = A^H, JAJ = -A^H, A \in C^{n \times n}\}. \end{aligned}$$

Let $i = \sqrt{-1}$.

DEFINITION 1.3. $J \in OASR^{n \times n}$.

- (i) $x \in C^n$, $iJx = x$
- (ii) $x \in C^n$, $iJx = -x$

In this paper, we are concerned with the following two problems.

I. Given $X, B \in C^{m \times m}$, find a matrix $A \in S$ such that

$$(1.1) \quad AX = B,$$

where $S = HHC^{n \times n}$.

II. Given $A^* \in C^{n \times n}$, find a matrix $\hat{A} \in S_{X,B}$ such that

$$(1.2) \quad \|A^* - \hat{A}\| = \min_{A \in S_{X,B}} \|A^* - A\|,$$

where $S_{X,B}$ is the solution set of Problem I.

Problems I and II have attracted many people's attention [7, 8, 9, 11, 12, 13, 14, 17, 18, 22]. Khatri and Mitra [8] studied the solution of Problem I where S consists of the Hermitian and nonnegative definite matrices. They established necessary and sufficient conditions for the existence of the solutions and obtained the general expression for solutions. Wu and Cain [9] investigated the solutions of Problem I where S is the set of all renonnegative definite matrices. They also proposed necessary and sufficient conditions for the solvability of the problem and derived an expression for the general solutions. Xie, Zhang, and Hu [12] gave the necessary and sufficient conditions for the solvability of Problem I with bisymmetric constraint and derived an expression for the general solutions. However, to our knowledge, so far, there seems to be no study on Problems I and II with $S = HHC^{n \times n}$. On the other hand, the Hamiltonian matrices are very important in engineering. For example, using Hamiltonian matrices to solve algebraic matrix Riccati equation is very effective in optimal control theory [19, 20]. In the following, we describe a practical problem which arises in a vibration system with n degree of freedom in detail.

Let $\omega_1, \dots, \omega_m$ ($m \leq n$) be the natural frequencies and $\phi_1, \dots, \phi_m$ be the corresponding vibration styles. Let

$$(1.3) \quad \phi = [\phi_1, \dots, \phi_m], \quad \Omega^2 = \text{diag}(\omega_1^2, \dots, \omega_m^2).$$

The mass matrix M is a known $n \times n$ real symmetric positive definite matrix, ϕ has been corrected [1], and the following orthogonal condition holds:

$$(1.4) \quad \phi^T M \phi = I_m.$$

Let K represent the stiffness matrix to be corrected. The matrix K_0 , a preliminary estimate of the stiffness matrix K , can be obtained by finite element analysis, experimental observation, or information of statistical distribution. However, the matrix K_0 is not necessarily a symmetric positive semidefinite matrix. Baruch and Bar-Itzhack [1], and Baruch [2, 3] presented a minimal weighted Euclidean norm method to correct K_0 . The problem can be stated as follows: Find K such that

$$(1.5) \quad \min \frac{1}{2} \|M^{-\frac{1}{2}}(K - K_0)M^{-\frac{1}{2}}\|,$$

where the stiffness matrix K satisfies the following constrained conditions:

(1) characteristic equation:

$$(1.6) \quad K\phi = M\phi\Omega^2;$$

(2) symmetric property:

$$(1.7) \quad K^T = K;$$

(3) positive semidefinite property:

$$(1.8) \quad K \geq 0,$$

where $K \geq 0$ means that K is a positive semidefinite matrix.

Let $A = M^{-\frac{1}{2}}KM^{-\frac{1}{2}}$, $A_0 = M^{-\frac{1}{2}}K_0M^{-\frac{1}{2}}$, $X = M^{\frac{1}{2}}\phi$, $\Lambda = \text{diag}(\lambda_1, \dots, \lambda_m) = \Omega^2$, and $S = \{A | AX = X\Lambda, A^T = A \text{ and } A \geq 0\}$.

By using the properties of the mass matrix M , it is not difficult to see that problem (1.5) is equivalent to the following problem: Given $X \in R^{n \times m}$, $\Lambda \in R^{m \times m}$, and $A_0 \in R^{n \times n}$, find $A \in S$ such that

$$(1.9) \quad \min \|A - A_0\|.$$

There have developed some numerical methods for solving problem (1.9). For example, the optimization based methods, such as the Lagrangian multiplier method by Baruch and Bar-Itzhack [1], Baruch [2, 3], Wei [4], Berman and Wei [5], and Berman and Nagy [6]. In some cases, the corrected stiffness matrix obtained by these methods may not be positive semidefinite. However, from a practical point of view, the stiffness matrix should be positive definite. There are also some matrix theory-based methods for solving (1.9) [7, 8, 9, 10, 11, 12, 13, 14, 15, 16]. In this paper, we will derive a general expression to the solution of Problem I.

This paper is organized as follows. In section 2, we discuss the properties of $HH^C^{n \times n}$. In section 3, we establish the necessary and sufficient conditions for the solvability of Problem I. We will also present a general representation of the solutions of Problem I if it has a solution. In section 4, we establish an existence and uniqueness theorem for Problem II. We will also give an expression of the optimal approximation solution in section 4.

2. Properties of the Matrix set $HHC^{n \times n}$. In this section, we discuss the properties of the matrix set $HHC^{n \times n}$.

THEOREM 2.1. *Let $J \in OASR^{n \times n}$. Then*

$$(2.1) \quad C^{n \times n} = HC^{n \times n} \oplus AHC^{n \times n},$$

$$(2.2) \quad HC^{n \times n} = HHC^{n \times n} \oplus HSHC^{n \times n},$$

where $C^{n \times n} = HC^{n \times n} \oplus AHC^{n \times n}$.

We prove only equality (2.2). Equality (2.1) can be proved in a similar way. In the following, we prove (2.2) by three steps.

(1) For $A \in HC^{n \times n}$, let $A_1 = \frac{1}{2}(A + JAJ)$ and $A_2 = \frac{1}{2}(A - JAJ)$. Then we immediately have $A_1 \in HHC^{n \times n}$, $A_2 \in HSHC^{n \times n}$, and $A = A_1 + A_2$.

(2) For $A \in HHC^{n \times n} \cap HSHC^{n \times n}$, we have $JAJ = A^H = A$ and $JAJ = -A^H = -A$. This implies $A = -A$, and hence $A = 0$.

(3) For $A_1 \in HHC^{n \times n}$ and $A_2 \in HSHC^{n \times n}$, we have by the definition of $HHC^{n \times n}$ and $HSHC^{n \times n}$ that $JA_1J = A_1^H = A_1$ and $JA_2J = -A_2^H = -A_2$. So, we get

$$\begin{aligned} \langle A_1, A_2 \rangle &= \langle JA_1J, -JA_2J \rangle \\ &= -\text{tr}[(JA_2J)^H JA_1J] \\ &= -\text{tr}(J^T A_2^H J^T JA_1J) \\ &= -\text{tr}(J^T A_2^H A_1J) \\ &= -\text{tr}(A_2^H A_1JJ^T) \\ &= -\text{tr}(A_2^H A_1) \\ &= -\langle A_1, A_2 \rangle. \end{aligned}$$

This yields $\langle A_1, A_2 \rangle = 0$.

The above argument has shown that (2.2) holds. In a similar way, we can prove (2.1). The proof is then complete. $\square$

It is not difficult to see from Theorem 2.1 that

$$C^{n \times n} = HHC^{n \times n} \oplus HSHC^{n \times n} \oplus AHC^{n \times n}.$$

Consequently, we have the following corollary.

COROLLARY 2.1. *Let $A \in C^{n \times n}$. Then $A_1 \in HHC^{n \times n}$, $A_2 \in HSHC^{n \times n}$, $A_3 \in AHC^{n \times n}$ and*

$$A = A_1 + A_2 + A_3, \quad \langle A_i, A_j \rangle = 0, \quad i \neq j, \quad i, j = 1, 2, 3.$$

where A_1, A_2 , and A_3 are defined by

$$A_1 = \frac{1}{4}[A + A^H + J(A + A^H)J], \quad A_2 = \frac{1}{4}[A + A^H - J(A + A^H)J], \quad A_3 = \frac{1}{2}(A - A^H).$$

For given $J \in OASR^{n \times n}$, let

$$P_1 = \frac{1}{2}(I + iJ), \quad P_2 = \frac{1}{2}(I - iJ).$$

By the definition of $OASR^{n \times n}$, it is not difficult to show that P_1 and P_2 are orthogonal projection matrices. Moreover, we have $P_1 + P_2 = I$ and $P_1P_2 = 0$.

The following lemma can be proved in a way similar to the proof of Theorem 2.1.

LEMMA 2.1. Let $x \in C^n$ and $x_1 = P_1x \in SC^n$, $x_2 = P_2x \in AC^n$, where $P_1 + P_2 = I_n$. Then $x = x_1 + x_2$, $\langle x_1, x_2 \rangle = 0$, and $C^n = SC^n \oplus AC^n$.

LEMMA 2.2. Let $x, b \in C^n$, $A \in HHC^{n \times n}$, and $Ax = b$. Then

- (i) $x \in SC^n$ and $b \in AC^n$
- (ii) $x \in AC^n$ and $b \in SC^n$

It follows from $Ax = b$, $A \in HHC^{n \times n}$, and $J^2 = -I_n$ that $JAJ(Jx) = -Jb$. This implies

$$(2.3) \quad A(iJx) = -iJb.$$

If $x \in SC^n$, we have $iJx = x$. This together with (2.3) and $Ax = b$ implies $b = -iJb$, i.e., $b \in AC^n$. If $x \in AC^n$, we have $iJx = -x$. This together with (2.3) implies $b = iJb$, i.e., $b \in SC^n$. $\square$

LEMMA 2.3. Let $E \in HSHC^{n \times n}$, $F \in C^{n \times n}$, and $M = EFE$. Then

- (i) $F \in HHC^{n \times n}$ and $M \in HHC^{n \times n}$
- (ii) $F \in HSHC^{n \times n}$ and $M \in HSHC^{n \times n}$

It is easy to get from $E \in HSHC^{n \times n}$ that $E^H = E$ and $JEJ = -E$. This implies

$$(2.4) \quad M^H = (EFE)^H = E^H F^H E^H = EF^H E$$

and

$$(2.5) \quad JMJ = JEF EJ = (JEJ)(JFJ)(JEJ) = -E^H JFJ(-E^H) = EJFJE.$$

If $F \in HHC^{n \times n}$, then $F^H = F$, $JFJ = F^H$. By (2.4), we have $M^H = M$. It then follows from (2.5) that $JMJ = M^H$. Consequently, we have $M \in HHC^{n \times n}$.

If $F \in HSHC^{n \times n}$, then we have $F^H = F$ and $JFJ = -F^H$. So, we get from (2.4) that $M^H = M$. This together with (2.5) implies $JMJ = -M^H$. Consequently, we have $M \in HSHC^{n \times n}$. $\square$

THEOREM 2.2. Let $E \in HSHC^{n \times n}$, $F \in HC^{n \times n}$, $F = F_1 + F_2$, $F_1 \in HHC^{n \times n}$, $F_2 \in HSHC^{n \times n}$, $M = EFE$, and $M \in HHC^{n \times n}$. Then $EF_2E = 0$ and $M = EF_1E$.

It is obvious that

$$(2.6) \quad M = EFE = EF_1E + EF_2E.$$

If $EF_2E = 0$, we get from (2.6) that $EF_1E = M$. By Lemma 2.3, it is clear that $M \in HHC^{n \times n}$. If $M \in HHC^{n \times n}$, we get from (2.6) again that $M - EF_1E = EF_2E$. It then follows from Lemma 2.3 that $M - EF_1E \in HHC^{n \times n}$ and $EF_2E \in HSHC^{n \times n}$. Therefore, we get from Theorem 2.1 that $M - EF_1E = EF_2E = 0$. This means $M = EF_1E$ and $EF_2E = 0$. $\square$

3. Solvability of Problem I. In this section, we establish the necessary and sufficient conditions for existence of the solution of Problem I and then derive an expression to the general solutions of Problem I. We first introduce the following two lemmas (see [1, 21]).

LEMMA 3.1. Let $X, B \in C^{n \times m}$, $A \in HHC^{n \times n}$ and $AX = B$.

$$(3.1) \quad X^H B = B^H X$$

$$(3.2) \quad BX^+X = B.$$

$$(3.3) \quad A = A_0 + A_0^H Q + QEQ \quad \forall E \in HHC^{n \times n},$$

$$A_0 = BX^+ \quad Q = I - XX^+$$

LEMMA 3.2. Let $Z \in C^{n \times m}$, $Y \in C^{m \times l}$, $Z^H Y = 0$.

$$(Z, Y)^+ = \begin{pmatrix} Z^+ \\ Y^+ \end{pmatrix}.$$

THEOREM 3.1. Let $X, B \in C^{n \times m}$. I. Let

$$(3.4) \quad X_1^H B_1 = B_2^H X_2$$

$$(3.5) \quad B_1 X_2^+ X_2 = B_1, \quad B_2 X_1^+ X_1 = B_2,$$

$$(3.6) \quad A = A_0 + A_0^H Q + QEQ, \quad E \in HHC^{n \times n},$$

$$A_0 = B_2 X_1^+ + B_1 X_2^+ \quad Q = I - X_1 X_1^+ - X_2 X_2^+$$

It follows from Lemma 2.1 that $X = P_1 X + P_2 X = X_1 + X_2$, where the columns of X_1 are symmetric vectors and the columns of X_2 are antisymmetric vectors. We also have $B = P_1 B + P_2 B = B_1 + B_2$ in which the columns of B_1 are symmetric and the columns of B_2 are antisymmetric. So, we claim that $A \in HHC^{n \times n}$ satisfies $AX = B$ if and only if $AX_1 + AX_2 = B_1 + B_2$. By Lemmas 2.1 and 2.2, we have $AX_1 = B_2$ and $AX_2 = B_1$. Thus, the matrix equation $AX = B$ is equivalent to the following equation:

$$(3.7) \quad A(X_1, X_2) = (B_2, B_1).$$

We now prove the necessity. Let Problem I have a solution in $HHC^{n \times n} \subset HHC^{n \times n}$. By Lemma 3.1, we have

$$(3.8) \quad (X_1, X_2)^H (B_2, B_1) = (B_2, B_1)^H (X_1, X_2)$$

and

$$(3.9) \quad (B_2, B_1)(X_1, X_2)^+(X_1, X_2) = (B_2, B_1).$$

It follows from (3.8) that

$$\begin{pmatrix} X_1^H B_2 & X_1^H B_1 \\ X_2^H B_2 & X_2^H B_1 \end{pmatrix} = \begin{pmatrix} B_2^H X_1 & B_2^H X_2 \\ B_1^H X_1 & B_1^H X_2 \end{pmatrix}.$$

Note that

$$X_1^H B_2 = (P_1 X)^H (P_2 B) = X^H P_1^H P_2 B = X^H (P_1 P_2) B = 0.$$

Similarly, we can prove $B_1^H X_2 = 0$. Therefore, we get $X_1^H B_1 = B_2^H X_2$.

By Lemmas 3.2 and (3.9), we have

$$(B_2 X_1^+ X_1 + B_1 X_2^+ X_1, B_2 X_1^+ X_2 + B_1 X_2^+ X_2) = (B_2, B_1).$$

By an elementary deduction, we get

$$\begin{aligned} X_1^+ X_2 &= X_1^+ X_1 X_1^+ X_2 = X_1^+ (X_1 X_1^+)^H X_2 \\ &= X_1^+ (X_1^+)^H (X_1^H X_2) = X_1^+ (X_1^+)^H (X^H P_1 P_2 X) = 0. \end{aligned}$$

Similarly, we have $X_2^+ X_1 = 0$. Consequently, we obtain $B_2 X_1^+ X_1 = B_2$ and $B_1 X_2^+ X_2 = B_1$.

We turn to verifying the sufficiency. Suppose that (3.4) and (3.5) hold. Let

$$(3.10) \quad A_1 = A_0 + A_0^H Q,$$

where $A_0 = (B_2, B_1)X^+ = B_2 X_1^+ + B_1 X_2^+$ and $Q = I - XX^+ = I - X_1 X_1^+ - X_2 X_2^+$. Then we have

$$(3.11) \quad A_1^H = A_0^H + Q^H A_0 = A_0^H + Q A_0 = A_0 + A_0^H Q + (Q - I)A_0 + A_0^H (I - Q).$$

Observe that P_1 and P_2 are orthogonal projection matrices. It follows from the fact that $B_i = P_i B$ and $X_i = P_i X$, $i = 1, 2$, that $B_1^H X_2 = 0$ and $B_2^H X_1 = 0$. We also have from (3.4) that

$$\begin{aligned} (Q - I)A_0 + A_0^H (I - Q) &= (-X_1 X_1^+ - X_2 X_2^+)(B_2 X_1^+ + B_1 X_2^+) + [(X_1^+)^H B_2^H + (X_2^+)^H B_1^H](X_1 X_1^+ + X_2 X_2^+) \\ &= -X_1 X_1^+ B_2 X_1^+ - X_1 X_1^+ B_1 X_2^+ - X_2 X_2^+ B_2 X_1^+ - X_2 X_2^+ B_1 X_2^+ + (X_1^+)^H B_2^H X_1 X_1^+ \\ &\quad + (X_1^+)^H B_2^H X_2 X_2^+ + (X_2^+)^H B_1^H X_1 X_1^+ + (X_2^+)^H B_1^H X_2 X_2^+ \\ &= -(X_1 X_1^+)^H B_2 X_1^+ - X_1 X_1^+ B_1 X_2^+ - X_2 X_2^+ B_2 X_1^+ - (X_2 X_2^+)^H B_1 X_2^+ \\ &\quad + (X_1^+)^H X_1^H B_1 X_2^+ + (X_2^+)^H X_2^H B_2 X_1^+ \\ &= -(X_1^+)^H X_1^H B_2 X_1^+ - X_1 X_1^+ B_1 X_2^+ - X_2 X_2^+ B_2 X_1^+ - (X_2^+)^H X_2^H B_1 X_2^+ \\ &\quad + (X_1 X_1^+)^H B_1 X_2^+ + (X_2 X_2^+)^H B_2 X_1^+ \\ &= 0. \end{aligned}$$

By using (3.11), it is easy to get $A_1^H = A_0 + A_0^H Q = A_1$. This implies $A_1 \in HC^{n \times n}$.

On the other hand, it is not difficult to deduce that

$$(iJX_1)^+ = iX_1^+ J, (iJX_2)^+ = iX_2^+ J, iJB_1 = B_1, iJB_2 = -B_2, iJX_1 = X_1, iJX_2 = -X_2.$$

Note that

$$\begin{aligned} JA_0J &= JB_2X_1^+J + JB_1X_2^+J \\ &= -(iJB_2)(iJX_1)^+ - (iJB_1)(iJX_2)^+ \\ &= -(-B_2)X_1^+ - B_1(-X_2)^+ \\ &= B_2X_1^+ + B_1X_2^+ = A_0. \end{aligned}$$

Similarly, we can show that $JQJ = -Q$, $Q^H = Q$. Therefore, we get $Q \in HSHC^{n \times n}$. Since $J^T = -J$, we get $JA_0^HJ = A_0^H$. Consequently, we obtain

$$\begin{aligned} JA_1J &= JA_0J + JA_0^HQJ = A_0 - JA_0^HJJQJ \\ &= A_0 - A_0^H(-Q) = A_0 + A_0^HQ = A_1 = A_1^H. \end{aligned}$$

This means $A_1 \in HHC^{n \times n}$. Moreover, it follows from (3.5) and the equalities $X_1^+X_2 = 0$ and $X_2^+X_1 = 0$ that

$$\begin{aligned} A_1(X_1, X_2) &= A_0(X_1, X_2) + A_0^HQ(X_1, X_2) \\ &= (B_2X_1^+X_1 + B_1X_2^+X_1, B_2X_1^+X_2 + B_1X_2^+X_2) \\ &\quad + A_0^H(X_1 - X_1X_1^+X_1 - X_2X_2^+X_1, X_2 - X_1X_1^+X_2 - X_2X_2^+X_2) \\ &= (B_2X_1^+X_1, B_1X_2^+X_2) = (B_2, B_1). \end{aligned}$$

This means that A_1 is a solution of $A(X_1, X_2) = (B_2, B_1)$ in $HHC^{n \times n}$. Therefore, A_1 is also a solution of the $AX = B$ in $HHC^{n \times n}$.

Let S be the solution set of Problem I in $HHC^{n \times n}$. It is obvious that $S_{X,B} \subseteq S$. From the proof of the sufficiency, it is not difficult to get that $Q \in HSHC^{n \times n}$, $A_1 = A_0 + A_0^HQ \in HHC^{n \times n}$. It is also easy to see from Lemma 2.3 and Theorem 2.2 that a necessary and sufficient condition for $S_{X,B} = S$ is $E \in HHC^{n \times n}$. Therefore, $S_{X,B} = \{A | A = A_0 + A_0^HQ + QEQ \ \forall E \in HHC^{n \times n}\}$. The proof is then complete. $\square$

4. The solution of Problem II. It is clear from (3.6) that the solution set $S_{X,B}$ of Problem I is a closed convex set. Hence, for any $A^* \in C^{n \times n}$, the optimal approximation solution exists.

THEOREM 4.1. Let $A^* \in C^{n \times n}$. Then, the optimal approximation solution of Problem II is given by

$$(4.1) \quad \hat{A} = A_0 + A_0^HQ + QA_1^*Q,$$

where

$$A_0 = B_2X_1^+ + B_1X_2^+, Q = I - X_1X_1^+ - X_2X_2^+, A_1^* = \frac{1}{4}[A^* + A^{*H} + J(A^* + A^{*H})J].$$

Let $Q_1 = I - Q$. It is easy to see that both matrices Q and Q_1 are orthogonal projection matrices and satisfy $QQ_1 = 0$, $Q_1 \in HSHC^{n \times n}$. By Theorem 2.1, there is a unique triple matrix (A_1^*, A_2^*, A_3^*) such that $A_1^* \in HHC^{n \times n}$, $A_2^* \in HSHC^{n \times n}$, $A_3^* \in AHC^{n \times n}$, and $A^* = A_1^* + A_2^* + A_3^*$. Moreover, the following relations hold:

$$\langle A_1^*, A_2^* \rangle = 0, \langle A_1^*, A_3^* \rangle = 0, \langle A_2^*, A_3^* \rangle = 0, A_1^* = \frac{1}{4}[A^* + A^{*H} + J(A^* + A^{*H})J].$$

For any $A \in S_{X,B}$, we have $A = A_0 + A_0^H Q + QEQ$, $E \in HHC^{n \times n}$. Moreover,

$$\begin{aligned} \|A^* - A\|^2 &= \|A_2^*\|^2 + \|A_3^*\|^2 + \|A_1^* - A_0 - A_0^H Q - QEQ\|^2 \\ &= \|A_2^*\|^2 + \|A_3^*\|^2 + \|(A_1^* - A_0)Q_1\|^2 + \|(A_1^* - A_0)Q - A_0^H Q - QEQ\|^2 \\ &= \|A_2^*\|^2 + \|A_3^*\|^2 + \|(A_1^* - A_0)Q_1\|^2 + \|Q_1(A_1^* - A_0 - A_0^H)Q\|^2 \\ &\quad + \|Q(A_1^* - A_0 - A_0^H)Q - QEQ\|^2. \end{aligned}$$

Therefore, problem $\min_{A \in S_{X,B}} \|A^* - A\|$ is equivalent to the following problem:

$$(4.2) \quad \min_{E \in HHC^{n \times n}} \|Q(A_1^* - A_0 - A_0^H)Q - QEQ\|.$$

Since $X_1^+ X_2 = 0$, $X_2^+ X_1 = 0$, we deduce that

$$\begin{aligned} QA_0Q &= Q(B_2X_1^+ + B_1X_2^+)(I - X_1X_1^+ - X_2X_2^+) \\ &= Q(B_2X_1^+ - B_2X_1^+X_1X_1^+ - B_2X_1^+X_2X_2^+ \\ &\quad + B_1X_2^+ - B_1X_2^+X_1X_1^+ - B_1X_2^+X_2X_2^+) \\ &= 0. \end{aligned}$$

Noting that $Q^H = Q$, we have $QA_0^H Q = 0$. Thus, we get $Q(A_0 + A_0^H)Q = 0$. This implies that problem (4.2) is equivalent to the following problem:

$$(4.3) \quad \min_{E \in HHC^{n \times n}} \|QA_1^*Q - QEQ\|.$$

It is easy to see that the solution of (4.3) is

$$(4.4) \quad \hat{E} = A_1^* + Q_1EQ_1 \quad \forall E \in HHC^{n \times n}.$$

Substituting (4.4) into (3.6) and taking into account $QQ_1 = 0$, we get (4.1). This completes the proof. $\square$

Acknowledgment. The authors would like to thank the anonymous referees for their helpful comments on the paper.

REFERENCES

- [1] M. BARUCH AND I. Y. BAR-ITZHACK, *Optimal weighted orthogonalization of measured modes*, AIAA J., 16 (1978), pp. 345–351.
- [2] M. BARUCH, *Optimization procedure to correct stiffness and flexibility matrices using vibration tests*, AIAA J., 16 (1978), pp. 1208–1210.
- [3] M. BARUCH, *Optimal correction of mass and stiffness matrices using measured modes*, AIAA J., 20 (1982), pp. 1623–1626.
- [4] F. S. WEI, *Stiffness matrix correction from incomplete test data*, AIAA J., 18 (1980), pp. 1274–1275.
- [5] A. BERMAN, *Mass matrix correction using an incomplete set of measured modes*, AIAA J., 17 (1979), pp. 1147–1148.
- [6] A. BERMAN AND E. J. NAGY, *Improvement of a large analytical model using test data*, AIAA J., 21 (1983), pp. 1168–1173.
- [7] J. GROB, *Explicit solutions to the matrix inverse problem $AX = B$* , Linear Algebra Appl., 289 (1999), pp. 131–134.

- [8] C. G. KHATRI AND S. K. MITRA, *Hermitian and nonnegative definite solutions of linear matrix equations*, SIAM J. Appl. Math., 31 (1976), pp. 579–585.
- [9] L. WU AND B. CAIN, *The re-nonnegative definite solutions to the matrix inverse problem $AX = B$* , Linear Algebra Appl., 245 (1996), pp. 171–190.
- [10] H. DAI AND P. LANCASTER, *Linear matrix equation from an inverse problem of vibration theory*, Linear Algebra Appl., 246 (1996), pp. 31–47.
- [11] F. J. H. DON, *On the symmetric solutions of a linear matrix equation*, Linear Algebra Appl., 93 (1987), pp. 1–7.
- [12] D. X. XIE, L. ZHANG, AND X. Y. HU, *Least-squares solutions of inverse problems for bisymmetric matrices*, Math. Numer. Sin., 22 (2000), pp. 29–40.
- [13] L. ZHANG, *A kind of inverse problem of matrices and its numerical solution*, Math. Numer. Sin., 9 (1987), pp. 431–437.
- [14] L. ZHANG, *The solvability conditions for the inverse problem of symmetric nonnegative definite matrices*, Math. Numer. Sin., 11 (1989), pp. 337–343.
- [15] J. G. SUN, *Two kinds of inverse eigenvalue problem for real symmetric matrices*, Math. Numer. Sin., 10 (1988), pp. 282–290.
- [16] J. G. SUN, *Backward perturbation analysis of certain characteristic subspaces*, Numer. Math., 65 (1993), pp. 357–382.
- [17] T. L. HAYDEN AND J. WELLS, *Approximation by matrices positive semidefinite on a subspace*, Linear Algebra Appl., 109 (1988), pp. 115–130.
- [18] A. JAMESON AND E. KREINDLER, *Inverse problem of linear optimal control*, SIAM J. Control Optim., 11 (1973), pp. 1–19.
- [19] M. JAMSHIDI, *An overview on the solutions of the algebraic matrix Riccati equation and related problems*, Large Scale Systems, 1 (1980), pp. 167–192.
- [20] H. K. WIMMER, *Decomposition and parametrization of semidefinite solutions of the continuous-time algebraic Riccati equation*, SIAM J. Control Optim., 32 (1994), pp. 995–1007.
- [21] A. BEN-ISRAEL AND T. N. E. GREVILLE, *Generalized Inverse: Theory and Applications*, John Wiley and Sons, New York, 1974.
- [22] G. H. GOLUB AND C. F. VAN LOAN, *Matrix Computations*, 2nd ed., The Johns Hopkins University, Baltimore, MD, 1989.

JORDAN CANONICAL FORM OF THE GOOGLE MATRIX: A POTENTIAL CONTRIBUTION TO THE PAGERANK COMPUTATION*

STEFANO SERRA-CAPIZZANO†

Abstract. We consider the web hyperlink matrix used by Google for computing the PageRank whose form is given by $A(c) = [cP + (1 - c)E]^T$, where P is a row stochastic matrix, E is a row stochastic rank one matrix, and $c \in [0, 1]$. We determine the analytic expression of the Jordan form of $A(c)$ and, in particular, a rational formula for the PageRank in terms of c . The use of extrapolation procedures is very promising for the efficient computation of the PageRank when c is close or equal to 1.

Key words. Google matrix, canonical Jordan form, extrapolation formulae

AMS subject classifications. 65F10, 65F15, 65Y20

DOI. 10.1137/S0895479804441407

1. Introduction. We look at the web as a huge directed graph whose nodes are all the web pages and whose edges are constituted by all the links between pages. In the following $\deg(i)$ denotes the cardinality of the pages which are reached by a direct link from page i . The basic Google matrix P is defined as $P_{i,j} = 1/\deg(i)$ if $\deg(i) > 0$ and there exists a link in the web from page i to a certain page $j \neq i$; for the rows i for which $\deg(i) = 0$ we assume $P_{i,j} = 1/n$, where n is the size of the matrix, i.e., the cardinality of all the web pages. This definition is a model for the behavior of a generic web user: if the user is visiting page i , then with probability $1/\deg(i)$ he will move to one of the pages j linked by i and if i has no links, then the user will make just a random choice with uniform distribution $1/n$. The basic PageRank is an n -sized vector which gives a measure of the importance of every page in the web: a simple reasoning shows that the basic PageRank is the left eigenvector of P associated to the dominating eigenvalue 1 (see, e.g., [9, 6]). Since the matrix P is nonnegative and has row sum equal to 1 it is clear that the right eigenvector related to 1 is $\mathbf{e}$ (the vector of all 1's) and that all the other eigenvalues are in modulus at most equal to 1. The structure of P is such that we have no guarantee for its aperiodicity and for its irreducibility: therefore the gap between 1 and the modulus of the second largest eigenvalue can be zero. This means that the computation of the PageRank by the application of the standard power method (see, e.g., [3]) to the matrix $A = P^T$ (or one of its variations for our specific problem) is not convergent or is very slowly convergent. A solution is found by considering a change in the model: given a value $c \in [0, 1]$, from the basic Google matrix P we define the parametric Google matrix $P(c)$ as $cP + (1 - c)E$ with $E = \mathbf{e}\mathbf{v}^T$ and $v_i > 0$, $\|\mathbf{v}\|_1 = 1$. This change corresponds to the following user behavior: if the user is visiting page i , then the next move will be with probability c according to the rule described by the basic Google matrix P and with probability $1 - c$ according to the rule described by $\mathbf{v}$. Generally a value

*Received by the editors February 20, 2004; accepted for publication (in revised form) by M. Chu February 21, 2005; published electronically September 15, 2005. This work was partially supported by MIUR grant 2004015437.

<http://www.siam.org/journals/simax/27-2/44140.html>

†Dipartimento di Fisica e Matematica, Università dell'Insubria, Sede di Como, Via Valleggio 11, 22100 Como, Italy (stefano.serrac@uninsubria.it, serra@mail.dm.unipi.it).

of c as 0.85 is considered in the literature (see, e.g., [6]). For $c < 1$, the good news is that the $\text{PageRank}(c)$, i.e., the left dominating eigenvector, can be computed in a fast way since $P(c)$ (which is now with row sum 1, nonnegative, irreducible, and aperiodic) has a second eigenvalue whose modulus is dominated by c [4]: therefore the convergence to $\text{PageRank}(c)$ is such that the error at step k decays as c^k . Of course the computation becomes slow if c is chosen close to 1 and there is no guarantee of convergence if $c = 1$. In this paper, given the Jordan canonical form of the basic Google matrix P , we describe in an analytical way the Jordan canonical form of the Google matrix $P(c)$ and, in particular, we obtain that

$$\text{PageRank}(c) = \text{PageRank} + R(c)$$

with $R(c)$ the rational vector function of c . Since $\text{PageRank}(c)$ can be computed efficiently when c is far away from 1, the use of vector extrapolation methods [1] should allow us to compute in a fast way $\text{PageRank}(c)$ when c is close or equal to 1 (see [2] for more details).

2. Closed form analysis of $\text{PageRank}(c)$. The analysis is given in two steps: first we give the Jordan canonical form and the rational expression of $\text{PageRank}(c)$ under the assumption that P is diagonalizable; then we consider the general case.

THEOREM 2.1. Let P be an $n \times n$ matrix with $n \geq 1$, $c \in (0, 1)$, $E = \mathbf{e}\mathbf{v}^T$, $\mathbf{v} \in \mathbb{R}^n$, $\mathbf{e} \in \mathbb{R}^n$, $\mathbf{e}_i > 0$, $\|\mathbf{v}\|_1 = 1$, $P(c) = cP + (1-c)E$, $P = X\text{diag}(1, \lambda_2, \dots, \lambda_n)X^{-1}$, $X = [\mathbf{e}|\mathbf{x}_2|\dots|\mathbf{x}_n]$, $[X^{-1}]^T = [\mathbf{y}_1|\mathbf{y}_2|\dots|\mathbf{y}_n]$

$$P(c) = Z\text{diag}(1, c\lambda_2, \dots, c\lambda_n)Z^{-1}, \quad Z = XR^{-1}.$$

- $1 \geq |\lambda_2| \geq \dots \geq |\lambda_n|$, $\lambda_2 = 1$, P is reducible
- $\lambda_2 \neq 1$, P is irreducible

$$R = I_n + \mathbf{e}_1\mathbf{w}^T, \quad \mathbf{w}^T = (0, w_2, \dots, w_n),$$

$$w_j = (1-c)\mathbf{v}^T\mathbf{x}_j/(1-c\lambda_j), \quad j = 2, \dots, n.$$

From the assumptions we have $P = X\text{diag}(\lambda_1, \lambda_2, \dots, \lambda_n)X^{-1}$, but $P\mathbf{e} = \mathbf{e}$ (P has row sum equal to 1) and P is nonnegative; therefore $X = [\mathbf{x}_1|\mathbf{x}_2|\dots|\mathbf{x}_n]$ with $\mathbf{x}_1 = \mathbf{e}$, $\lambda_1 = 1$, and $|\lambda_j| \leq 1 = \|P\|_\infty$; moreover, $\lambda_2 = 1$ if P is reducible and its graph has at least two irreducible closed sets by standard Markov theory (see, e.g., [4]). Consequently, $[\mathbf{y}_1|\mathbf{y}_2|\dots|\mathbf{y}_n]^T X = X^{-1}X = I_n$ and, in particular, $[\mathbf{y}_1|\mathbf{y}_2|\dots|\mathbf{y}_n]^T \mathbf{e} = X^{-1}\mathbf{e} = \mathbf{e}_1$ (the first vector of the canonical basis). We have

$$P(c) = cP + (1-c)\mathbf{e}\mathbf{v}^T = X\text{diag}(c, c\lambda_2, \dots, c\lambda_n)X^{-1} + (1-c)\mathbf{e}\mathbf{v}^T,$$

and hence

$$\begin{aligned} X^{-1}P(c)X &= \text{diag}(c, c\lambda_2, \dots, c\lambda_n) + (1-c)X^{-1}\mathbf{e}\mathbf{v}^TX \\ &= \text{diag}(c, c\lambda_2, \dots, c\lambda_n) + (1-c)\mathbf{e}_1\mathbf{v}^TX \\ &= \text{diag}(c, c\lambda_2, \dots, c\lambda_n) + (1-c)\mathbf{e}_1[\mathbf{v}^T\mathbf{e}, \mathbf{v}^T\mathbf{x}_2, \dots, \mathbf{v}^T\mathbf{x}_n]. \end{aligned}$$

But $\mathbf{v}^T \mathbf{e} = 1$ since $\mathbf{v}$ is a probability vector by the hypothesis. In conclusion we have

$$(2.1) \quad X^{-1}P(c)X = \begin{bmatrix} 1 & (1-c)\mathbf{v}^T \mathbf{x}_2 & \cdots & (1-c)\mathbf{v}^T \mathbf{x}_{n-1} & (1-c)\mathbf{v}^T \mathbf{x}_n \\ & c\lambda_2 & & & \\ & & \ddots & & \\ & & & c\lambda_{n-1} & \\ & & & & c\lambda_n \end{bmatrix}.$$

The last step is to diagonalize the previous matrix: calling R the matrix

$$(2.2) \quad R = \begin{bmatrix} 1 & \frac{(1-c)\mathbf{v}^T \mathbf{x}_2}{(1-c\lambda_2)} & \cdots & \frac{(1-c)\mathbf{v}^T \mathbf{x}_{n-1}}{(1-c\lambda_{n-1})} & \frac{(1-c)\mathbf{v}^T \mathbf{x}_n}{(1-c\lambda_n)} \\ & 1 & & & \\ & & \ddots & & \\ & & & 1 & \\ & & & & 1 \end{bmatrix}$$

and taking into account (2.1), a direct computation shows that

$$R[X^{-1}P(c)X] = \text{diag}(1, c\lambda_2, \dots, c\lambda_n)R,$$

i.e.,

$$X^{-1}P(c)X = R^{-1}\text{diag}(1, c\lambda_2, \dots, c\lambda_n)R,$$

and finally $P(c) = Z\text{diag}(1, c\lambda_2, \dots, c\lambda_n)Z^{-1}$, $Z = XR^{-1}$. $\square$

COROLLARY 2.2. *Let P be a stochastic matrix and let $c \in (0, 1)$. Then $\text{PageRank}(c)$ is the transpose of the first row of the matrix $Z^{-1} = RX^{-1}$.*

$$(2.3) \quad [\text{PageRank}(c)]^T = \mathbf{y}_1^T + (1-c) \sum_{j=2}^n \mathbf{v}^T \mathbf{x}_j \mathbf{y}_j^T / (1-c\lambda_j),$$

where $\mathbf{y}_1^T$ is the first row of Z^{-1} and $\mathbf{y}_j^T$ is the j -th row of Z^{-1} for $j = 2, \dots, n$.

For $c = 1$ there is nothing to prove since $P(c) = P$ and therefore $\text{PageRank}(c) = \text{PageRank}$. We take $c < 1$. By Theorem 2.1 we have that $\text{PageRank}(c)$ is the transpose of the first row of the matrix $Z^{-1} = RX^{-1}$.

Since $X^{-1} = [\mathbf{y}_1 | \mathbf{y}_2 | \cdots | \mathbf{y}_n]^T$, the claimed thesis follows from the structure of R in (2.2). $\square$

We now take into account the case where P is general (and therefore possibly not diagonalizable): the conclusions are formally identical except for the rational expression $R(c)$ which is a bit more involved. We first observe that if $\lambda_2 = 1$, then, as proved in [4], the graph of P has at least two irreducible closed sets: therefore the geometric multiplicity of the eigenvalue 1 also must be at least 2. In summary in the general case we have $P = XJX^{-1}$, where

$$(2.4) \quad J = \begin{bmatrix} 1 & & & & \\ & \lambda_2 & * & & \\ & & \ddots & & \\ & & & \lambda_{n-1} & * \\ & & & & \lambda_n \end{bmatrix}$$

with $*$ denoting a value that can be 0 or 1.

THEOREM 2.3. *Let P be a stochastic matrix of order n and $c \in (0, 1)$. Let $E = \mathbf{e}\mathbf{v}^T$ with $\mathbf{v} = (v_1, \dots, v_n)^T$ and $\mathbf{e} = (1, \dots, 1)^T$ with $v_i > 0$ and $\|\mathbf{v}\|_1 = 1$. Let $P(c) = cP + (1 - c)E$. Let $P = XJ(1)X^{-1}$ with $X = [\mathbf{e}|\mathbf{x}_2|\dots|\mathbf{x}_n]$, $[X^{-1}]^T = [\mathbf{y}_1|\mathbf{y}_2|\dots|\mathbf{y}_n]$*

$$J(c) = \begin{bmatrix} 1 & & & \\ & c\lambda_2 & & c \cdot * \\ & & \ddots & \\ & & & c\lambda_{n-1} & c \cdot * \\ & & & & c\lambda_n \end{bmatrix},$$

$$(2.5) \quad J(c) = D \begin{bmatrix} 1 & & & \\ & c\lambda_2 & & * \\ & & \ddots & \\ & & & c\lambda_{n-1} & * \\ & & & & c\lambda_n \end{bmatrix} D^{-1}, \quad D = \text{diag}(1, c, \dots, c^{n-1}),$$

where $*$ is either $c\lambda_i$ or c and $0 \leq \lambda_i \leq 1$.

$$P(c) = ZJ(c)Z^{-1}, \quad Z = XR^{-1},$$

where R is a nonsingular matrix satisfying the following conditions:

- $1 \geq |\lambda_2| \geq \dots \geq |\lambda_n|$ and $\lambda_2 = 1$, P is aperiodic and aperiodicity index is 1;
- $\mathbf{v}^T \mathbf{x}_j = 0$ for $j = 2, \dots, n$.

$$R = I_n + \mathbf{e}_1 \mathbf{w}^T, \quad \mathbf{w}^T = (0, w_2, \dots, w_n),$$

$$(2.6) \quad w_2 = (1 - c) \mathbf{v}^T \mathbf{x}_2 / (1 - c\lambda_2),$$

$$(2.7) \quad w_j = [(1 - c) \mathbf{v}^T \mathbf{x}_j + [J(c)]_{j-1,j} w_{j-1}] / (1 - c\lambda_j), \quad j = 3, \dots, n.$$

From the assumptions we have $P = X \text{diag}(\lambda_1, \lambda_2, \dots, \lambda_n) X^{-1}$, but $P\mathbf{e} = \mathbf{e}$ (P has row sum equal to 1) and P is nonnegative: therefore $X = [\mathbf{x}_1|\mathbf{x}_2|\dots|\mathbf{x}_n]$ with $\mathbf{x}_1 = \mathbf{e}$, $\lambda_1 = 1$, and $|\lambda_j| \leq 1 = \|P\|_\infty$; moreover, $\lambda_2 = 1$ if the graph of P has at least two irreducible closed sets by standard Markov theory (see, e.g., [4]). Consequently, $[\mathbf{y}_1|\mathbf{y}_2|\dots|\mathbf{y}_n]^T X = X^{-1}X = I_n$ and, in particular, $[\mathbf{y}_1|\mathbf{y}_2|\dots|\mathbf{y}_n]^T \mathbf{e} = X^{-1}\mathbf{e} = \mathbf{e}_1$ (the first vector of the canonical basis). From the relation

$$P(c) = cP + (1 - c)\mathbf{e}\mathbf{v}^T = X \text{diag}(c, c\lambda_2, \dots, c\lambda_n) X^{-1} + (1 - c)\mathbf{e}\mathbf{v}^T,$$

we deduce

$$\begin{aligned} X^{-1}P(c)X &= cJ(1) + (1 - c)X^{-1}\mathbf{e}\mathbf{v}^T X \\ &= cJ(1) + (1 - c)\mathbf{e}_1 \mathbf{v}^T X \\ &= cJ(1) + (1 - c)\mathbf{e}_1 [\mathbf{v}^T \mathbf{e}, \mathbf{v}^T \mathbf{x}_2, \dots, \mathbf{v}^T \mathbf{x}_n]. \end{aligned}$$

But $\mathbf{v}^T \mathbf{e} = 1$ since $\mathbf{v}$ is a probability vector by the hypothesis. In summary we infer

$$(2.8) \quad X^{-1}P(c)X = \begin{bmatrix} 1 & (1-c)\mathbf{v}^T \mathbf{x}_2 & \cdots & (1-c)\mathbf{v}^T \mathbf{x}_{n-1} & (1-c)\mathbf{v}^T \mathbf{x}_n \\ & c\lambda_2 & c \cdot * & & \\ & & \ddots & & \\ & & & c\lambda_{n-1} & c \cdot * \\ & & & & c\lambda_n \end{bmatrix}.$$

The last step is to diagonalize the previous matrix: setting R the matrix

$$(2.9) \quad \begin{bmatrix} 1 & w_2 & \cdots & w_{n-1} & w_n \\ & 1 & & & \\ & & \ddots & & \\ & & & 1 & \\ & & & & 1 \end{bmatrix},$$

with w_j as in (2.6)–(2.7) and using (2.8), a direct computation shows that

$$R[X^{-1}P(c)X] = J(c)R,$$

i.e.,

$$X^{-1}P(c)X = R^{-1}J(c)R,$$

and therefore $P(c) = ZJ(c)Z^{-1}$, $Z = XR^{-1}$. As a final remark we observe that the identity in (2.5) can be proved by direct inspection. $\square$

COROLLARY 2.4. *Let $\mathbf{v}$ be the left eigenvector of $P(c)$ corresponding to the eigenvalue 1. Then*

$$(2.10) \quad [\text{PageRank}(c)]^T = \mathbf{y}_1^T + \sum_{j=2}^n w_j \mathbf{y}_j^T,$$

where $\mathbf{y}_1^T, \mathbf{y}_2^T, \dots, \mathbf{y}_n^T$ are the right eigenvectors of $P(c)$ corresponding to the eigenvalues $1, c, c^2, \dots, c^{n-1}$ respectively, and w_j are the weights defined in (2.6)–(2.7).

By Theorem 2.3, the decomposition

$$P(c) = ZJ(c)Z^{-1}, \quad Z = XR^{-1},$$

with $J(c)$ as in (2.5), is a Jordan decomposition where

$$\text{diag}(1, c^{-1}, \dots, c^{1-n})Z^{-1}$$

is the left eigenvector matrix. Therefore $[\text{PageRank}(c)]^T$ is the first row of the matrix

$$\text{diag}(1, c^{-1}, \dots, c^{1-n})Z^{-1} = \text{diag}(1, c^{-1}, \dots, c^{1-n})RX^{-1}.$$

Since $X^{-1} = [\mathbf{y}_1 | \mathbf{y}_2 | \cdots | \mathbf{y}_n]^T$, the claimed thesis follows from the structure of R in (2.9) and from the fact that the first row of $\text{diag}(1, c^{-1}, \dots, c^{1-n})$ is $\mathbf{e}_1^T$. $\square$

3. Discussion and future work. The theoretical results presented in this paper have two main consequences: (a) The vector $\text{PageRank}(c)$ can be very different from $\text{PageRank}(1)$ since the number n appearing in (2.3) and (2.10) is huge, being the whole number of the web pages. This shows that a small change in the value of c (say from 1 to $1 - \epsilon$ for a given fixed small $\epsilon > 0$) gives a dramatic change in the vector: the latter statement also agrees with the conditioning of the numerical problem described in [5], where it is shown that the conditioning of the computation of $\text{PageRank}(c)$ grows as $(1 - c)^{-1}$. In relation to the work of Kamvar and others, it should be mentioned that the proof in [4] of the behavior of the second eigenvalue of the Google matrix turns out to be quite elementary by following the argument in part (b), exercise 7.1.17, of [8]. It is possible that a similar argument could help in simplifying even more the proofs of Theorems 2.1 and 2.3. (b) A strong challenge posed by the formulae (2.3) and (2.10) is the possibility of using vector extrapolation [1] for obtaining the expression of $\text{PageRank}(1)$ using few evaluations of $\text{PageRank}(c)$ for some values of c far enough from 1 so that the computational problem is well-conditioned [5] and the known algorithms based on the classical power method are efficient [6, 7]; this subject is under investigation in [2].

Now we have to discuss a more philosophical question. We presented a potential strategy (see [2] for more details) for the computation of $\text{PageRank} = \text{PageRank}(1)$ through a vector extrapolation technique applied to (2.10). A basic and preliminary criticism is that in practice (i.e., in the real Google matrix) the matrix $A(1) = A$ is reducible: therefore the nonnegative dominating eigenvector with unitary l^1 norm is not unique, while the nonnegative dominating eigenvector $\text{PageRank}(c)$ with unitary l^1 norm of $A(c)$ with $c \in [0, 1)$ is not only unique but strictly positive. Therefore, since $\text{PageRank}(c)$ is a rational expression without poles at $c = 1$, there exists (unique) the limit as c tends to 1 of $\text{PageRank}(c)$. Clearly this vector is unique and, consequently, our proposal concerns a special PageRank problem. We have to face two problems: how to characterize this limit vector and whether this special PageRank vector has a specific meaning. To understand the situation and to give answers, we have to make a careful study of the set of all the (normalized) PageRank vectors for $c = 1$. From the theory of stochastic matrices, the matrix A is similar (through a permutation matrix) to

$$\hat{A} = \begin{bmatrix} P_{1,1} & 0 & & & 0 \\ P_{2,1} & P_{2,2} & 0 & & 0 \\ \vdots & & \ddots & & \vdots \\ P_{r,1} & \cdots & \cdots & P_{r,r} & 0 \\ 0 & & & 0 & P_{r+1,r+1} & 0 & 0 \\ 0 & & & 0 & 0 & P_{r+2,r+2} & 0 & 0 \\ \vdots & & & & & \ddots & \vdots \\ 0 & & & & & 0 & P_{m,m} \end{bmatrix},$$

where $P_{i,i}$ is either a null matrix or is strictly substochastic and irreducible for $i = 1, \dots, r$, and $P_{j,j}$ is stochastic and irreducible for $j = r + 1, \dots, m$. In the case of the Google matrix we have $m - r$ quite large. Therefore we have the following:

- $\lambda = 1$ is an eigenvalue of $\hat{A}$ (and hence of A) with algebraic and geometric multiplicity equal to $m - r$ (exactly 1 for every $P_{j,j}$ with $j = r + 1, \dots, m$, and every other eigenvalue of unitary modulus has geometric multiplicity coinciding with the algebraic multiplicity);

- all the remaining eigenvalues of $\hat{A}$ (and hence of A) have modulus weakly dominated by 1 since each matrix $P_{i,i}$ for $i = 1, \dots, r$ is either a null matrix or is strictly substochastic and irreducible;
- the canonical basis of the dominating eigenspace is given by $\{\mathbf{Z}[i] \geq 0, i = 1, \dots, t = m - r\}$ with

$$\hat{A}\mathbf{Z}[i] = \hat{A} \begin{bmatrix} 0 \\ \mathbf{z}[i] \\ 0 \end{bmatrix} = \begin{bmatrix} 0 \\ P_{r+i, r+i}\mathbf{z}[i] \\ 0 \end{bmatrix} = \begin{bmatrix} 0 \\ \mathbf{z}[i] \\ 0 \end{bmatrix} = \mathbf{Z}[i]$$

and $\sum_{j=1}^n (\mathbf{Z}[i])_j = 1$, $(\mathbf{Z}[i])_j \geq 0$, for all $j = 1, \dots, n$, for all $i = 1, \dots, t$. In conclusion we are able to characterize all the nonnegative normalized dominating eigenvectors of the Google matrix as

$$\mathbf{v}(\lambda_1, \dots, \lambda_t) = \sum_{i=1}^t \lambda_i \mathbf{Z}[i], \quad \sum_{i=1}^t \lambda_i = 1, \quad \lambda_i \geq 0.$$

Now if we put the above relations into Corollary 2.4, we deduce that (2.10) can be read as

$$[\text{PageRank}(c)]^T = \mathbf{y}_1^T + (\mathbf{v}^T \mathbf{x}_2) \mathbf{y}_2^T + \dots + (\mathbf{v}^T \mathbf{x}_t) \mathbf{y}_t^T + \sum_{j=t+1}^n w_j \mathbf{y}_j^T,$$

with $w_j = w_j(c)$ and $\lim_{c \rightarrow 1} w_j(c) = 0$. Therefore the unique vector $\text{PageRank}(1) = \mathbf{y}_1 + (\mathbf{v}^T \mathbf{x}_2) \mathbf{y}_2 + \dots + (\mathbf{v}^T \mathbf{x}_t) \mathbf{y}_t$ that we compute is one of the vectors $\mathbf{v}(\lambda_1, \dots, \lambda_t) = \sum_{i=1}^t \lambda_i \mathbf{Z}[i]$, $\sum_{i=1}^t \lambda_i = 1$, $\lambda_i \geq 0$. The question is, which $\lambda_1, \dots, \lambda_t$? By comparison with Corollary 2.4, the answer is

$$\lambda_j = \sum_{i=1}^t (\mathbf{v}^T \mathbf{x}_i) \alpha_{i,j},$$

where $\alpha = (\alpha_{i,j})_{i,j=1}^t$ is the transformation matrix from the basis $\{\mathbf{Z}[i] \geq 0, i = 1, \dots, t\}$ to the basis $\{\mathbf{y}_i, i = 1, \dots, t\}$, i.e., $\mathbf{y}_i = \sum_{j=1}^t \alpha_{i,j} \mathbf{Z}[j]$, $i = 1, \dots, t$. It is interesting to observe that the vector that we compute, as limit of $\text{PageRank}(c)$, depends on $\mathbf{v}$, and this is welcome since according to the model, it is correct that the personalization vector $\mathbf{v}$ decides which PageRank vector is chosen.

Acknowledgments. We give warm thanks to the referees for very pertinent and useful remarks and to Michele Benzi, Claude Brezinski, and Michela Redivo Zaglia for interesting discussions.

REFERENCES

- [1] C. BREZINSKI AND M. REDIVO ZAGLIA, *Extrapolation Methods. Theory and Practice*, Stud. Comput. Math., North-Holland, Amsterdam, 1991.
- [2] C. BREZINSKI, M. REDIVO ZAGLIA, AND S. SERRA-CAPIZZANO, *Extrapolation methods for Page-Rank computations*, Les Comptes Rendus de l'Academie de Sciences de Paris Ser. 1, 340 (2005), pp. 393–397.
- [3] G.H. GOLUB AND C.F. VAN LOAN, *Matrix Computations*, Johns Hopkins Ser. Math. Sci. 3, The Johns Hopkins University Press, Baltimore, MD, 1983.
- [4] H. HAVELIWALA AND S.D. KAMVAR, *The Second Eigenvalue of the Google Matrix*, Technical report, Stanford University, Stanford, CA, 2003.

- [5] S.D. KAMVAR AND H. HAVELIWALA, *The Condition Number of the PageRank Problem*, Technical report, Stanford University, Stanford, CA, 2003.
- [6] S.D. KAMVAR, H. HAVELIWALA, C.D. MANNING, AND G.H. GOLUB, *Extrapolation methods for accelerating PageRank computations*, in Proceedings of the 12th International WWW Conference, Budapest, Hungary, 2003.
- [7] S.D. KAMVAR, H. HAVELIWALA, C.D. MANNING, AND G.H. GOLUB, *Exploiting the block structure of the Web for computing PageRank*, SCCM TR.03-9, Stanford University, Stanford, CA, 2003.
- [8] C.D. MEYER, *Matrix Analysis and Applied Linear Algebra*, SIAM, Philadelphia, 2000.
- [9] L. PAGE, S. BRIN, R. MOTWANI, AND T. WINOGRAD, *The PageRank citation ranking: Bringing order to the web*, Stanford Digital Libraries Working Paper, 1998.

STRATEGIES FOR SCALING AND PIVOTING FOR SPARSE SYMMETRIC INDEFINITE PROBLEMS*

IAIN S. DUFF[†] AND STÉPHANE PRALET[‡]

Abstract. We consider ways of implementing preordering and scaling for symmetric systems and show the effect of using this technique with a multifrontal code for sparse symmetric indefinite systems. After having presented a new method for scaling, we propose a way of using an approximation to a symmetric weighted matching to predefine 1×1 and 2×2 pivots prior to the ordering and analysis phase. We also present new classes of orderings called “(relaxed) constrained orderings” that mix structural and numerical criteria.

Key words. direct solver for sparse symmetric indefinite matrices, matching, scaling, ordering

AMS subject classifications. 05C70, 65F05, 65F35, 65F50

DOI. 10.1137/04061043X

1. Introduction. We study techniques for scaling and choosing pivots when using multifrontal methods in the LDL^T factorization of symmetric indefinite matrices, where L is a lower triangular matrix and D is a block diagonal matrix with 1×1 and 2×2 blocks.

Our main contribution is to define a new method for scaling and a way of using an approximation to a symmetric weighted matching to predefine 1×1 and 2×2 pivots prior to the ordering and analysis phase. We also present new classes of orderings called “(relaxed) constrained orderings” that select pivots during the symbolic Gaussian elimination using two graphs: the first one contains information about the structure of the reduced matrix, and the second one gives information about the numerical values.

Prior to the LU factorization of an unsymmetric matrix A , the package **MC64** [10, 11] can be used to get a maximum weighted matching so that the corresponding permutation will place large entries on the diagonal. The matrix can then be scaled so that diagonal entries have modulus one and off-diagonals have modulus less than or equal to one. This has been found to greatly improve the numerical stability of the subsequent LU factorization. If, however, **MC64** is applied to a symmetric matrix, the resulting permutation will not normally preserve symmetry. In this paper, we examine ways in which symmetric preprocessing can be applied and, in particular, how **MC64** can be used while still preserving symmetry. Our preprocessing will be followed by a standard symmetric permutation in order to decrease the fill-in in the factors.

We will use our symmetric preprocessing with a symmetric multifrontal code **MA57** [8] to validate our heuristics on real test problems that we have divided into two sets: the first consists of augmented matrices, and the second contains general indefinite matrices.

In section 2, we present some general characteristics of multifrontal symmetric indefinite solvers that will be useful in understanding our preprocessing strategies and experimental results. In section 3, we describe our experimental environment.

*Received by the editors June 23, 2004; accepted for publication (in revised form) by A. Frommer January 13, 2005; published electronically September 15, 2005.

<http://www.siam.org/journals/simax/27-2/61043.html>

[†]Atlas Centre, RAL, Oxon OX11 0QX, UK (i.s.duff@rl.ac.uk).

[‡]CERFACS, 42, av. G. Coriolis, 31057 Toulouse Cedex 01, France (Stephane.Pralet@cerfacs.fr).

Section 4 shows how **MC64** scaling can be used for the symmetric indefinite case and studies the effects on the factorization of different kinds of scaling. Section 5 discusses orderings to decrease the fill-in and to give good preselected pivots. Our experimental results are given in section 6, and the best strategies are discussed. Finally, in section 7, we summarize our results and consider future improvements.

We make extensive use of routines from HSL [23]. Any code with a name beginning with **MA** or **MC** is from HSL or is a derivative of an HSL code, for example **MC64** or **MA57**.

2. Symmetric indefinite multifrontal solvers and numerical pivoting.

2.1. Multifrontal approach. Multifrontal methods [13, 14] use an elimination tree [26] to represent the dependencies of the computation. Each node of this tree is associated with a frontal matrix that is assembled (summed) by contributions from its children and the original matrix. It is of the form

$$\begin{array}{ccc} & \text{fully summed columns} & \text{partially summed columns} \\ & \downarrow & \downarrow \\ \begin{array}{l} \text{fully summed rows} \rightarrow \\ \text{partially summed rows} \rightarrow \end{array} & \rightarrow & \begin{bmatrix} F_{11} & F_{12} \\ F_{21} & F_{22} \end{bmatrix}. \end{array}$$

Then elimination operations are performed using pivots from within the fully summed block, F_{11} , and the Schur complement or contribution block $F_{22} \leftarrow F_{22} - F_{21}F_{11}^{-1}F_{12}$ is computed. In the symmetric case, $F_{12} = F_{21}^T$, the matrices F_{11} and F_{22} are symmetric, pivots are chosen from the diagonal, as discussed in the following section, and operations and storage are about half that of the general case.

2.2. Numerical pivoting. In the unsymmetric case, at step k of the Gaussian elimination, the pivot (p, q) is selected from the fully summed rows and columns, and the entries a_{ij} of the remaining submatrix are updated:

$$a_{ij}^{(k+1)} \leftarrow a_{ij}^{(k)} - \frac{a_{ip}^{(k)} a_{qj}^{(k)}}{a_{pq}^{(k)}}.$$

To limit the growth of the entries in the factors, and thus to have a more accurate factorization, a test on the magnitude of the pivot is commonly used. a_{pq} can be selected if and only if

$$(2.1) \quad |a_{pq}| \geq u \max_j |a_{pj}|,$$

where u is a threshold between 0 and 1. Thanks to this criterion of selection the growth factor is limited to $1/u$ at each step of the factorization.

In the symmetric indefinite case, we have to perform 1×1 and 2×2 pivoting if we want to keep the symmetry while maintaining stability. Pivot selection can be done using the Bunch–Parlett [4] or Bunch–Kaufman [3] algorithm. In the context of sparse matrices, the criterion of the Duff–Reid algorithm ([13], as modified in [16]) can be used to ensure a growth factor lower than $1/u$ at each step of Gaussian elimination. A 1×1 diagonal pivot can be selected if and only if it satisfies the inequality (2.1). A 2×2 pivot $P = \begin{pmatrix} a_{pp} & a_{pq} \\ a_{qp} & a_{qq} \end{pmatrix}$ can be selected if and only if it satisfies

$$(2.2) \quad |P^{-1}| \begin{pmatrix} \max_k |a_{pk}| \\ \max_k |a_{qk}| \end{pmatrix} \leq \begin{pmatrix} 1/u \\ 1/u \end{pmatrix},$$

where $|P^{-1}|$ denotes the matrix whose values are the absolute values of P^{-1} . During the elimination, it may not be possible to eliminate some fully summed variables. Elimination of these variables must then be delayed to the parent. This has the effect of causing extra fill-in and thus increases the memory and the number of operations. Too many delayed pivots can severely slow down the factorization.

3. Experimental environment.

3.1. Test machine. Our experiments are conducted on one node of a COMPAQ Alpha Server SC45 at CERFACS. There are 4 GBytes of memory shared between 4 EV68 processors per node, and we disable three of the processors so that we can use all the memory of the node with the remaining single processor. We use the Fortran 90 compiler, f90 version 5.5 with -O option. Our use of integer pointers restricts our array size to 2 GBytes. If the factorization needs to allocate an array larger than 2 GBytes or requires more than 30 minutes, we consider the factorization to not be successful.

3.2. Matrices. We conduct our experiments on a number of test problems that we divide into two sets. We decided to test our preprocessings on matrices of order between 10000 and 100000 and to restrict the number of matrices of the same type to two or three in each set in order to avoid class effects. The matrices have an identification number that will be used to represent them on the x -axis of some figures. Except for bloweybl, bloweybq, cvxqp3, and qpband, which come from CUTer and are available from ftp.numerical.rl.ac.uk/pub/matrices/symmetric/indef2/, the matrices are available from ftp.numerical.rl.ac.uk/pub/matrices/symmetric/indef/ and correspond to a subset of the matrices collected by [21] for testing symmetric sparse solvers. The matrices come from the University of Florida collection (UF) [6], the Maros and Mészáros quadratic programming collection (M2) [27], the CUTer optimization test set (CUTer) [20], and Kumpf and Pöthen (KP) [25]. Some problems were generated by Andy Wathen (AW), Mario Arioli (MA), and Miroslav Tuma (MT). The c-* matrices were obtained from Olaf Schenk (OS) and are also available in [6]. These problems are described in Tables 3.1 and 3.2.

Many of the test matrices correspond to augmented matrices of the form

$$\mathcal{K}_{H,A} = \begin{pmatrix} H & A \\ A^T & 0 \end{pmatrix}.$$

We performed some earlier experiments [12] with matrices of this form, where H is the zero matrix. We note that matrices of this form are structurally singular unless A is square nonsingular. The set of matrices described in Table 3.1, which will be referred to as the *AUGSYS* set, are of the above form but with $H \neq 0$. Note that some of them are structurally singular, and we show the $\text{rank}(A)$, which is defined as the number of zeros on the diagonal after applying a maximum transversal algorithm. The second set, which will be referred to as the *GENSYM* set, in Table 3.2 corresponds to general (nonzero diagonal) indefinite symmetric matrices. Note that we also include the BOYD1 matrix, which is an augmented system in this set because its number of constraints (order of the zero block) is negligible.

3.3. Measures and methodology. In our earlier experiments [12], we clearly identified that MA57 was not tailored to augmented matrices with a zero $(1,1)$ block. These were much better handled by MA47 [15], which respects the two zero blocks that exist when 2×2 pivots with two zero entries on the diagonal (referred to as 2×2 pivots) are eliminated. Gould and Scott [21] also found that MA47 and MA67 performed

TABLE 3.1

Augmented matrices of $H \neq 0$ type (AUGSYS set). (x): x is the structural deficiency of the matrix. Id: identification number. $\lambda+$: number of positive eigenvalues. $\lambda-$: number of negative eigenvalues.

Matrix	Id	n	nnz	$\lambda+$	$\lambda-$	Origin
A0NSDSIL	1	80016	200021	50698	29318	Linear complementarity problem (CUTer)
A2NNSNSL	2	80016	196115	51232	28784	Linear complementarity problem (CUTer)
A5ESINDL	3	60008	145004	38978	21030	Linear complementarity problem (CUTer)
AUG3DCQP	4	35543	77829	27543	8000	Expanded system-3D PDE (CUTer)
BLOCKQP1	5	60012	340022	20011	40001	QP with block structure (CUTer)
BLOWEYA	6	30004	90006	20002	10002	Cahn-Hilliard problem (CUTer)
BRAINPC2	7	27607	96601	13807	13800	Biological model (CUTer)
BRATU3D	8	27792	88627	15625	12167	3D Bratu problem (CUTer)
CONT-201	9	80595	239596	40397	40198	KKT matrix-Convex QP (M2)
K1.SAN	10	67759 (1)	303364	46954	20804	Straz pod Ralskem mine model (MT)
NCVXQP1	11	12111	40537	7111	5000	KKT matrix-nonconvex QP (CUTer)
NCVXQP5	12	62500	237483	28534	33966	KKT matrix-nonconvex QP (CUTer)
NCVXQP7	13	87500	312481	37500	50000	KKT matrix-nonconvex QP (CUTer)
SIT100	14	10262	34094	7143	3119	Straz pod Ralskem mine model (MT)
bloweybl	15	30003 (1)	60000	20001	10001	Cahn-Hilliard problem (CUTer)
cvxqp3	16	17500	62481	10000	7500	Convex QP (CUTer)
mario001	17	38434	114643	23130	15304	Stokes equation (MA)
olesnik0	18	88263	402623	61030	27233	Straz pod Ralskem mine model (MT)
qpband	19	20000	30000	15000	5000	QP (CUTer)
stokes128	20	49666	295938	33281	16385	Stokes equation (MA)
stokes64	21	12546	74242	8449	4097	Stokes equation (AW)
tuma1	22	22967	50560	13360	9607	Mine model (MT)
tuma2	23	12992	28440	7515	5477	Mine model (MT)

TABLE 3.2

General symmetric indefinite matrices (GENSYM set). *: BOYD1 is in the GENSYM set because the number of constraints is negligible. Id: identification number. $\lambda+$: number of positive eigenvalues. $\lambda-$: number of negative eigenvalues.

Matrix	Id	n	nnz	$\lambda+$	$\lambda-$	Origin
BOYD1*	24	93279	652246	93261	18	KKT matrix-Convex QP (CUTer)
DIXMAANL	25	60000	179999	54660	5340	Dixon-Maany optimization example (CUTer)
HELM3D01	26	32226	230335	25455	6771	Helmholtz problem (MA)
LINVERSE	27	11999	53988	9161	2838	Matrix inverse approximation (CUTer)
SPMSRTLS	28	29995	129971	14016	15979	Sparse matrix square root (CUTer)
bcstkt35	29	30237	740200	30234	3	Stiffness matrix-automobile seat frame (UF)
bcstkt39	30	46772	1068033	46771	1	Shuttle rocket booster (UF)
bloweybq	31	10001	30000	10001	0	Cahn-Hilliard problem (CUTer)
c-68	32	64810	315403	36546	28264	Optimization model (OS)
c-71	33	76638	468079	44814	31824	Optimization model (OS)
copter2	34	55476	407714	27750	27726	Helicopter rotor blade (KP)
crystk02	35	13965	491274	13964	1	Stiffness matrix-crystal free vibration (UF)
crystk03	36	24696	887937	24695	1	Stiffness matrix-crystal free vibration (UF)
dawson5	37	51537	531157	25743	25794	Aeroplane actuator system (UF)
qa8fk	38	66127	863353	66127	0	FE matrix from 3D acoustics (UF)
vibrobox	39	12328	177578	12325	3	Vibroacoustic problem (UF)

better on these problems and recorded some failures of MA57 on this class. This is why we have decided not to try to improve MA57 on this set of matrices and why we will not use matrices of this type in our experiments. We ran the MA57 factorization with a pivoting threshold equal to 10^{-2} and obliged its analysis to merge two nodes if both require less than 16 eliminations ($\text{ICNTL}(12) = 16$).

We compare our codes using the performance profiling system of [7]. These profiles aim at evaluating and comparing the performance of a set of solvers $\mathcal{S}$ on a test set of problems $\mathcal{P}$. For each solver $i \in \mathcal{S}$ and each problem $j \in \mathcal{P}$, we measure a statistic s_{ij} with the conventions that if the solver did not succeed in solving the problem, then $s_{ij} = \infty$, and that the smaller this statistic the better the code. For example, we can measure the factorization time, the number of nonzeros in the factors: let j be

a problem, and let $\hat{s}_j = \min_{i \in \mathcal{S}} \{s_{ij}\}$ be the best statistic obtained for this problem. For each $\alpha \geq 1$ and each solver $i \in \mathcal{S}$, we define

$$p_i(\alpha) = \frac{\text{size } \{j \in \mathcal{P} \text{ such that } s_{ij} \leq \alpha \hat{s}_j\}}{\text{size } \mathcal{P}}$$

as the fraction of the problem set that is solved by the solver i within a factor α of the best. $p_i(1)$ gives the fraction of examples for which solver i is the best and $\lim_{\alpha \rightarrow \infty} p_i(\alpha)$ gives the fraction of the problems for which the solver succeeded. In our experiments, for each solver i we will plot its performance profile p_i against α .

To compare our different approaches, we will look at the factorization time, the memory needed to complete the factorization, and the reliability of the analysis in terms of memory forecasting. We would like to have an estimation of the required memory after the analysis that is not too far from that actually needed by the factorization. The quality of the analysis prediction will be assessed using the ratio between the memory actually used during the factorization and that predicted by the analysis.

It is standard practice to increase the storage estimated by the analysis and allocate more storage in the hope that the factorization will be successful even if some additional numerical pivoting occurs. The percentage by which we increase the estimate from the analysis will be called the memory relaxation. Clearly, this memory relaxation parameter is important for algorithms that are implemented in Fortran. However, although C or C++ implementations allow dynamic memory allocation, their cost may be not negligible. Finally, note that the quality of the memory estimation is all the more critical in a distributed memory environment. For example, the buffers used for communications need to be well estimated. In our experiments, we use a memory relaxation of 20% and 50%. When we evaluate the analysis prediction, we consider only runs where the factorization can be performed within the relaxed analysis estimation. When we want to evaluate the factorization time without taking into account the reliability of the memory estimations, we allocate 4 GBytes of memory.

First, in section 4, we study the influence of scaling on the MA57 factorization phase. Second, the effect of additional numerical and structural preprocessing are analyzed when they are combined with different orderings (an approximate minimum degree ordering, AMD [1] in 6.2.1, an ordering based on multilevel nested dissection algorithm, MEIS [24] in 6.2.2, and an approximate minimum fill-in ordering, AMF [28, 30] in 6.2.3). The best approaches will be discussed in sections 6.1 and 6.2.4.

4. Scaling. In this section, we examine the effect of scaling and identify the most robust approach. First, we describe existing scaling approaches; second, we propose an alternative based on the symmetrization of an unsymmetric scaling; and, finally, we analyze our experimental results.

4.1. Existing symmetric scalings. Let A be a square symmetric sparse matrix. MC30 [5] scales A so that the scaled matrix DAD has its nonzero entries close to 1 in absolute value. In practice, it minimizes $\sum_{a_{ij} \neq 0} (\log |a_{ij}|)^2$.

If A is structurally nonsingular, MC77 [31] used with the $\|\cdot\|_p$ -norm, $\|\cdot\|_p$, computes a sequence of matrices $D_r^{(k)} A D_c^{(k)}$ that converge to a matrix that has its column and row norms equal to 1 (which corresponds to a doubly stochastic matrix if $p = 1$). If $p \neq \infty$, this limit is unique. If A is symmetric, then this scaling is symmetric ($D_r^{(k)} = D_c^{(k)}$). We will study the effect of MC77 with $p = 1$ and $p = \infty$. These scalings will be identified by MC77one and MC77inf, respectively. Note that when the matrix is

structurally singular, the convergence of the algorithm is not guaranteed. Moreover, the convergence to a doubly stochastic matrix with $p = 1$ is slower than with $p = \infty$. In our experiments, we limit the number of MC77 steps to 20.

4.2. Adaptation of MC64 scaling.

4.2.1. Maximum weighted matching. It is common to represent a symmetric matrix A of order n by a weighted undirected graph given by $G = (V, E)$, where $V = \{1, 2, \dots, n\}$ and each undirected edge (i, j) of E corresponds to the off-diagonal nonzeros a_{ij} and a_{ji} of A and has a weight of $2|a_{ij}|$ if $i \neq j$. If $i = j$, depending on the context we may or may not want to add a self-loop to G of weight $|a_{ii}|$. The matrix can also be represented by a bipartite graph $G = (R, C, E)$, where $(i, j) \in R \times C$ belongs to E if and only if $a_{ij} \neq 0$ and has a weight of $|a_{ij}|$. In the following we will associate a weight ω with a set of edges $\mathcal{S}$, defined by

$$(4.1) \quad \omega(\mathcal{S}) = \prod_{(i,j) \in \mathcal{S}} |e_{ij}|, \text{ where } |e_{ij}| \text{ is the weight of the edge } (i, j),$$

where we use a product rather than a sum for compatibility with the recommended use of MC64 on unsymmetric systems [11].

We define a matching on the bipartite graph as a set of edges, no two of which have the same vertex as an endpoint. A maximum matching is a matching with maximum cardinality that will be n if the matrix is structurally nonsingular. We will later (in section 5.1) relate matchings on the bipartite graph to matchings on another undirected graph that is a modified version of the undirected graph described above.

4.2.2. Symmetrization. When MC64 is used with a product metric as in (4.1) to define the weight of a set of edges, it also returns a row and column scaling. We now show how a symmetric scaling can be built from the MC64 scaling. It will be called MC64SYM.

DEFINITION 4.1. Let $B = (b_{ij})$ be the matrix returned by MC64. Then $\exists \sigma \in S_n$ such that $\forall i, |b_{i\sigma(i)}| = 1$ and $\forall (i, j) \neq (i, \sigma(i)), |b_{ij}| \leq 1$.

The above definition says that the magnitude of all the entries in the matrix B is less than or equal to 1 and that we can permute the columns of B so that it has entries of magnitude 1 on the diagonal.

PROPERTY 4.1. Let $\mathcal{M}$ be the maximum matching returned by MC64. Let $D_r = (d_{r_i})$, $D_c = (d_{c_i})$ be the row and column scaling matrices returned by MC64. Let $D = (d_i) = \sqrt{D_r D_c}$. Then DAD is symmetric.

The entry of $|DAD|$ in position (i, j) is $|d_i d_j a_{ij}| = \sqrt{|d_{r_i} d_{c_j} a_{ij}|} \sqrt{|d_{r_j} d_{c_i} a_{ji}|}$. This is less than or equal to one because it is the square root of the product of two entries of $|D_r A D_c|$. Let σ be the column permutation associated with $\mathcal{M}$. Then, for all i , $|d_{r_i} d_{c_{\sigma(i)}} a_{i\sigma(i)}| = 1$. Let $\lambda_i = |d_{c_i} d_{r_{\sigma(i)}} a_{i\sigma(i)}| = |d_{c_i} d_{r_{\sigma(i)}} a_{\sigma(i)i}|$. Then $\lambda_i \leq 1 \forall i$ because it corresponds to an entry of $|D_r A D_c|$. Then

$$\prod_{1 \leq i \leq n} \lambda_i = \left| \prod_{1 \leq i \leq n} d_{r_i} \right| \left| \prod_{1 \leq i \leq n} d_{c_i} \right| \left| \prod_{1 \leq i \leq n} a_{i\sigma(i)} \right| = \prod_{1 \leq i \leq n} |d_{r_i} d_{c_{\sigma(i)}} a_{i\sigma(i)}| = 1,$$

where we exchange the terms in the products to get the product of the scaled terms in the MC64 maximum matching. That is, the product of the numbers, λ_i , each of which is less than or equal to one, is one so that $\lambda_i = 1 \forall i$, and so

$$\forall i, |d_i d_{\sigma(i)} a_{i\sigma(i)}|^2 = |d_{r_i} d_{c_{\sigma(i)}} a_{i\sigma(i)}| |d_{r_{\sigma(i)}} d_{c_i} a_{i\sigma(i)}| = 1 \times \lambda_i = 1. \quad \square$$

4.2.3. Structurally singular matrices. Structurally singular matrices are quite common in optimization. The notion of a maximum weighted matching with the product metric is not well defined for MC64. All the maximum matchings of MC64 will have zero weight. Moreover, it is impossible to find scaling factors that satisfy the MC64 constraints for structurally singular matrices. (It is impossible to find a permutation σ that satisfies the relation of Definition 4.1.)

Suppose that the weight of a matching on a structurally singular matrix is given by the product of the absolute values of the matched entries (which are nonzeros). Then we can define a maximum weighted matching on a structurally singular matrix as a matching of maximum weight among the matchings of maximum size. We did not modify the MC64 code to get this kind of matching, but we used a less complicated algorithm to get an approximation to a maximum weighted matching. We first apply MC64 to A , and we get a maximum matching $\mathcal{M}$. Then we extract a structurally nonsingular symmetric submatrix $\tilde{A}$ from A using Property 4.2. We then apply MC64 this time to $\tilde{A}$.

PROPERTY 4.2. $A = \begin{bmatrix} A_{11} & A_{12} \\ A_{21} & A_{22} \end{bmatrix}$, $\mathcal{M} = \mathcal{I} \otimes \mathcal{J}$, $\tilde{A} = \begin{bmatrix} A_{11} & A_{12} \\ A_{21} & A_{22} \end{bmatrix}$, $\mathcal{I} \times \mathcal{I}$, $\mathcal{J} \times \mathcal{J}$. We define the bipartite graph $G = (R, C, E)$, where $a_{ij} \neq 0$ if and only if $(i^{\mathcal{R}}, j^{\mathcal{C}}) \in E$. The notation $\mathcal{R}$ (resp., $\mathcal{C}$) will be added to a set of indices when we want to explicitly mention that it refers to a set of rows (resp., columns) included in R (resp., in C). A path in this graph from i_1 to i_k is defined by a sequence $(i_1, \dots, i_k)$, where $(i_t^{\mathcal{R}}, i_{t+1}^{\mathcal{C}})_{t=1, \dots, k-1} \in E$. This path is a cycle if $i_k \equiv i_1$. By definition, the length of this path is k . We can similarly define a path using the maximum matching $\mathcal{M} \subset E$, where $(i_t^{\mathcal{R}}, i_{t+1}^{\mathcal{C}})_{t=1, \dots, k-1} \in \mathcal{M}$. A maximum open path of the matching is defined as a path $(i_1, \dots, i_k)$ where there are no v such that $(v^{\mathcal{R}}, i_1^{\mathcal{C}}) \in \mathcal{M}$ or $(i_k^{\mathcal{R}}, v^{\mathcal{C}}) \in \mathcal{M}$. All matching edges are either in cycles or in maximum open paths.

The length of a maximum open path must be odd since, if it were even, the symmetry of the matrix would allow us to extend the matching by using the edges $(i_1^{\mathcal{R}}, i_2^{\mathcal{C}}), (i_2^{\mathcal{R}}, i_1^{\mathcal{C}}), (i_3^{\mathcal{R}}, i_4^{\mathcal{C}}), (i_4^{\mathcal{R}}, i_3^{\mathcal{C}}), \dots, (i_{k-1}^{\mathcal{R}}, i_k^{\mathcal{C}}), (i_k^{\mathcal{R}}, i_{k-1}^{\mathcal{C}})$, thus contradicting the fact that we have a maximum matching. Figure 1(a) shows this extension for $k = 4$.

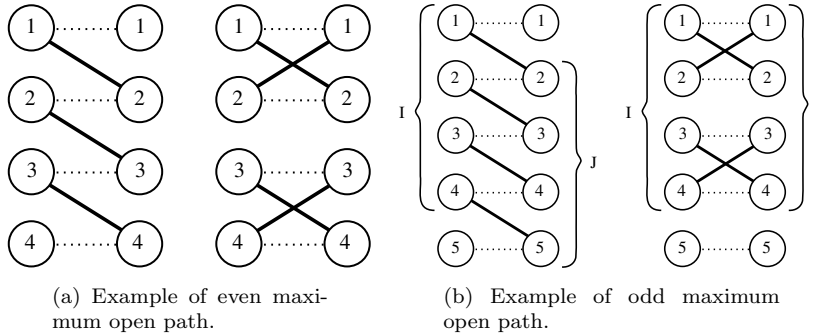


FIG. 1. Maximum open paths. Bold edges are edges of the matching. Dashed lines represent the diagonal of the matrix, which may not be entries.

We now construct a maximum matching on $\mathcal{I}^{\mathcal{R}} \times \mathcal{I}^{\mathcal{C}}$ of the same cardinality as that on $\mathcal{I}^{\mathcal{R}} \times \mathcal{J}^{\mathcal{C}}$. Clearly, any cycle gives corresponding matching edges in $\mathcal{I}^{\mathcal{R}} \times \mathcal{I}^{\mathcal{C}}$. Consider a maximum open path $(i_1, \dots, i_k)$, $i_1, \dots, i_{k-1} \in \mathcal{I}$, $i_k \in \mathcal{J} \setminus \mathcal{I}$. Then we can replace the matching edges corresponding to this path by $(i_1^{\mathcal{R}}, i_2^{\mathcal{C}}), (i_2^{\mathcal{R}}, i_1^{\mathcal{C}}), \dots, (i_{k-2}^{\mathcal{R}}, i_{k-1}^{\mathcal{C}}), (i_{k-1}^{\mathcal{R}}, i_k^{\mathcal{C}})$, all of whose endpoints are in $\mathcal{I}$. This is illustrated by the example in Figure 1(b) for $k = 5$.

TABLE 4.1

Factorization time (columns *Facto*) and scaling time (columns *Scal*). Times in seconds. **AMD** ordering used. Number of steps for **MC77** set to 20. **CPU**: the maximum CPU time exceeded. **MEM**: **MA57** ran out of memory. **PREC**: problem in precision of the solution.

Matrix	No scaling	MC77inf		MC77one		MC64SYM	
	Facto	Facto	Scal	Facto	Scal	Facto	Scal
Augmented systems (<i>AUGSYS</i> set)							
BLOWEYA	CPU	0.08	0.04	0.09	0.04	0.08	0.03
BRATU3D	PREC	PREC	–	77.6	0.06	PREC	–
NCVXQP1	108.	13.6	0.03	19.3	0.01	12.8	0.43
NCVXQP5	MEM	138.	0.19	140.	0.18	139.	2.94
NCVXQP7	MEM	MEM	–	CPU	–	1307.	21.9
bloweybl	CPU	0.06	0.03	0.08	0.04	0.06	0.03
cvxqp3	263.	30.6	0.03	37.0	0.04	37.0	0.94
stokes128	2.03	0.81	0.21	0.79	0.16	0.81	0.29
stokes64	0.18	0.14	0.04	0.14	0.01	0.13	0.06
tuma2	0.08	0.08	0.01	0.06	0.01	0.04	0.01
TOTAL solved	18/23	21/23		22/23		22/23	
General symmetric indefinite (<i>GENSYM</i> set)							
BOYD1	CPU	CPU	–	39.9	0.44	33.9	14.8
c-68	24.5	23.1	0.29	28.5	0.21	22.2	0.13
c-71	76.4	76.2	0.48	108.	0.29	76.4	0.28
vibrobox	1.66	1.26	0.04	1.29	0.06	1.28	0.03
TOTAL solved	15/16	15/16		16/16		16/16	

This then gives us a maximum matching on $\mathcal{I} \times \mathcal{I}$ of cardinality $|\mathcal{I}|$. $\square$

We relax the **MC64** constraints on the scaling. Let D_r and D_c be the scaling factors returned by **MC64** on $\tilde{A}$. We build D so that if i corresponds to an index of $\tilde{A}$, $d_i = \sqrt{d_{r_i} d_{c_i}}$; otherwise

$$d_i = \frac{1}{\max_{k \in \text{index}(\tilde{A})} |a_{ik} d_k|} \text{ with the convention } \frac{1}{0} = 1.$$

Note that the entries of $|DAD|$ are less than 1, and entries in the matching are 1. Moreover, the maximum entry in absolute value in each nonempty row/column is 1.

4.3. Influence of scaling. It is important to note that scaling does not change the pivot order returned by the analysis but changes the selection of the numerically stable 1×1 and 2×2 pivots during the factorization. Scaling can have a profound effect on the subsequent factorization. A good scaling can avoid many numerical difficulties, whereas a bad scaling can actually cause numerical problems, can produce a lot of delayed pivots when not necessary, can increase the memory requirements, and can consequently severely slow down the factorization.

On the *AUGSYS* and *GENSYM* sets, the approach without any scaling (No scaling) fails on 6 matrices, the approaches using **MC30**, **MC77inf**, **MC77one**, and **MC64SYM** fail on 3, 3, 1, and 1 matrices, respectively (see Table 4.1). The use of scaling thus improves the robustness of the **MA57** factorization in terms of the number of successful computations. Table 4.1 shows that, most of the time, failures are due to numerical pivoting, which increases the CPU time and memory requirement (CPU and MEM failures). It seems difficult to get a good solution for the **BRATU3D** problem, except with the **MC77one** scaling. Figure 2 compares the influence of the different scalings on the CPU factorization time of **MA57** on the *AUGSYS* and *GENSYM* sets.

On the *GENSYM* set (Figure 2(b)), the **MC30** scaling degrades the **MA57** performance, whereas the other scalings have a positive effect on these general indefinite

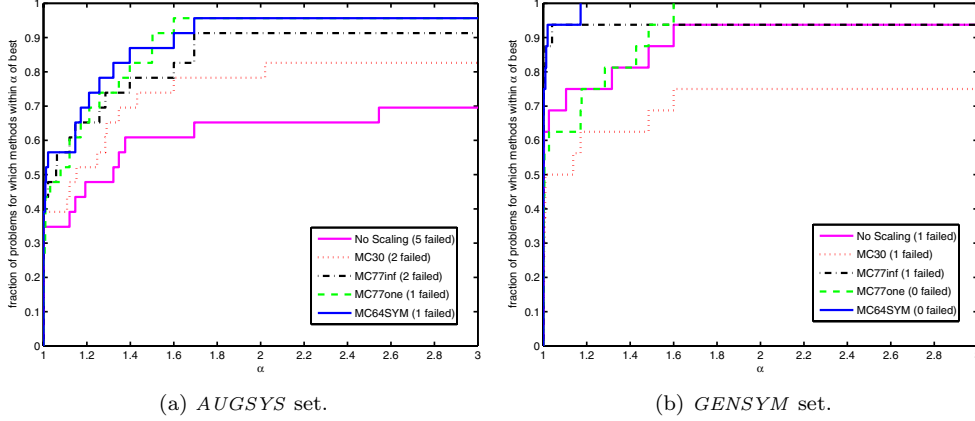


FIG. 2. Profile showing influence of the scaling on the CPU factorization time (AMD ordering).

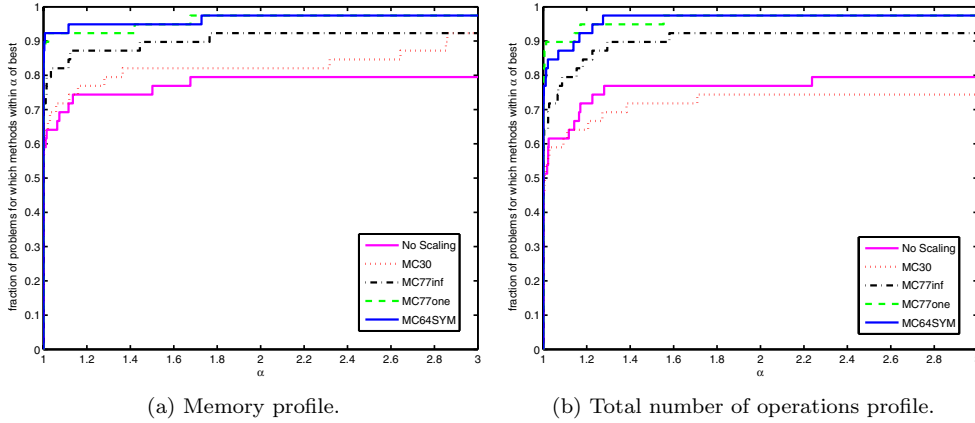


FIG. 3. Profile showing influence of the scaling on the memory and the number of operations for the factorization (AUGSYS and GENSYM sets together).

matrices. The negative effect of MC30 on indefinite problems has also been observed by [21]. Contrary to the other scalings (MC64, MC77), MC30 does not take into account structural information when doing the scaling and uniformly tries to scale all entries to be as close to one as possible. We suspect that the structural decision of the analysis is less compatible with the MC30 scaling. This difference can provide a partial answer about the degradation of performance when using MC30 scaling. We recommend that the MC30 scaling is not used on symmetric indefinite problems. Furthermore, the MC64SYM scaling seems to be the most robust scaling on this set. (The MC77inf approach is also good but only for 95% of the problems, whereas the MC64SYM scaling is within a factor of 1.17 of the best on 100% of the problems.) MC64SYM and MC77one have a similar behavior on the augmented systems and are slightly better than MC77inf (see Figure 2(a)).

The CPU factorization times in Figure 2, the number of operations in Figure 3(b), the memory used in Figure 3(a), and the number of delayed pivots have similar characteristics. Figure 4(a) shows that the number of delayed pivots is sometimes very large when using the MC30 scaling. The average behavior of MA57 using the MC30

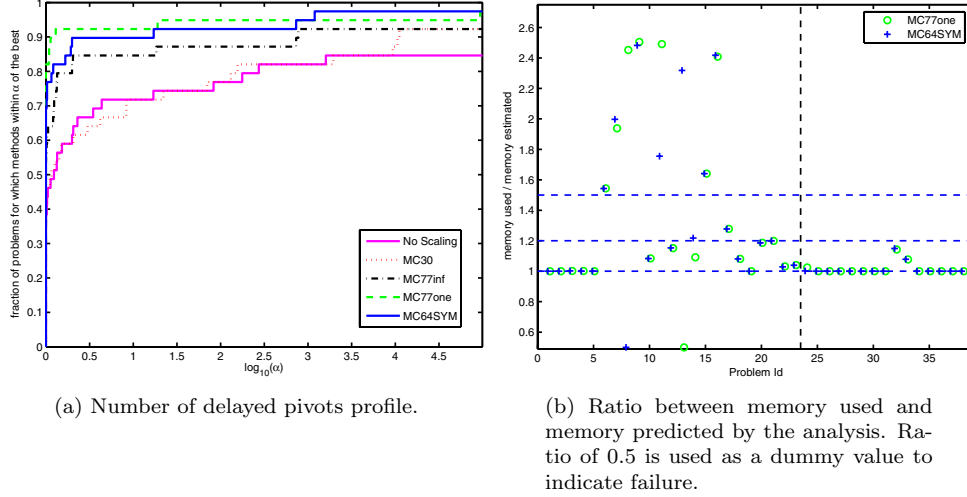


FIG. 4. Profile showing influence of scaling on number of delayed pivots and memory (AMD ordering).

scaling is similar to applying no scaling. We show in Figure 4(b) the ratio between the memory used by the factorization and the memory predicted by the analysis. This indicates that the memory predictions of the analysis are sometimes not respected. It also indicates that most of the numerical problems appear on *AUGSYS* set (to the left of the vertical line). Indeed, there are no problems in the *GENSYM* set over the 20% limit. If we allocate only 50% more memory than recommended by the analysis instead of 4 GBytes, we would have 7 additional failures with the *MC64SYM* and *MC77one* scalings (points above the highest horizontal dotted line in Figure 4(a)).

Table 4.1 summarizes the *MC64SYM*, *MC77inf*, or *MC77one* scaling impact on the factorization of symmetric indefinite problems. Using the *MC64SYM* scaling, we get infinite speedup on 5 problems, a speedup greater than 5 on 7 matrices, a speedup of between 2 and 5 on two matrices. This approach is faster than no scaling on 30 matrices, and on 9 others it never exceeds the factorization time for no scaling by a factor of more than 1.25. We see also that on the *GENSYM* set, the *MC77one* scaling has problems, whereas *MC64SYM* does not (see the c-68 and c-71 matrices). As mentioned before, on the *GENSYM* set, the same order of speedup is obtained with the *MC77inf* scaling even if it is less robust in terms of the number of failures.

It may be interesting to use the *MC77inf* scaling on the *GENSYM* set and the *MC77one* scaling on the *AUGSYS* set because they have a comparable influence on the factorization and are less costly to compute (see Table 4.1). Nevertheless, *MC64SYM* remains competitive because it is more robust than *MC77*: the maximum number of iterations of *MC77* has been set to 20 in our experiments, and the norms of the rows and columns may still be far from 1 (for example with *MC77one* on the *GENSYM* set). A safe approach with *MC77* could be to check the convergence every 20 steps and continue iterating until we are close to the limit. It would involve an additional cost that is not included in Table 4.1. On the contrary, when *MC64SYM* has finished, it guarantees that our scaled matrix satisfies Property 4.1. Second, because the scaling time has to be considered relative to the factorization time, Table 4.1 shows that the *MC64SYM* cost is usually compensated for by the gain in factorization time.

In the rest of this paper, we try to improve the CPU factorization time and to obtain a better behavior with respect to other criteria (for example, the quality of the memory prediction). A first way of improving the **MA57** factorization comes from using other orderings (**METIS** and **AMF**) to decrease the fill-in, the memory, and the number of operations. A second way is to influence a priori the ordering with numerical and structural information about the 2×2 pivots. The main goal of this approach is to preselect the pivots that will be used during the factorization and thus decrease the number of delayed pivots. As **MC64** is used in the approach that tries to answer the second point (i.e., the use of the **MC64SYM** scaling does not involve additional computation) and as the **MC64SYM** scaling seems to be a robust approach on both sets, we will systematically use this scaling in the remainder of this paper. From now on, A will refer to the matrix symmetrically scaled by **MC64SYM**. (Its entries are in $[-1, 1]$.)

5. Ordering and weighted matching. In this section, we present ordering approaches that aim at decreasing fill-in in the factors and at preselecting good pivots during the analysis. Thus we are interested in an algorithm that gives us a good approximation about what will happen during the factorization and that does not generate too much fill-in in the factors. We will first discuss our choices in section 5.1. Then we present our different approaches.

The matrix

$$A_{33}(x) = \begin{pmatrix} 0 & 1 & 1 & x & 0 & 0 \\ 1 & 0 & 1 & 0 & 0 & 0 \\ 1 & 1 & 0 & 0 & 0 & 0 \\ x & 0 & 0 & 0 & 1 & 1 \\ 0 & 0 & 0 & 1 & 0 & 1 \\ 0 & 0 & 0 & 1 & 1 & 0 \end{pmatrix}$$

that has two 3×3 cycles with edges of weight 1 will be used in this section to illustrate our discussion.

5.1. Symmetric maximum weighted matching. In the unsymmetric case, a maximum weighted matching of the bipartite graph associated with the matrix is computed to put large entries onto the diagonal of the permuted matrix. A natural way of adapting this to the symmetric case is to search for a *symmetric maximum weighted matching*, $\mathcal{M}_s^{opt}$ (a maximum weighted matching such that if (i, j) is in $\mathcal{M}_s^{opt}$, then (j, i) is in $\mathcal{M}_s^{opt}$). Property 5.1 presents a method to get a symmetric maximum weighted matching. We will then explain why we decided not to use this method and to build only an approximation of it using the **MC64** maximum weighted matching. The fact that using **MC64** may not return the optimum can be seen in the example $A_{33}(10^{-1})$. **MC64** could return the matching $\mathcal{M} = \{(1, 2), (2, 3), (3, 1), (4, 5), (5, 6), (6, 4)\}$, where we use the convention that, on a bipartite graph, the first index of an edge corresponds to a row vertex and the second index to a column vertex. However, the symmetric maximum weighted matching is $\mathcal{M}_s^{opt} = \{(1, 4), (4, 1), (2, 3), (3, 2), (6, 5), (5, 6)\}$. (If $x \neq 0$, it is the unique symmetric matching of size 6, and so it is the symmetric maximum weighted matching.) It is impossible to build $\mathcal{M}_s^{opt}$ using only information in $\mathcal{M}$ because $(1, 4)$ does not appear in it.

PROPERTY 5.1. Let $\mathcal{M}$ be a maximum weighted matching of the bipartite graph associated with the matrix A . Let $\mathcal{M}_s^{opt}$ be a symmetric maximum weighted matching of the same graph. Then, if $(i, j) \in \mathcal{M}_s^{opt}$, then $(j, i) \in \mathcal{M}_s^{opt}$.

A constructive proof of Property 5.1 can be found in [29], which gives us a way of computing a symmetric maximum weighted matching and says that the complexity of this problem is of the same order as the complexity of the computation of the maximum weighted matching on a graph with the same number of vertices and with the same number of edges as in the bipartite graph of A . Solving the problem of the maximum matching on a nonbipartite graph is much more complicated than on a bipartite one. A solution to this difficult problem has been found by [17, 18], and its complexity was bounded by $\mathcal{O}(n^4)$. This complexity bound was later decreased to $\mathcal{O}(nnz(A)n + n^2 \log n)$ by [19].

We prefer to find an approximation to the symmetric maximum matching using information returned by **MC64** for the following reasons:

- We want a preprocessing with reasonable complexity with respect to the rest of the analysis and the factorization.
- We do not know an efficient available code that uses Property 5.1.
- In the unsymmetric case, **MC64** has a complexity in the worst case of $\mathcal{O}(\tau n \log n)$, where τ is the number of nonzeros in A and in practice has a better average behavior.
- Using **MC64** enables us to compute both the **MC64SYM** scaling and an approximation of a symmetric maximum weighted matching in linear time. We get two complementary preprocessings for the cost of one.

That is why we decided to have a weaker formulation of the problem—we want to find a symmetric weighted matching that is not too far from the optimum and to use the maximum matching technique on weighted bipartite graphs. On 33 matrices out of the 39 in our test set, a solution to the weak formulation leads to a solution of the symmetric maximum weighted matching [29]. In [29] we also give theoretical bounds as to how far this formulation can be from the symmetric maximum matching. For example, if a structurally nonsingular matrix satisfies the **MC64** constraints, then a solution to the weak formulation $\mathcal{M}_s$ satisfies $\omega(\mathcal{M}_s) \geq \omega(\mathcal{M}_s^{opt})\omega(\mathcal{M}_{1 \times 1})$, where $\mathcal{M}_s^{opt}$ is a symmetric maximum weighted matching and $\mathcal{M}_{1 \times 1}$ are the diagonal entries that belong to $\mathcal{M}_s$.

5.2. Selection of 2×2 pivots and symmetric weighted matchings. **MC64** is first used to compute a maximum weighted matching $\mathcal{M}$. This will normally have many entries that are not on the diagonal of A . Any diagonal entries that are in the matching are immediately considered as potential 1×1 pivots and are held in a set $\mathcal{M}_{1 \times 1}$. We then build a set $\mathcal{M}_{2 \times 2}$ of potential 2×2 pivots. We use the 2×2 pivot selection strategy suggested by [9] but with a structural metric instead of a numerical criterion. The basis for this strategy is to express the computed permutation, σ , in terms of its component cycles. For example, Figures 5 and 6 show cycles of length 4 and 5, respectively. Because of the scaling, all the entries in the cycles of σ are 1 in absolute value, so we choose a structural criterion to select the potential 2×2 pivots. A cycle of length 1 corresponds to an entry a_{ii} in the matching. A cycle of length 2 corresponds two nodes i and j , where a_{ij} and a_{ji} are both in the matching. We can extract k 2×2 pivots from even cycles of length $2k$ or from odd cycles of length $2k + 1$. The principle of this extraction is to select k entries a_{ij} and their symmetric counterpart and to discard the other matched entries. For even cycles there are only two alternatives for this extraction. For example, in Figure 5, we can select the entries $a_{1,2}$ and $a_{3,4}$ and their symmetric counterparts $a_{2,1}$ and $a_{4,3}$ and discard the entries $a_{2,3}$ and $a_{4,1}$ (left plot). For odd cycles of length $2k + 1$ there are $2k + 1$ possible combinations of 2×2 pivots (see Figure 6).

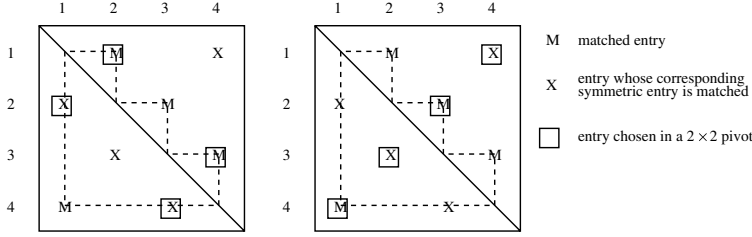


FIG. 5. Example of selection of 2×2 pivots in an even cycle. Matched entries are in positions (1, 2), (2, 3), (3, 4), and (4, 1). We show the two possibilities for selecting two 2×2 pivots.

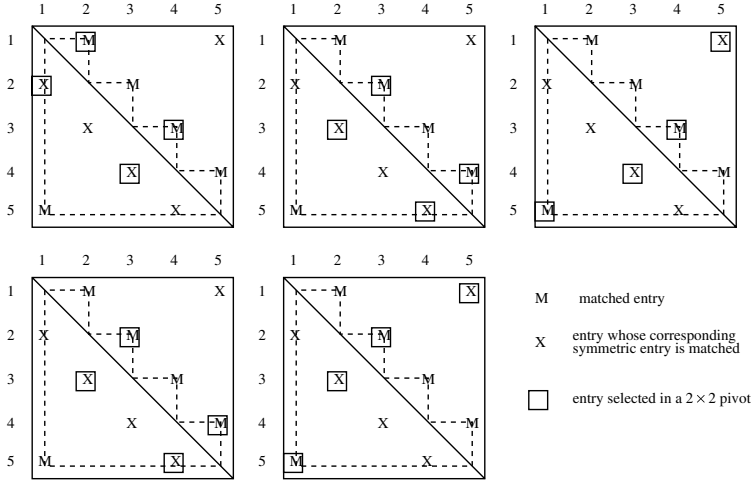


FIG. 6. Example of selection of 2×2 pivots in an odd cycle. Matched entries are in positions (1, 2), (2, 3), (3, 4), (4, 5), and (5, 1). We show the five possibilities for selecting two 2×2 pivots.

To make a choice from the alternative sets of pivots in Figures 5 and 6, we define a metric that is denoted by $metric(i, j)$ for a potential 2×2 pivot in rows i and j . In our approach, we use $metric(i, j) = |R_i \cap R_j| / |R_i \cup R_j|$, where R_i and R_j are the structure of row i and j , respectively. The motivation for this metric is to associate rows in 2×2 pivots that have as similar a structure as possible. We will see in section 5.3 that before the ordering phase, for each 2×2 pivot (i, j) , we assume that its nonzero structure is the merge of the structure of row i and j . So maximizing this metric will tend to decrease the fill-in due to the use of 2×2 pivots. For each cycle in σ , we then select 2×2 pivots such that the sequence of 2×2 pivots (i_k, j_k) maximizes $\prod metric(i_k, j_k)$. This selection can be done in $\mathcal{O}(sn)$ time, where $\mathcal{O}(s)$ is an upper bound on the time to compute $metric(i, j)$ (see [29]). In our selection, the time to compute $metric(i, j)$ is $\mathcal{O}(|R_i| + |R_j|)$ and the total time to select the 2×2 pivots is $\mathcal{O}(\tau)$, where τ is the number of nonzeros in A .

At the end of this $\mathcal{M}_{2 \times 2}$ computation, we add to $\mathcal{M}_{1 \times 1}$ the nonzero diagonal entries that were not selected during the pass over odd cycles. We define $\mathcal{M}_s$ as $\mathcal{M}_{1 \times 1} \cup \mathcal{M}_{2 \times 2}$ corresponding to a symmetric matching obtained from $\mathcal{M}$. We note that $\mathcal{M}_{1 \times 1} \cap \mathcal{M}_{2 \times 2} = \emptyset$. Let $\mathcal{M}_s^c$ be the set $\{(i, i) \text{ such that } i \text{ does not belong to any pivots in } \mathcal{M}_s\}$, that is, the complementary set to $\mathcal{M}_s$. It may be nonempty because of odd cycles with zero diagonals. In practice, we note that this set is small. ($\mathcal{M}_s^c = \emptyset$

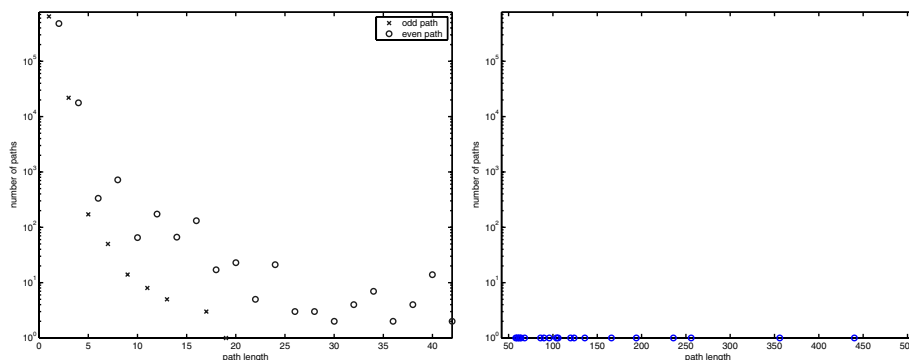


FIG. 7. Length of MC64 paths.

on all the matrices of the *AUGSYS* and *GENSYM* sets except the structurally singular ones.) Thus we will not investigate approaches to increase the size of $\mathcal{M}_s$ further.

In practice, most of the cycles from the MC64 permutation are of length 1 or 2. On our test matrices, 55% of the cycles are of length 1, 41% are of length 2, and less than 4% have a length greater than 2. We illustrate this in Figure 7, where we show the total number of paths of varying lengths for runs over all of our test matrices. That is why we will not try different structural criteria to select the 2×2 pivots.

5.3. Coupling detected pivots and orderings.

5.3.1. Ordering on the compressed graph. Let $\mathcal{M}$ be a maximum matching on A from which we have obtained a set of 1×1 and 2×2 candidate pivots, $\mathcal{M}_s$. The undirected graph $\mathcal{G}$ associated with the matrix A has a set of vertices V_A , corresponding to the rows (and columns) of A and a set of unordered edges E_A , where the unordered pair $(i, j) \in E_A$ if and only if $a_{ij} \neq 0$.

We define R , the *compressed graph*, as the square matrix of order the number of pivots in $\mathcal{M}_s$ whose associated undirected graph will be referred to as the *compressed graph*. (It is a generalization of the compressed graph of [2], where only indistinguishable vertices are compressed.) Each of its vertices $\mathcal{I}_i$ is weighted and either associated with a row/column of A corresponding to a candidate 1×1 pivot (weight of 1) or a pair of rows/columns (i_k, i_l) corresponding to a candidate 2×2 pivot (weight of 2). The edge set E_R consists of the unordered pairs $(\mathcal{I}_i, \mathcal{I}_j)$, where there exists an edge in E_A between one of the constituent vertices of $\mathcal{I}_i$ and one of the constituent vertices of $\mathcal{I}_j$. In other words, the candidate 2×2 pivots of $\mathcal{M}_s$ are compressed into one vertex, and the union of their adjacency defines the adjacency of the new vertex. The weight of the vertices will be used to initialize the size of the supervariables and to compute the appropriate metric. For example, the external degree of a supervariable $\mathbf{i}$ will be initialized to $\text{ext\_deg}(\mathbf{i}) = \sum_{\mathbf{j} \in \text{Adj}(\mathbf{i})} |\mathbf{j}|$, where $\text{Adj}(\mathbf{i})$ is the set of vertices adjacent to $\mathbf{i}$ in the compressed graph and $|\mathbf{j}|$ is the weight of the vertex $\mathbf{j}$. Obviously, when two supervariables $\mathbf{i}$ and $\mathbf{j}$ are merged, either in the phase of supervariable detection for a greedy ordering or in the coarsening phase for a partitioning, they form a new supervariable of size $|\mathbf{i}| + |\mathbf{j}|$. Note that the rows/columns of A that are not represented in $\mathcal{M}_s$ will not be represented in the reduced matrix.

Let P_{red} be a symmetric permutation on R . Then it is easy to extend P_{red} to a permutation P_{aug} on A by expanding each component corresponding to a 2×2 composite node to the two rows/columns of A associated with that composite node

and by putting the rows/columns of A that were not represented in R ($\mathcal{M}_s^c$ entries) at the end of this permutation. This ensures that zero pivots in $\mathcal{M}_s^c$ will be filled by the previous eliminations if we assume that no numerical cancellation occurs and the matrix is structurally nonsingular. Clearly, this expansion can be done in a single pass through P_{red} . Note that P_{aug} is not unique: first, when a node corresponding to a 2×2 pivot (i, j) is expanded, we have the choice of taking i or j first in P_{aug} ; second, when we visit the entries in $\mathcal{M}_s^c$, there is no a priori order for placing them in P_{aug} . This approach on the compressed graph generates an ordering that is expected to have preselected good numerical pivots. Algorithm 1 summarizes the main steps of this preprocessing.

Algorithm 1. MAIN STEPS OF THE ORDERING ON THE COMPRESSED GRAPH.

- 1 Apply a symmetric MC64 scaling and get $\mathcal{M}_s$.
 - 2 Compute R , a reduced matrix of A relative to $\mathcal{M}_s$.
 - 3 Set P_{red} , the symmetric permutation returned by an ordering on R .
 - 4 Compute P_{aug} , an extension of P_{red} relative to $\mathcal{M}_s$.
 - 5 Put the components of $\mathcal{M}_s^c$ in the last positions.
-

To illustrate the compression and the expansion, let us take the following matrices:

$$A = \begin{pmatrix} 2 & -1 & 1 & 0 & 0 \\ -1 & 2 & 0 & 0 & 0 \\ 1 & 0 & 0 & 2 & 1 \\ 0 & 0 & 2 & 0 & 1 \\ 0 & 0 & 1 & 1 & 0 \end{pmatrix} \text{ and } R = \begin{pmatrix} 1 & X & X \\ X & 1 & 0 \\ X & 0 & 2 \end{pmatrix}.$$

In A , we detect a 2×2 pivot in rows/columns 3 and 4 and two 1×1 pivots on the diagonal. We have $\mathcal{M}_s = \{(1, 1), (2, 2), (3, 4)/(4, 3)\}$ and $\mathcal{M}_s^c = \{(5, 5)\}$. R is a reduced matrix of A relative to $\mathcal{M}_s$, where row/column 3 corresponds to the 2×2 pivot. If $P_{red} = [3, 2, 1]$, then $P_{aug} = [3, 4, 2, 1, 5]$ or $[4, 3, 2, 1, 5]$ is an expansion of P_{red} .

5.3.2. Constrained ordering. The main principle of the orderings presented in this section and section 5.3.3 is to relax the above preselection of the 2×2 pivots by splitting some of them into two 1×1 pivots before the compression. Moreover, when there is a danger of making a bad numerical decision, we will add some dependency about the precedence between some of the 1×1 pivots coming from split 2×2 pivots. We will propose an algorithm that can be implemented in the context of orderings based on local heuristics such as AMD or AMF.

For each 2×2 pivot, $P_{ij} = \begin{pmatrix} a_{ii} & a_{ij} \\ a_{ij} & a_{jj} \end{pmatrix}$, with (i, j) ordered such that $|a_{ii}| \geq |a_{jj}|$ and with $|a_{ij}| = 1$, i is said to be the *strong* pivot and j the *weak* pivot. Let $\theta \in [0, 1]$ be a real constant. P_{ij} is said to be

- (a) a *locked* pivot according to θ if and only if $|a_{ii}| \leq \theta$ and $|a_{jj}| \leq \theta$,
- (b) a *constrained* pivot if and only if $|a_{ii}| > \theta$ and $|a_{jj}| \leq \theta$,
- (c) a *splittable* pivot if and only if $|a_{ii}| > \theta$ and $|a_{jj}| > \theta$.

The set of MC64SYM detected 2×2 pivots is separated into three sets, LP_θ , CP_θ , and SP_θ , the set of locked pivots, constrained pivots, and splittable pivots, respectively. Note that when $\theta = 0$, locked pivots correspond to $\begin{pmatrix} 0 & 1 \\ 1 & 0 \end{pmatrix}$ pivots (two zeros on the diagonal), constrained pivots to $\begin{pmatrix} 0 & 1 \\ 1 & 1 \end{pmatrix}$ pivots (one zero on the diagonal), and splittable pivots to full 2×2 pivots. When $\theta = 1$, none of the 2×2 pivots is split, and the ordering will behave as the ordering on the compressed graph. The more θ decreases,

the more pivots are split, and the risks of making a bad decision increase. In the rest of our discussion, we will omit the θ from our notation.

During the ordering, we manipulate two kinds of (super)variables, *free* and *constrained*. At the beginning of the ordering, a supervariable i is said to be a free supervariable if and only if one of the following conditions hold:

- (1) i belongs to a splittable pivot;
- (2) i is the leading variable of a constrained pivot;
- (3) i belongs to $\mathcal{M}_{1 \times 1}$;
- (4) i is a locked pivot.

Otherwise, a supervariable i is said to be a constrained if it appears as the trailing variable of a constrained pivot.

During the pivot selection, there are two main rules:

- (R1) a free supervariable can be eliminated whenever we want (it does not depend on the elimination of another one);
- (R2) a constrained supervariable can be eliminated if and only if a free pivot with which it is associated has already been eliminated.

The second rule is equivalent to marking constrained supervariables as free as soon as (R2) is satisfied. Thus, during the ordering, constrained supervariables can become free. If i is the leading variable of a constrained 2×2 pivot, we will say that i releases j , where j is the variable associated with i in this pivot. (It is a sufficient condition to make the supervariable j free.) The free supervariables that correspond to entries in $\mathcal{M}_{1 \times 1}$ or to locked pivots can be eliminated, but they do not release any constrained supervariables.

At each step of the ordering, FV and CV will be used to denote, respectively, the set of supervariables that can be eliminated and the set of pivots which cannot be eliminated. For each free supervariable i belonging to a constrained pivot, we define $assoc(i)$, its associated constrained supervariable. For other supervariables $assoc(i) = \emptyset$.

Algorithm 2. CONSTRAINED ORDERING SCHEME.

```

Determine free and constrained supervariables according to  $\mathcal{M}_s$ .
 $CV \leftarrow \{ \text{constrained supervariables} \}$  and  $FV \leftarrow \{ \text{free supervariables} \}$ .
while there are uneliminated supervariables do
  Select pivot  $i \in FV$  with priority based on a metric  $m$ :  $i \leftarrow \arg \min_{p \in FV} m(p)$ 
  Do symbolic elimination of  $i$ 
   $FV \leftarrow (FV \cup \{assoc(i)\}) \setminus \{i\}$ 
   $CV \leftarrow CV \setminus \{assoc(i)\}$ 
end while

```

Algorithm 2 describes our constrained ordering. At each step of the symbolic elimination, we select the best pivot in the set FV . Moreover, when the leading supervariable of a constrained pivot is selected, its associated constrained supervariable is released. (It is inserted in the FV set and removed from the CV set.) Intuitively, if the leading part of a constrained pivot is eliminated, then it is possible that the modified value of the trailing supervariable becomes large because our scaling ensures that $|a_{pq}| = 1$ and $|a_{pp}| \leq 1$, and thus that the corresponding entry becomes numerically acceptable. This constrained ordering has been implemented with the **AMD** and **AMF** orderings. As in the ordering on the compressed graph, we put the variables in $\mathcal{M}_s^c$ at the end of the permutation.

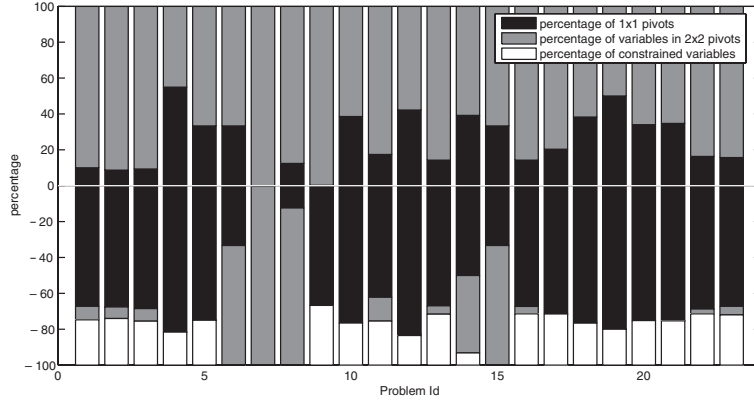


FIG. 8. Percentage of each type of variable with an ordering based on the compressed graph (positive part of the y -axis) and after the preprocessing for a constrained ordering (negative part of the y -axis).

Figure 8 shows the percentage of each type of variable before performing the ordering on the *AUGSYS* set. Our constrained approach splits most of the detected 2×2 pivots (the grey area decreases) into either two 1×1 free variables or in one free variable and one constrained variable (white area), except for five matrices (BLOWEYA, BRAINPC2, BRATU3D, SIT100, and bloweybl), where almost no splitting is done.

5.3.3. Relaxation of constrained ordering. This approach uses the same terminology as the previous one. Here we relax the selection of the pivots; that is, we enlarge the set of free pivots. We fix a dropping threshold $drop \in [0, 1]$ and build a matrix $C = (c_{ij})$ that is called the *constrained dependency matrix*. This matrix together with $\mathcal{M}_s$ will be used to define the constrained and free supervariables and the relations between them (who releases whom). First, the row and column of C that correspond to indices that appear in locked pivots are set to 0. Then the rest of the entries c_{ij} of the constrained matrix are set to 1 if $|a_{ij}| \geq drop$; otherwise they are set to 0. C describes the dependency among the free and constrained supervariables: i can release j if and only if $c_{ij} \neq 0$. The above construction prevents a locked pivot from releasing a constrained supervariable.

Let C_i be the structure of the i th row of C . Each nonzero off-diagonal entry of C defines a 2×2 submatrix. A supervariable i is free if and only if one of the following conditions hold:

- (1) i appears in the indices of a potential splittable pivot in C ;
- (2) i is the leading variable of a potential constrained pivot in C ;
- (3) i corresponds to an entry in $\mathcal{M}_{1 \times 1}$;
- (4) i is a locked pivot.

Here again the supervariables of $\mathcal{M}_{1 \times 1}$ and the locked pivots are systematically included in the set of free variables. During the ordering, a constrained pivot j can be selected if and only if it is reachable from eliminated supervariables in C . In other words, it is possible that it has been updated by large enough entries.

Algorithm 3 describes this approach. At each step, the entries in CV that satisfy the above condition are added to FV . We have implemented both the constrained and the relaxed constrained approaches with an **AMD/AMF** ordering, where small changes are made to allow supervariable detection and locked pivot supervariables are initialized

Algorithm 3. RELAXED CONSTRAINED ORDERING SCHEME.

```

Define the free supervariables and the constrained supervariables according to the constraint matrix
C and  $\mathcal{M}_s$ .
 $CV \leftarrow \{ \text{constrained supervariables} \}$  and  $FV \leftarrow \{ \text{free supervariables} \}$ .
while there are uneliminated supervariables do
  Select pivot  $i \in FV$  with priority based on a metric  $m$ :  $i \leftarrow \arg \min_{p \in FV} m(p)$ 
  Do symbolic elimination of  $i$ 
  Remove from  $C$  the row/column structures that correspond to  $i$ 
   $FV \leftarrow (FV \cup C_{i,:}) \setminus \{i\}$ 
   $CV \leftarrow CV \setminus C_{i,:}$ 
end while

```

with a weight of 2. The total overhead of the manipulation of these sets is $\mathcal{O}(\text{nnz}(C))$, which is negligible with respect to the total complexity of the ordering, shown in [22] to be $\mathcal{O}(n \times \text{nnz}(A))$.

We have presented two alternatives to the ordering based on the compressed graph: a constrained ordering and a relaxed constrained ordering. We expect to have less fill-in using the (relaxed) constrained ordering than using the ordering on the compressed graph. However, we expect that the prediction of the ordering on the compressed graph will be more exact than the prediction of the (relaxed) constrained ordering where there is a greater risk of making a bad numerical decision. We will see in section 6 that these intuitions are verified and that the constrained and the relaxed constrained ordering have a similar behavior.

6. Experimental results. We now consider the effect of our pivoting strategies on MA57. We have four approaches from the above strategies. All four strategies start with an MC64SYM scaling.

- **MA57_plain** refers to the approach using the MC64SYM scaling coupled with an ordering which can be AMD, AMF, or METIS. Note that this approach does not include any cycle-detection process.
- **MA57_compressed** refers to the approach using the MC64SYM scaling, the preselection of the 2×2 pivots and the ordering (AMD, AMF, or METIS) based on the compressed graph (section 5.3.1).
- **MA57_constrained** refers to the approach using the MC64SYM scaling, the preselection of the 2×2 pivots and the constrained ordering scheme (section 5.3.2). This strategy is compatible with AMD or AMF.
- **MA57_relaxed** refers to the approach using the MC64SYM scaling, the preselection of the 2×2 pivots and the relaxed constrained ordering scheme (section 5.3.3). This strategy is compatible with AMD or AMF.

METIS is called using the routine METIS_NodeWND in the MA57_compressed approach in order to take into account the size of the supervariables in the metrics while trying to find a balanced partition. We use the routine METIS_NodeND in the MA57_plain approach since in this case all the supervariables have a size of 1. In MA57_constrained and MA57_relaxed, a threshold of 10^{-2} is used to determine if the diagonal entries correspond to free variables. We tested different values and remark that increasing it from 10^{-2} degrades the fill-in returned by the ordering (there are too many constraints and not enough free variables) and that decreasing it too much from 10^{-2} degrades the memory prediction (on some matrices it did not guarantee good numerical pivots) and did not significantly decrease the fill-in and the number of operations. In MA57_relaxed, a threshold of 0.9 is used to determine the constraint matrix. Here also, we tested different values with the same conclusions as above.

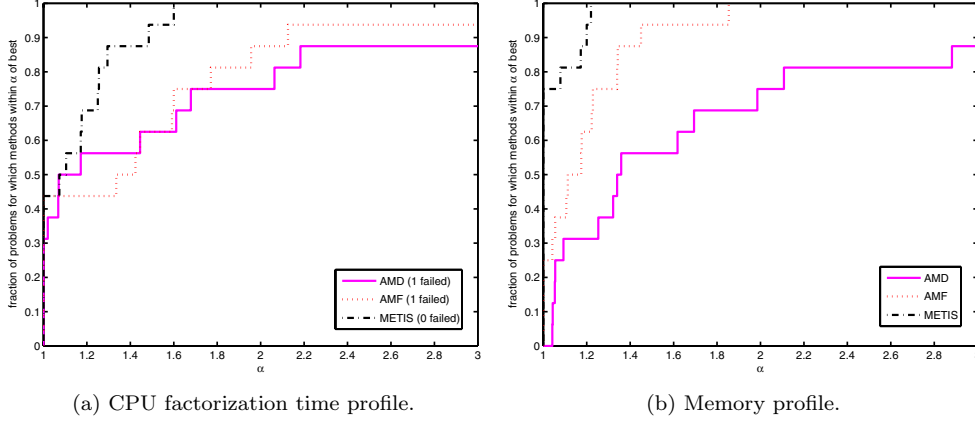


FIG. 9. AMD, AMF, METIS comparison (MA57_plain on the GENSYM set).

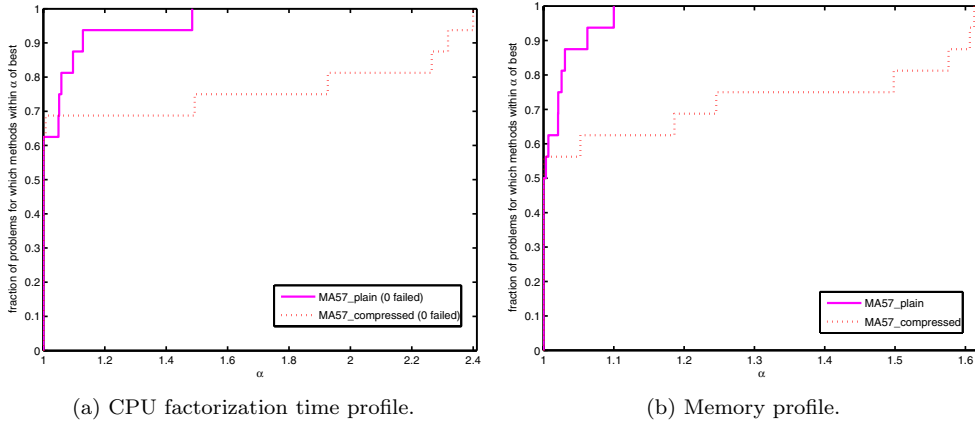


FIG. 10. CPU factorization time and memory (METIS, GENSYM set).

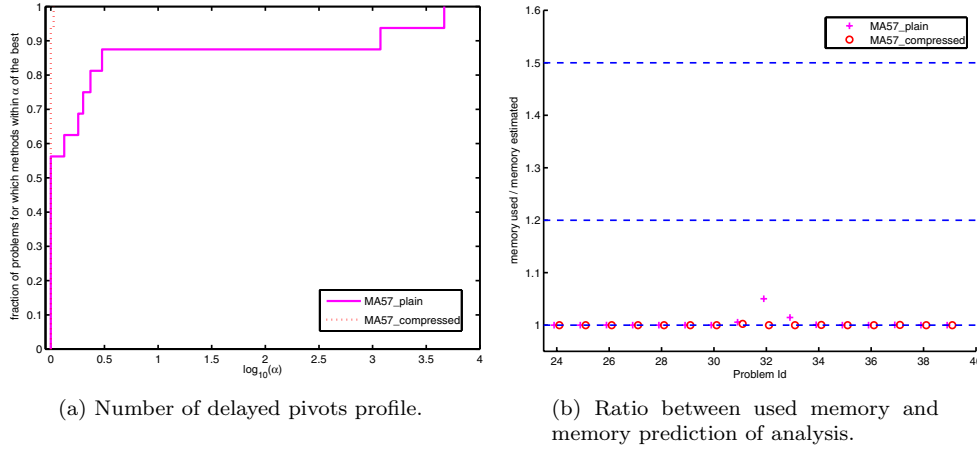
6.1. General symmetric indefinite matrices (GENSYM set). We first determine the best approach on the *GENSYM* set. We will give details about the results for only the METIS based ordering for three reasons.

First, METIS is clearly better than AMD and AMF in terms of CPU factorization time (see Figure 9(a)) and memory (see Figure 9(b)).

Second, in other experiments reported in [29], the relative behavior between AMD+MA57_plain and AMD+MA57_compressed and between AMF+MA57_plain and AMF+MA57_compressed are similar to the relative behavior between METIS+MA57_plain and METIS+MA57_compressed.

Finally, on this set, we noticed that in early experiments [29] the constrained ordering of section 5.3.2 and the relaxed constrained ordering of section 5.3.3 do not significantly improve the CPU factorization time and do not decrease the size of the factors compared with the approach on the compressed graph (section 5.3.1). On the contrary, we will see in the next section that they clearly improve the factorization on the *AUGSYS* set.

Figure 10(a) shows that MA57_plain is faster than MA57_compressed on the *GEN-*

FIG. 11. *Quality of analysis prediction (METIS, GENSYM set).*

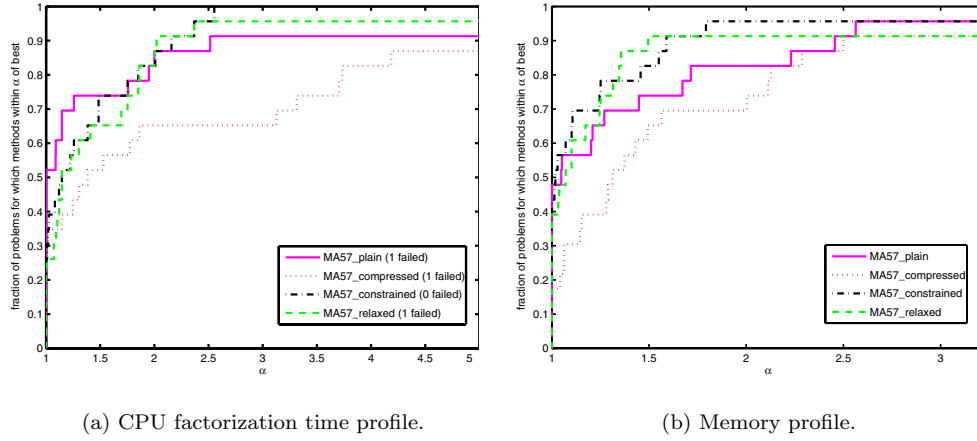
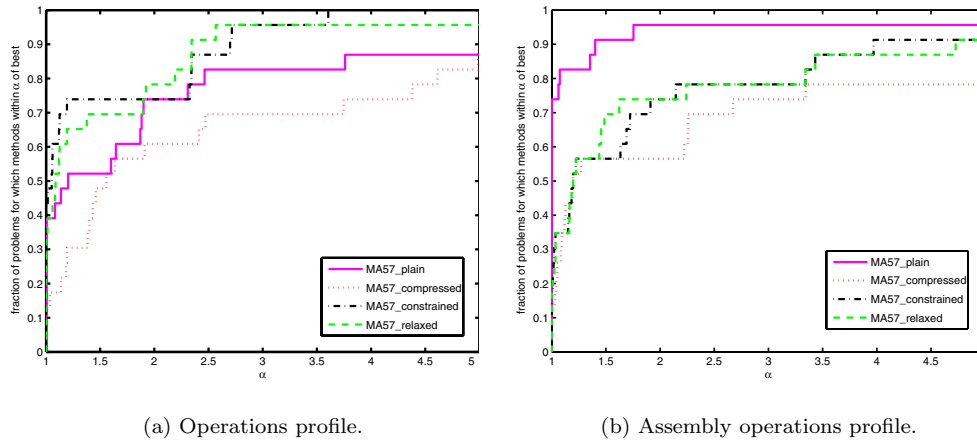
SYM set. This execution time difference is due to the fill-in and the number of operations. Indeed, the approach on the compressed graph merges some rows of the initial matrix, and the METIS ordering is called on a coarsened graph. Thus, the **MA57_compressed** pretreatment implies nonnegligible constraints on the ordering because of the a priori selection of the 2×2 pivots. These constraints severely degrade the quality of the ordering. In particular, they increase the memory requirement (see Figure 10(b)).

The **MA57_compressed** analysis succeeds in selecting good pivots for the factorization and delays fewer pivots than **MA57_plain** (see Figure 11(a)). Furthermore, the memory estimation is accurate for both approaches: the ratio between the predicted and the used memory remains close to 1 (see Figure 11(b)).

6.2. Augmented systems (AUGSYS set). We will now study the behavior of the **MA57** factorization when it is coupled with the three orderings **AMD**, **AMF**, and **METIS** on the *AUGSYS* set of our test matrices. We will see that the influence of our preprocessing depends on the ordering with which it is associated. We first do relative comparisons between the four **AMD**-based codes, between the two **METIS**-based codes, and between the four **AMF**-based codes. Then we select the best approaches and discuss them further.

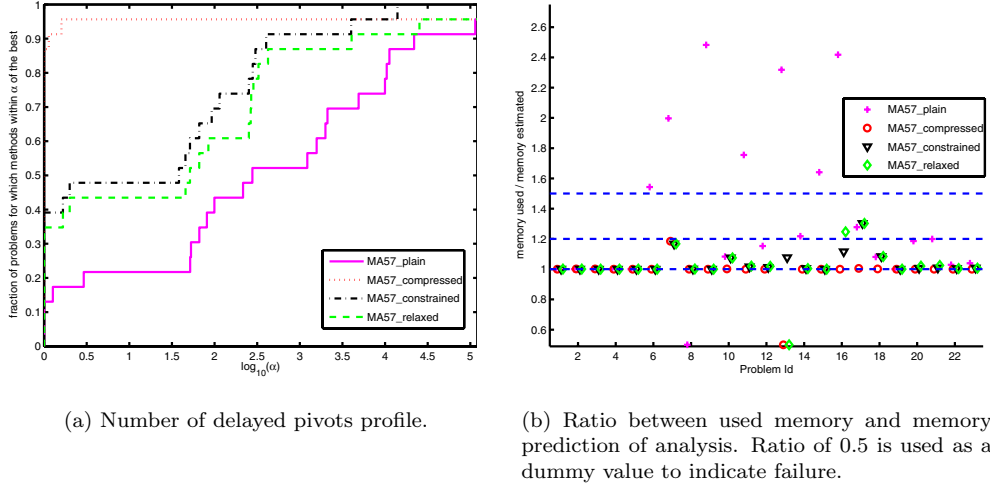
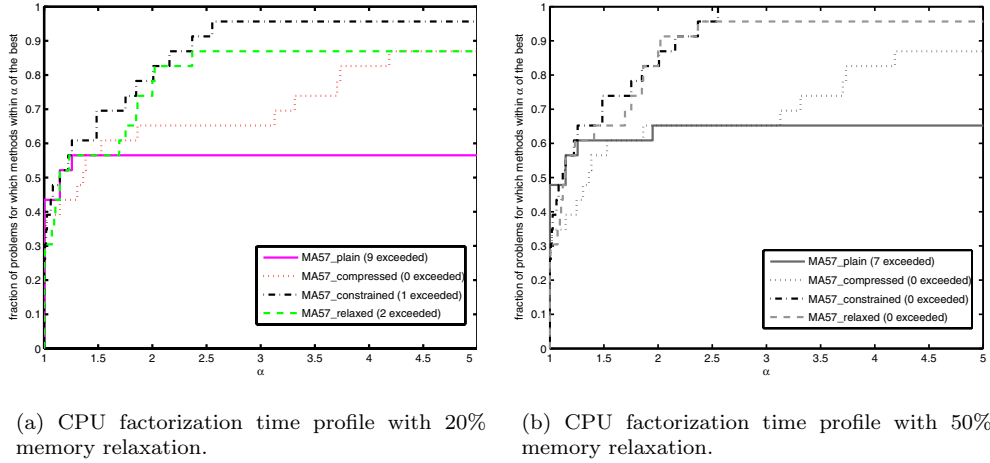
6.2.1. AMD-based approaches. Although **MA57_plain**, **MA57_compressed**, and **MA57_relaxed** fail on one matrix (they exceeded the CPU time limit on **NCVXQP7**), **MA57_constrained** does not fail. Figure 12(a) shows that, in terms of CPU factorization time, **MA57_plain** is within a factor of 1.25 of the best 75% of the time, while **MA57_constrained** and **MA57_relaxed** are comparable.

MA57_compressed can be far from the best ordering because of the constraints that we impose in the analysis phase. That is why it performs more computation and needs more memory than the other codes (see Figures 13(a) and 12(b)) and thus is also slower. **MA57_constrained** and **MA57_relaxed** do fewer operations than **MA57_plain**. Thus the total number of operations cannot explain why **MA57_plain** is faster on 75% of the problems. We also make the empirical observation that **MA57_constrained** and **MA57_relaxed** perform more assembly operations (which are more costly than eliminations) than **MA57_plain** (see Figure 13(b)). Indeed, the assembly operations in-

FIG. 12. *CPU factorization time and memory (AMD, AUGSYS set).*FIG. 13. *Number of floating-point operations (AMD, AUGSYS set).*

volve indirect addressing, whereas elimination operations call level 3 BLAS dense kernels, so that assembly operations are slower than eliminations, which explains why `MA57_constrained` and `MA57_relaxed` are slower than `MA57_plain`.

The three new approaches clearly improve the reliability of the analysis predictions. `MA57_compressed` decreases the number of delayed pivots as shown in Figure 14(a). Moreover, `MA57_compressed`, `MA57_constrained`, and `MA57_relaxed` improve the memory estimations. The memory estimates need to be increase by more than 50% for a successful factorization on seven problems for `MA57_plain` (see points above the 50% horizontal line of Figure 14(b)). For the other three options we never require more than 50% for a successful factorization.

FIG. 14. *Quality of analysis predictions (AMD, AUGSYS set).*FIG. 15. *CPU factorization time profile with relaxed memory (AMD, AUGSYS set).*

With respect to the above aspects (CPU factorization time, memory, number of delayed pivots, and quality of the analysis prediction), **MA57_constrained** seems to be the best if compromises have to be made between CPU time and memory for an **AMD**-based ordering (see Figure 15). **MA57_plain** remains the fastest on most of the problems if a large fixed amount of memory can be allocated for the factorization. That is why, concerning the **AMD**-based ordering, we keep the approach with only the **MC64SYM** scaling (**MA57_plain**) and the approach with a constrained ordering (**MA57_constrained**) for our final comparison.

6.2.2. METIS-based approaches. When our approaches are coupled with **METIS**, we did not get any failures. Moreover, 85% of the time **MA57_plain** is faster and **MA57_compressed** is within a factor of 1.6 of the best (see Figures 16(a) and 16(b)).

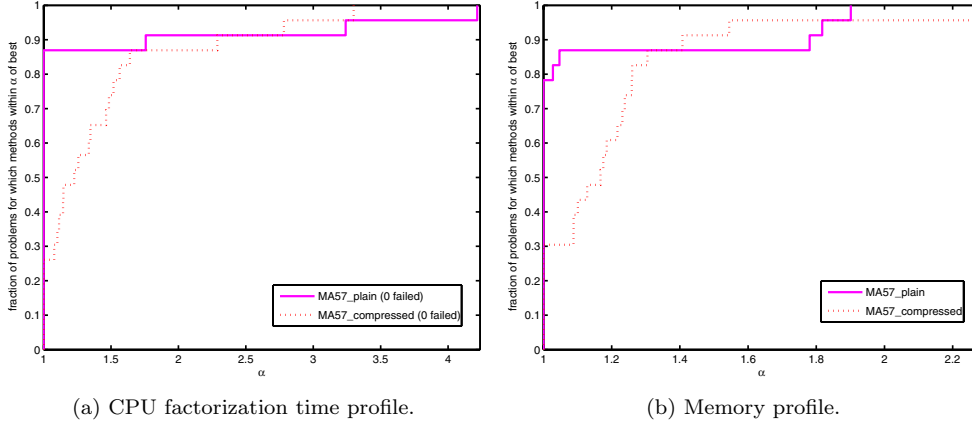


FIG. 16. Influence on the CPU factorization time and memory (METIS, AUGSYS set).

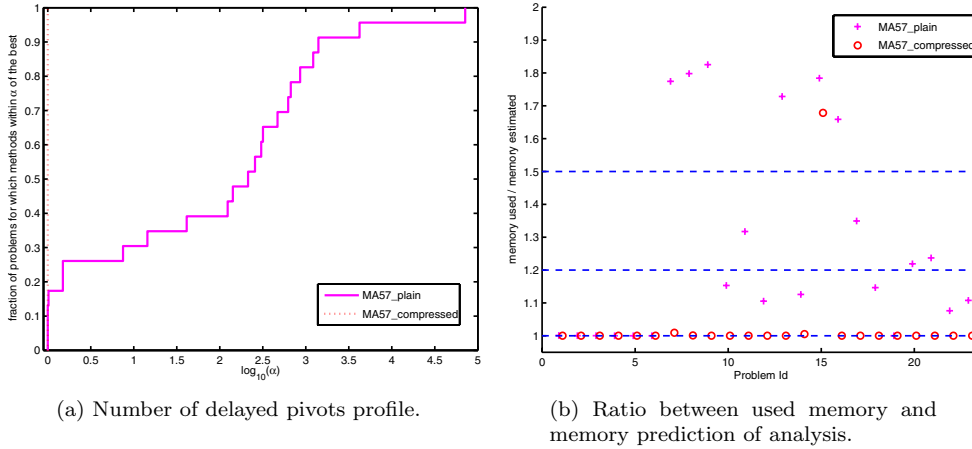


FIG. 17. Quality of analysis predictions (METIS, AUGSYS set).

Figure 17 shows that MA57_plain does not give an accurate prediction for the subsequent factorization. There are 10 and 6 matrices over the 20% and 50% limits, respectively. This can be explained by the huge number of delayed pivots (MA57_plain delays 10000 times more pivots than MA57_compressed in the worst cases; see Figure 17(a)). On the contrary, MA57_compressed exceeds the 20% limit only once, and otherwise the ratio between the memory predicted and the memory needed is always close to one. This better memory estimation is clearly shown by the profile of Figure 18, where MA57_compressed is faster than MA57_plain with 20% memory relaxation. With respect to the above comments, we keep the two METIS approaches for our final comparison.

6.2.3. AMF-based approaches. We observe a similar relative behavior between MA57_plain and MA57_compressed with an AMF-based ordering instead of an AMD-based one. The main advantage of an AMF-based ordering is that MA57_constrained and MA57_relaxed are not penalized by a large number of assembly operations. Thus MA57_constrained and MA57_relaxed are the fastest approaches (Figure 19(a)) and need

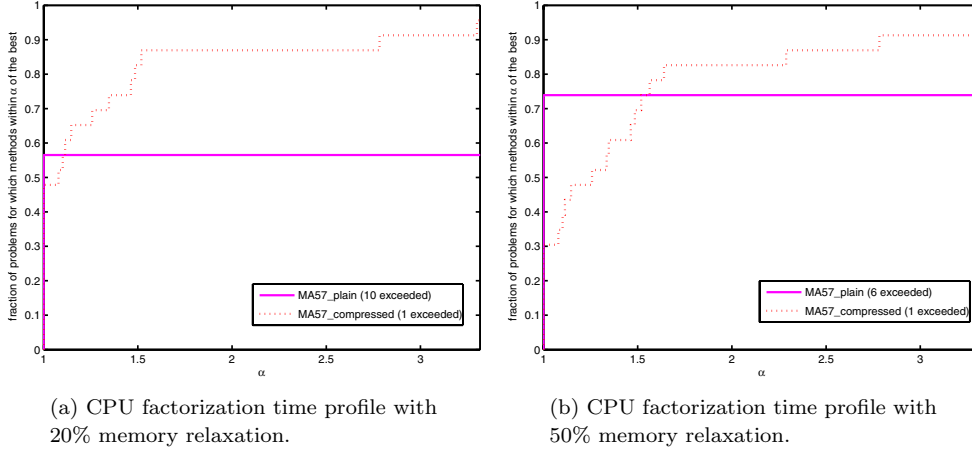


FIG. 18. CPU factorization time profile with relaxed memory (METIS, AUGSYS set).

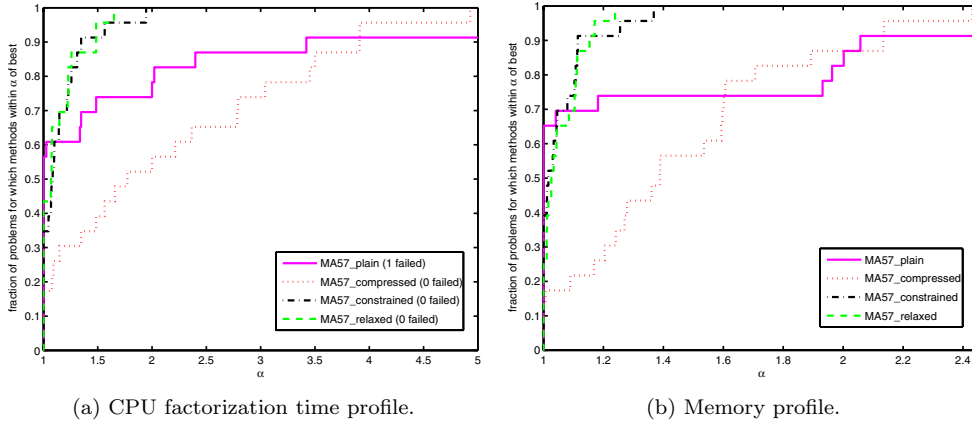
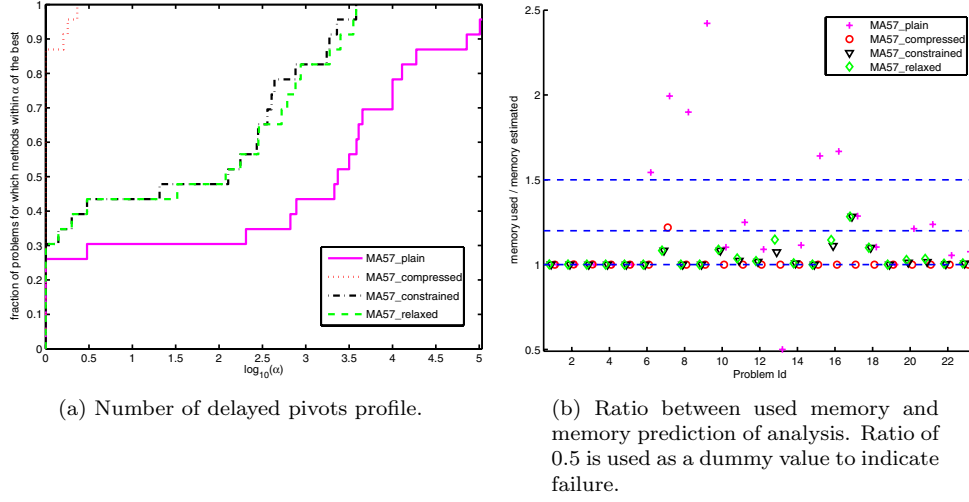
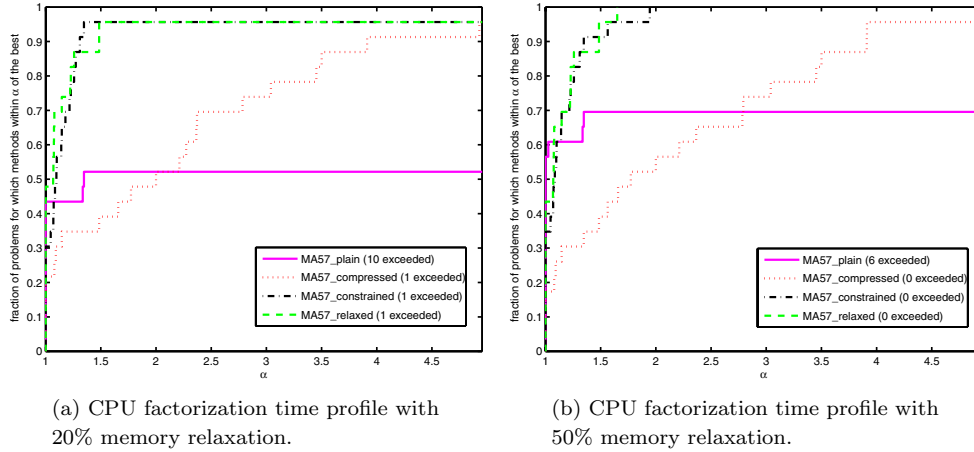


FIG. 19. Influence on the CPU factorization time and memory (AMF, AUGSYS set).

less memory than the other approaches (Figure 19(b)). Moreover, they decrease the number of delayed pivots (Figure 20(a)) and improve the memory estimation (Figure 20(b)). They decrease the number of failures from seven with MA57_plain to none if 50% relaxation is used between analysis and factorization. That is why they are also the best approaches in terms of CPU factorization time with a fixed memory relaxation as shown in Figure 21. For AMF, we keep only MA57_relaxed for our final comparison.

6.2.4. Best approaches discussion. In this section, we discuss the best approach for the AUGSYS set. We have kept five codes for the comparison because of the earlier discussion: MA57_plain/constrained with AMD, MA57_plain/compressed with METIS, and MA57_relaxed with AMF.

Figure 22 shows that the AMD-based approaches are the slowest and the most memory consuming. The pure METIS-based approach is the best on most of the problems, and the two other approaches (MA57_compressed(METIS) and MA57_relaxed(AMF)) are comparable. Figure 23 presents the factorization time if a relaxation parameter is used between analysis and factorization. With 20% relaxation, MA57_compressed(METIS) and MA57_relaxed(AMF) are clearly the best approaches (see Figure 23(a)). With 50% re-

FIG. 20. *Quality of analysis predictions (AMF, AUGSYS set).*FIG. 21. *CPU profile with relaxed memory (AMF, AUGSYS set).*

laxation, MA57_plain(MEIS) is the fastest on many problems, and MA57_relaxed(AMF) is the most robust (see Figure 23(b)).

Thus, if a large amount of memory is available, the approach with only the MC64SYM scaling and MEIS is sufficient. But, if the memory of the factorization needs to be estimated, we recommend the use of the MA57_relaxed+AMF approach.

7. Conclusions and future work. We have shown how MC64 can be used when the matrix is symmetric to effect a symmetric scaling and identify potential 2×2 pivots. We have observed that the use of an appropriate scaling (the MC64 symmetrized scaling, MC64SYM) solves many computational difficulties. We also noticed that MC77

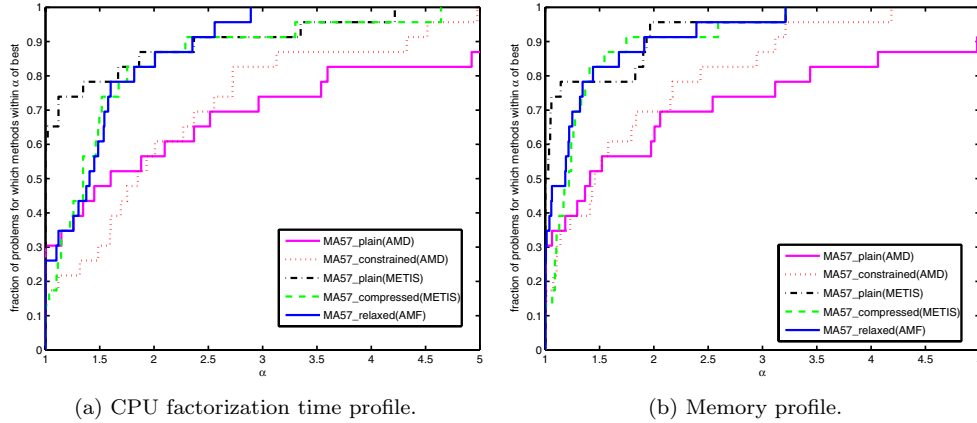


FIG. 22. AMD/METIS/AMF comparison on the AUGSYS set.

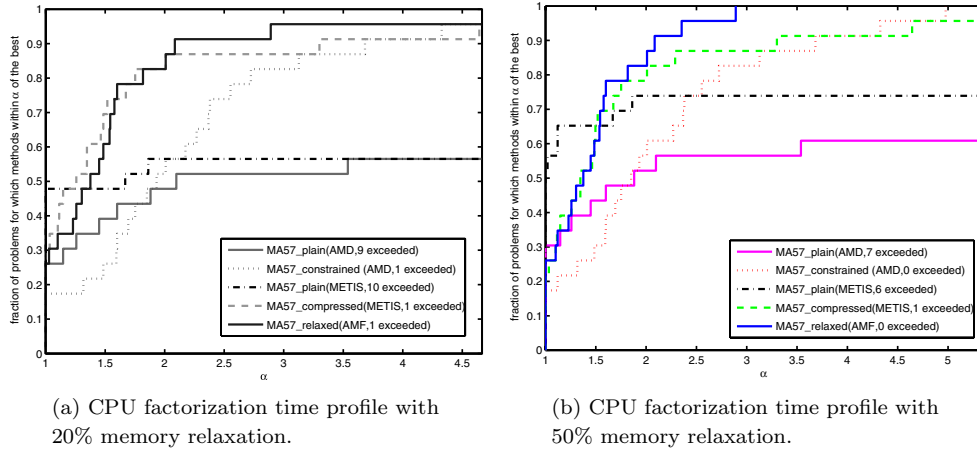


FIG. 23. AMD/METIS/AMF comparison (AUGSYS set): CPU factorization time profile with relaxed memory.

can sometimes be a good alternative and can benefit from its cheap cost. Perhaps this scaling has to be better understood before becoming a default approach.

The performance of MA57 depends very much on the nature of the matrix, but we have shown that it can benefit from this preprocessing, sometimes with no change, or only minor changes to the analysis. Another benefit of our work is that the analysis phase gives a better indication of the work and storage required by the subsequent factorization. In our future work, we will try to answer the following questions.

How do we decrease the fill-in in the MA57_compressed(METIS) analysis while keeping a stable factorization? We have succeeded in decreasing the number of operations in AMD and AMF with a (relaxed) constrained ordering. Fill-in may be lower if we extend this idea to have a tighter coupling with METIS. It is not obvious how to do this, so this will be one of our future research directions.

How can we adapt our approaches to a solver like MA47? MA47 is designed for augmented systems and will a priori do less computation, because it can manage the L and U pivots better.

How can we design a code with static 2×2 and 1×1 pivoting? Static pivoting can address a large variety of problems in the unsymmetric case when it is coupled with MC64 preprocessing and has the advantage of giving exact memory estimation. It is also more friendly for a parallel distributed implementation. Can we adapt the unsymmetric static pivoting to the symmetric indefinite case, and does it address a large range of symmetric indefinite matrices?

Acknowledgments. We are very grateful to Jacko Koster, Daniel Ruiz, and John Gilbert for discussions on various aspects of this work and to Jennifer Scott for her comments on this paper. We would like to thank the referees for providing useful and constructive remarks for improving the presentation of this paper.

REFERENCES

- [1] P. R. AMESTOY, T. A. DAVIS, AND I. S. DUFF, *An approximate minimum degree ordering algorithm*, SIAM J. Matrix Anal. Appl., 17 (1996), pp. 886–905.
- [2] C. ASHCRAFT, *Compressed graphs and the minimum degree algorithm*, SIAM J. Sci. Comput., 16 (1995), pp. 1404–1411.
- [3] J. R. BUNCH AND L. KAUFMAN, *Some stable methods for calculating inertia and solving symmetric linear systems*, Math. Comp., 31 (1977), pp. 162–179.
- [4] J. R. BUNCH AND B. N. PARLETT, *Direct methods for solving symmetric indefinite systems of linear systems*, SIAM J. Numer. Anal., 8 (1971), pp. 639–655.
- [5] A. R. CURTIS AND J. K. REID, *On the automatic scaling of matrices for Gaussian elimination*, J. Inst. Math. Appl., 10 (1972), pp. 118–124.
- [6] T. A. DAVIS, *University of Florida Sparse Matrix Collection*, 2002, <http://www.cise.ufl.edu/research/sparse/matrices/>.
- [7] E. D. DOLAN AND J. J. MORÉ, *Benchmarking optimization software with performance profiles*, Math. Program., 91 (2002), pp. 201–213.
- [8] I. S. DUFF, *MA57—a code for the solution of sparse symmetric definite and indefinite systems*, ACM Trans. Math. Software, 30 (2004), pp. 118–144.
- [9] I. S. DUFF AND J. R. GILBERT, *Maximum-weight matching and block pivoting for symmetric indefinite systems*, in an abstract book of Householder Symposium XV, Peebles, Scotland, 2002.
- [10] I. S. DUFF AND J. KOSTER, *The design and use of algorithms for permuting large entries to the diagonal of sparse matrices*, SIAM J. Matrix Anal. Appl., 20 (1999), pp. 889–901.
- [11] I. S. DUFF AND J. KOSTER, *On algorithms for permuting large entries to the diagonal of a sparse matrix*, SIAM J. Matrix Anal. Appl., 22 (2001), pp. 973–996.
- [12] I. S. DUFF AND S. PRALET, *Experiments in Preprocessing and Scaling Symmetric Problems for Multifrontal Solutions*, Tech. report WN/PA/04/17, CERFACS, Toulouse, France, 2004.
- [13] I. S. DUFF AND J. K. REID, *The multifrontal solution of indefinite sparse symmetric linear systems*, ACM Trans. Math. Software, 9 (1983), pp. 302–325.
- [14] I. S. DUFF AND J. K. REID, *The multifrontal solution of unsymmetric sets of linear systems*, SIAM J. Sci. Statist. Comput., 5 (1984), pp. 633–641.
- [15] I. S. DUFF AND J. K. REID, *MA47, a Fortran Code for Direct Solution of Indefinite Sparse Symmetric Linear Systems*, Tech. report RAL 95-001, Rutherford Appleton Laboratory, Oxon, UK, 1995.
- [16] I. S. DUFF AND J. K. REID, *Exploiting zeros on the diagonal in the direct solution of indefinite sparse symmetric linear systems*, ACM Trans. Math. Software, 22 (1996), pp. 227–257.
- [17] J. EDMONDS, *Maximum matching and a polyhedron with 0,1-vertices*, J. Res. Nat. Bur. Standards Sect. B, 69 (1965), pp. 125–130.
- [18] J. EDMONDS, *Paths, trees and flowers*, Canad. J. Math., 17 (1965), pp. 449–467.
- [19] H. N. GABOW, *Data structures for weighted matching and nearest common ancestors with linking*, in Proceedings of the First Annual ACM-SIAM Symposium on Discrete Algorithms, SIAM, Philadelphia, 1990, pp. 434–443.
- [20] N. I. M. GOULD, D. ORBAN, AND P. L. TOINT, *CUTEr and SifDec: A constrained and unconstrained testing environment, revisited*, ACM Trans. Math. Software, 29 (2003), pp. 373–394.
- [21] N. I. M. GOULD AND J. A. SCOTT, *A numerical evaluation of HSL packages for the direct solution of large sparse, symmetric linear systems of equations*, ACM Trans. Math. Software,

- 30 (2004), pp. 300–325.
- [22] P. HEGGERNES, S. EISENSTAT, G. KUMFERT, AND A. POTHEN, *The computational complexity of the minimum degree algorithm*, in Proceedings of the Norwegian Conference on Computer Science NIK, University of Tromsø, Norway, 2002.
 - [23] HSL, *A Collection of Fortran Codes for Large Scale Scientific Computation*, 2004.
 - [24] G. KARYPIS AND V. KUMAR, *MEPS – A Software Package for Partitioning Unstructured Graphs, Partitioning Meshes, and Computing Fill-Reducing Orderings of Sparse Matrices – Version 4.0*, University of Minnesota, Minneapolis, MN, 1998.
 - [25] G. KUMFERT AND A. POTHEN, *Two improved algorithms for envelope and wavefront reduction*, BIT, 37 (1997), pp. 559–590.
 - [26] J. W. H. LIU, *The role of elimination trees in sparse factorization*, SIAM J. Matrix Anal. Appl., 11 (1990), pp. 134–172.
 - [27] I. MAROS AND C. MESZAROS, *A repository of convex quadratic programming problems*, Optim. Methods Softw., 11/12 (1999), pp. 671–681.
 - [28] E. G. NG AND P. RAGHAVAN, *Performance of greedy ordering heuristics for sparse Cholesky factorization*, SIAM J. Matrix Anal. Appl., 20 (1999), pp. 902–914.
 - [29] S. PRALET, *Constrained Orderings and Scheduling for Parallel Sparse Linear Algebra*, Ph.D. thesis, CERFACS, Toulouse, France, 2004, TH/PA/04/105.
 - [30] E. ROTHBERG AND S. C. EISENSTAT, *Node selection strategies for bottom-up sparse matrix ordering*, SIAM J. Matrix Anal. Appl., 19 (1998), pp. 682–695.
 - [31] D. RUIZ, *A Scaling Algorithm to Equilibrate Both Row and Column Norms in Matrices*, Tech. report RAL-TR-2001-034, Rutherford Appleton Laboratory, Oxon, UK, 2001; also appeared as ENSEEIHT-IRIT report RT/APO/01/4.

SOME FAST ALGORITHMS FOR SEQUENTIALLY SEMISEPARABLE REPRESENTATIONS*

S. CHANDRASEKARAN[†], P. DEWILDE[‡], M. GU[§], T. PALS[¶], X. SUN^{||},
A.-J. VAN DER VEEN[‡], AND D. WHITE^{**}

Abstract. An extended sequentially semiseparable (SSS) representation derived from time-varying system theory is used to capture, on the one hand, the low-rank of the off-diagonal blocks of a matrix for the purposes of efficient computations and, on the other, to provide for sufficient descriptive richness to allow for backward stability in the computations. We present (i) a fast algorithm (linear in the number of equations) to solve least squares problems in which the coefficient matrix is in SSS form, (ii) a fast algorithm to find the SSS form of X such that $AX = B$, where A and B are in SSS form, and (iii) a fast model reduction technique to improve the SSS form.

Key words. backward stability, fast direct solver, fast multipole method, integral equations, method of moments, scattering theory, semiseparable matrices, spectral methods, Moore–Penrose inverses, time-varying systems

AMS subject classifications. 65F05, 65F20, 65F30

DOI. 10.1137/S0895479802405884

1. Introduction. In this paper we will present fast backward stable algorithms for the solution of a large class of least squares problems with structured matrices which can be either sparse or dense. The structure concerned is called “semiseparable” and is a matrix analogue of semiseparable integral kernels as described by Kailath in [33]. This matrix analogue was most likely first described by Gohberg, Kailath, and Koltracht in [22]. In that paper it is shown that, under further technical restrictions, an LDU factorization is possible with a complexity p^2N , where p is the complexity of the semiseparable description and N the dimension of the matrix, in effect an algorithm linear in the size of the matrix, when p is small. In a number of papers Alpay, Dewilde, and Dym introduce a new formalism for time-varying systems which provides for a framework closely analogous to the classical time-invariant state space description and which allows for the generalization of many time-invariant methods to the time-varying case [1, 2]. When applied to matrices, this formalism generalizes the formalism used in [22] and allows for more general types of efficient operations. (By

*Received by the editors April 17, 2002; accepted for publication (in revised form) by S. Van Huffel October 25, 2004; published electronically September 15, 2005.

<http://www.siam.org/journals/simax/27-2/40588.html>

[†]Department of Electrical and Computer Engineering, University of California at Santa Barbara, Santa Barbara, CA 93106-9560 (shiv@ece.ucsb.edu). This author’s research was supported in part by grant B521537 from Lawrence Livermore National Laboratory, NSF Career Award CCR-9734290, and NSF grant CCR-0204388.

[‡]Faculty of Electrical Engineering, Delft University of Technology, Mekelweg 4, 2628 CD Delft, The Netherlands (p.dewilde@its.tudelft.nl, A.J.vanderVeen@its.tudelft.nl).

[§]Department of Mathematics, University of California at Berkeley, Berkeley, CA 94720-3840 (mgu@math.berkeley.edu). This author’s research was supported in part by NSF grant CCR-0204388 and Lawrence Livermore National Laboratory grant ISCR-03-24.

[¶]Department of Electrical and Computer Engineering, University of California at Santa Barbara (tim@kipling.ece.ucsb.edu). This author’s research was supported in part by Lawrence Livermore National Laboratory grant ISCR-03-24.

^{||}Department of Mechanical Engineering, University of California at Berkeley (sunx@me.berkeley.edu).

^{**}Center for Applied Scientific Computing, Lawrence Livermore National Laboratory, P.O. Box 808, L-561, Livermore, CA 94551 (dwhite@llnl.gov).

TABLE 1

Dimensions of matrices in (1). k_i and l_i are column dimensions of U_i and P_i , respectively.

Matrix	U_i	V_i	W_i	P_i	Q_i	R_i
Dimensions	$m_i \times k_i$	$m_i \times k_{i-1}$	$k_{i-1} \times k_i$	$m_i \times l_i$	$m_i \times l_{i+1}$	$l_{i+1} \times l_i$

“efficient” we mean operations that are linear in the size of the matrix.) In the book [16], Dewilde and van der Veen describe the various operations that are possible on time-varying systems in great detail, including the efficient application of orthogonal transformations. In particular, they show how a URV -type transformation on a general (possibly infinite dimensional) semiseparable system can be done with an efficient recursive procedure. This procedure is based on the ideas presented in [40] and then further elaborated by Dewilde and van der Veen in [17] and by Eidelman and Gohberg in [21]. In the former paper the connection with Kalman filtering as a special case of the procedures is also discussed. Other related works include [35, 41].

To be more specific, let A be an $N \times N$ (possibly complex) matrix satisfying the matrix structure. Then there exist n positive integers $m_1, \dots, m_n$ with $N = m_1 + \dots + m_n$ to block-partition A as

$$A = (A_{ij}), \quad \text{where } A_{ij} \in \mathbf{C}^{m_i \times m_j} \text{ satisfies } A_{ij} = \begin{cases} D_i & \text{if } i = j, \\ U_i W_{i+1} \cdots W_{j-1} V_j^H & \text{if } j > i, \\ P_i R_{i-1} \cdots R_{j+1} Q_j^H & \text{if } j < i. \end{cases} \quad (1)$$

Here we use the superscript H to denote the Hermitian transpose. The sequences $\{U_i\}_{i=1}^{n-1}$, $\{V_i\}_{i=2}^n$, $\{W_i\}_{i=2}^{n-1}$, $\{P_i\}_{i=2}^n$, $\{Q_i\}_{i=1}^{n-1}$, $\{R_i\}_{i=2}^{n-1}$, and $\{D_i\}_{i=1}^n$ are all matrices whose dimensions are defined in Table 1. While any matrix can be represented in this form for large enough k_i 's and l_i 's, our main focus will be on matrices of this special form that have relatively small values for the k_i 's and l_i 's. In the above equation, empty products are defined to be the identity matrix. For $n = 4$, the matrix A has the form

$$A = \begin{pmatrix} D_1 & U_1 V_2^H & U_1 W_2 V_3^H & U_1 W_2 W_3 V_4^H \\ P_2 Q_1^H & D_2 & U_2 V_3^H & U_2 W_3 V_4^H \\ P_3 R_2 Q_1^H & P_3 Q_2^H & D_3 & U_3 V_4^H \\ P_4 R_3 R_2 Q_1^H & P_4 R_3 Q_2^H & P_4 Q_3^H & D_4 \end{pmatrix}.$$

We say that the matrices D_i , U_i , W_i , V_i , P_i , Q_i , and R_i provide a sequentially semiseparable (SSS) representation of the matrix A if they satisfy (1).

In the case where all W_i and R_i are identities, A reduces to a block-diagonal plus semiseparable matrix, which can be handled directly using techniques in Chandrasekaran and Gu [8]. It is shown in [16] that this class of matrices is closed under inversion and includes banded matrices and semiseparable matrices, as well as their inverses as special cases.

It should be noted that the SSS structure of a given matrix A depends on the sequence m_i . Different sequences will lead to different representations.

In this paper we present a fast backward stable algorithm for solving a linear least squares problem where the coefficient matrix is a rectangular matrix with a structure similar to (1). As a consequence, this work effectively shows that the Moore–Penrose inverses of such matrices have similar structures and can be computed rapidly as

well. We also present a fast algorithm for computing the SSS representation of X , such that $AX = B$, when the SSS representation of both A and B are known. For both these algorithms we present numerical experiments on timing and accuracy. Finally, we present a fast model reduction algorithm for improving the efficiency of the SSS representation when necessary. This latter part of the paper is, in effect, a generalization of the method presented in [8], which is not repeated here.

A companion technical report [7] is also available and is more comprehensive in the topics it covers.

2. One-pass solution for the Moore–Penrose inverse of a general system. Given a general matrix A in SSS form and a vector b with the same number of rows as A , we wish to find a vector x of smallest possible magnitude $\|x\|_2 = \sqrt{\sum \|x_i\|^2}$, which minimizes $\|Ax - b\|_2$. Our goal is to find a backward stable algorithm that is one-pass and top-down. As before, we assume that we have a realization consisting of block matrices $\{U_i\}$, $\{V_i\}$, etc., with the only restriction being that these realization matrices must be appropriately sized to fit the products in the representation. In particular, the diagonal entries $\{D_i\}$ do not have to be square or nonsingular. In [16] a general algorithm was given to determine the Moore–Penrose inverse of a general time-varying operator. In this paper, we adapt this algorithm to the matrix case using the ideas of [6]. This results in a fast one-pass and top-down algorithm for the structured least squares problem.

The one-pass character is obtained by making judicious use of left and right multiplications by elementary orthogonal matrices. These matrices then combine to form semiseparable operators of the same complexity as the original semiseparable matrix. What makes the Moore–Penrose case more complex than the square invertible case is the fact that exact solutions of reduced subsystems may not contribute to the Moore–Penrose solution, except in the special circumstance that all rows in the system are linearly independent. For example, the system of equations

$$\begin{bmatrix} 1 \\ 1 \end{bmatrix} x = \begin{bmatrix} b_1 \\ b_2 \end{bmatrix}$$

has the least squares solution $x = (b_1 + b_2)/2$; that is, the first equation cannot be solved independently from the rest of the system. Hence care has to be exercised in converting the system first to a system of row-independent matrices; that is, the left null-space has to be implicitly determined (see what follows).

The strategy we follow to obtain the reduction to a row-independent form works in two steps and produces a system of independent rows in block upper triangular form. To find the Moore–Penrose solution a further step then consists in solving the resulting row-independent system using the method described in [6] (this time actually on a simpler system) or just by constructing the inverse directly as is done in [16]. The resulting complexity will roughly be twice that of the direct inversion algorithm, in which invertibility of the matrix is assumed. Lazy evaluation allows the algorithm to proceed from the upper left corner down to the lower right corner as before. A summary of the procedure is as follows:

1. Using elementary orthogonal (or in case of complex entries unitary) transforms on the columns (on the right of the matrix), we gradually transform the double-sided representation into a block upper triangular one, that is, one in which the P -, Q -, and R -type matrices are all zero. The block upper triangular representation will have the same complexity as the original. The result is the representation of a block upper triangular matrix A_1 , related to

A through

$$A_1 = A\hat{Q},$$

where $\hat{Q}$ assembles the column transformations. (We keep using the symbol “ Q ” with a qualification such as a hat or a tilde to indicate unitary transformation matrices. These Q ’s should not be confused with the Q_i in the original representation. A similar remark applies to the use of R ’s.)

2. Using elementary orthogonal transformations on the rows (i.e., on the left) we gradually (lazily) determine the left null-space and range of A_1 :

$$A_1 = [\tilde{Q}^{(1)} \quad \tilde{Q}^{(2)}] \begin{bmatrix} 0 \\ A_o \end{bmatrix},$$

in which A_o has linearly independent rows. The columns of $\tilde{Q}^{(2)}$ then form an orthonormal basis for the range of A_1 and the columns of $\tilde{Q}^{(1)}$ for its left null-space (i.e., the kernel of A_1^H).

3. The combination of the two previous steps has produced

$$A = [\tilde{Q}^{(1)} \quad \tilde{Q}^{(2)}] \begin{bmatrix} 0 \\ A_o \end{bmatrix} \hat{Q}.$$

With $y = \hat{Q}x$ and $\begin{bmatrix} b_o^{(1)} \\ b_o^{(2)} \end{bmatrix} = \begin{bmatrix} \tilde{Q}^{(1)H}b \\ \tilde{Q}^{(2)H}b \end{bmatrix}$ we find that the Moore–Penrose solution satisfies

$$A_o y = b_o^{(2)},$$

which is now a system with linearly independent rows that can be solved exactly with the techniques of [6, 7], leaving an irreducible residue $b_o^{(1)}$. Hence the least squares error will be $\|b_o^{(1)}\|$.

We use the following 4×4 -block SSS matrix as an example to illustrate the procedure:

$$(2) \quad A = \begin{bmatrix} D_1 & U_1 V_2^H & U_1 W_2 V_3^H & U_1 W_2 W_3 V_4^H \\ P_2 Q_1^H & D_2 & U_2 V_3^H & U_2 W_3 V_4^H \\ P_3 R_2 Q_1^H & P_3 Q_2^H & D_3 & U_3 V_4^H \\ P_4 R_3 R_2 Q_1^H & P_4 R_3 Q_2^H & P_4 Q_3^H & D_4 \end{bmatrix}.$$

2.1. Right multiplication—Converting to a block upper triangular matrix.

1. Compute $Q_1 = [\hat{Q}_{11} \quad \hat{Q}_{12}] \begin{bmatrix} 0 \\ \hat{R}_2 \end{bmatrix}$ via QL factorization. $\hat{R}_2$ is a square matrix.
2. Right multiply $\begin{bmatrix} \hat{Q}_{11} & \hat{Q}_{12} \end{bmatrix}$ to the first block column of A and obtain

$$(3) \quad \begin{bmatrix} D_1 \hat{Q}_{11} & D_1 \hat{Q}_{12} & U_1 V_2^H & U_1 W_2 V_3^H & U_1 W_2 W_3 V_4^H \\ 0 & P_2 \hat{R}_2^H & D_2 & U_2 V_3^H & U_2 W_3 V_4^H \\ 0 & P_3 R_2 \hat{R}_2^H & P_3 Q_2^H & D_3 & U_3 V_4^H \\ 0 & P_4 R_3 R_2 \hat{R}_2^H & P_4 R_3 Q_2^H & P_4 Q_3^H & D_4 \end{bmatrix} \\ = \begin{bmatrix} A_1^{(1)} & \begin{bmatrix} D_1 \hat{Q}_{12} & U_1 V_2^H \\ P_2 \hat{R}_2^H & D_2 \\ P_3 R_2 \hat{R}_2^H & P_3 Q_2^H \\ P_4 R_3 R_2 \hat{R}_2^H & P_4 R_3 Q_2^H \end{bmatrix} & \tilde{A}_1^{(1)} \end{bmatrix}.$$

Step 2

1. Compute $\begin{bmatrix} \hat{R}_2 R_2^H \\ Q_2 \end{bmatrix} = \begin{bmatrix} \hat{Q}_{21}^{(1)} & \hat{Q}_{22}^{(1)} \\ \hat{Q}_{21}^{(2)} & \hat{Q}_{22}^{(2)} \end{bmatrix} \begin{bmatrix} 0 \\ \hat{R}_3 \end{bmatrix}$ via QL factorization. $\hat{R}_3$ is a square matrix.
2. Right multiply $\begin{bmatrix} \hat{Q}_{21}^{(1)} & \hat{Q}_{22}^{(1)} \\ \hat{Q}_{21}^{(2)} & \hat{Q}_{22}^{(2)} \end{bmatrix}$ to the 2nd and 3rd block columns of A after Step 1, and we obtain

$$(4) \quad \begin{bmatrix} A_1^{(1)} \left[\begin{array}{cc|cc} [D_1 \hat{Q}_{12} & U_1 V_2^H] \hat{Q}_{21} & [D_1 \hat{Q}_{12} & U_1 V_2^H] \hat{Q}_{22} \\ [P_2 \hat{R}_2^H & D_2] \hat{Q}_{21} & [P_2 \hat{R}_2^H & D_2] \hat{Q}_{22} \\ 0 & P_3 \hat{R}_3^H \\ 0 & P_4 R_3 \hat{R}_3^H \end{array} \right] \tilde{A}_1^{(1)} \end{bmatrix} \\ = \begin{bmatrix} A_1^{(2)} \left[\begin{array}{cc|cc} [D_1 \hat{Q}_{12} & U_1 V_2^H] \hat{Q}_{22} & U_1 W_2 V_3^H \\ [P_2 \hat{R}_2^H & D_2] \hat{Q}_{22} & U_2 V_3^H \\ P_3 \hat{R}_3^H & D_3 \\ P_4 R_3 \hat{R}_3^H & P_4 Q_3^H \end{array} \right] \tilde{A}_1^{(2)} \end{bmatrix}.$$

Step 3

1. Compute $\begin{bmatrix} \hat{R}_3 R_3^H \\ Q_3 \end{bmatrix} = \begin{bmatrix} \hat{Q}_{31}^{(1)} & \hat{Q}_{32}^{(1)} \\ \hat{Q}_{31}^{(2)} & \hat{Q}_{32}^{(2)} \end{bmatrix} \begin{bmatrix} 0 \\ \hat{R}_4 \end{bmatrix}$ via QL factorization. $\hat{R}_4$ is a square matrix.
2. Right multiply $\begin{bmatrix} \hat{Q}_{31}^{(1)} & \hat{Q}_{32}^{(1)} \\ \hat{Q}_{31}^{(2)} & \hat{Q}_{32}^{(2)} \end{bmatrix}$ to the 3rd and 4th block columns of A after Step 2, and we obtain

$$(5) \quad \begin{bmatrix} A_1^{(3)} \left[\begin{array}{ccc|c} [D_1 \hat{Q}_{12} & U_1 V_2^H] \hat{Q}_{22} & U_1 W_2 V_3^H & \hat{Q}_{32} & U_1 W_2 W_3 V_4^H \\ [P_2 \hat{R}_2^H & D_2] \hat{Q}_{22} & U_2 V_3^H & \hat{Q}_{32} & U_2 W_3 V_4^H \\ [P_3 \hat{R}_3^H & D_3] \hat{Q}_{32} & U_3 V_4^H & & U_3 W_4^H \\ P_4 \hat{R}_4^H & & D_4 & & \end{array} \right] \end{bmatrix},$$

where

$$(6) \quad A_1^{(3)} = \begin{bmatrix} A_1^{(2)} \left[\begin{array}{ccc|c} [D_1 \hat{Q}_{12} & U_1 V_2^H] \hat{Q}_{22} & U_1 W_2 V_3^H & \hat{Q}_{31} \\ [P_2 \hat{R}_2^H & D_2] \hat{Q}_{22} & U_2 V_3^H & \hat{Q}_{31} \\ [P_3 \hat{R}_3^H & D_3] \hat{Q}_{31} & & \\ 0 & & & \end{array} \right] \end{bmatrix}.$$

2.2. Time-varying system notation representation. We will use the following time-varying system notations to preserve and represent the SSS structure for the reduction procedure described above. If A is a block upper triangular matrix with an SSS representation, we use

$$\langle A \rangle_i = \left[\begin{array}{c|c} W_i & V_i^H \\ \hline U_i & D_i \end{array} \right],$$

collecting the i th components of the SSS representation in a single block 2×2 matrix. In the above notation, some of the block rows or columns may disappear. The dimensions of the blocks have to be congruent, of course. The entry $A_{i,j}$ for $i < j$ is given by $A_{i,j} = U_i W_{i+1} \cdots W_{j-1} V_j$ and $A_{i,i} = D_i$.

We use the terms “realization” and “representation” indiscriminately to indicate the matrices used in diverse semiseparable representations. In the next sections we assume that all the given representations are of minimal dimensions. (They will be if the construction procedures of [6] have been used.)

With these notations, we rewrite the right multiplication procedure as follows.

- Step 1
1. Compute $Q_1 = [\hat{Q}_{11} \quad \hat{Q}_{12}] [\hat{R}_2^0]$ via QL.
 2. $\langle \hat{Q} \rangle_1$

$$= \left[\begin{array}{c|c} \cdot & \cdot \\ \hline \hat{Q}_{12} & \hat{Q}_{11} \end{array} \right],$$

where “.” denotes a 0-dimension matrix.

3. $\langle A_1 \rangle_1$

$$= \left[\begin{array}{c|c} \cdot & \cdot \\ \hline [D_1 \hat{Q}_{12} \quad U_1] & D_1 \hat{Q}_{11} \end{array} \right].$$

- Step 2
1. Compute $[\hat{R}_2 R_2^H] = [\hat{Q}_{21}^{(1)} \quad \hat{Q}_{22}^{(1)}] [\hat{R}_3^0]$ via QL.
 2. $\langle \hat{Q} \rangle_2$

$$= \left[\begin{array}{c|c} \hat{Q}_{22}^{(1)} & \hat{Q}_{21}^{(1)} \\ \hline \hat{Q}_{22}^{(2)} & \hat{Q}_{21}^{(2)} \end{array} \right].$$

3. $\langle A_1 \rangle_2$

$$= \left[\begin{array}{c|c} \begin{bmatrix} \hat{Q}_{22}^{(1)} & 0 \\ V_2^H \hat{Q}_{22}^{(2)} & W_2 \end{bmatrix} & \begin{bmatrix} \hat{Q}_{21}^{(1)} \\ V_2^H \hat{Q}_{21}^{(2)} \end{bmatrix} \\ \hline [P_2 \hat{R}_2^H \hat{Q}_{22}^{(1)} + D_2 \hat{Q}_{22}^{(2)} \quad U_2] & P_2 \hat{R}_2^H \hat{Q}_{21}^{(1)} + D_2 \hat{Q}_{21}^{(2)} \end{array} \right].$$

On completing this step, we are able to construct the first two block columns of A_1 :

$$(7) \quad A_1 = \begin{bmatrix} D_1 \hat{Q}_{11} & [D_1 \hat{Q}_{12} \quad U_1] \begin{bmatrix} \hat{Q}_{21}^{(1)} \\ V_2^H \hat{Q}_{21}^{(2)} \end{bmatrix} & ? & ? \\ 0 & P_2 \hat{R}_2^H \hat{Q}_{21}^{(1)} + D_2 \hat{Q}_{21}^{(2)} & ? & ? \\ 0 & 0 & ? & ? \\ 0 & 0 & ? & ? \end{bmatrix} = \begin{bmatrix} A_1^{(2)} & \begin{bmatrix} ? & ? \\ ? & ? \\ ? & ? \\ ? & ? \end{bmatrix} \end{bmatrix}.$$

- Step 3
1. Compute $[\hat{R}_3 R_3^H] = [\hat{Q}_{31}^{(1)} \quad \hat{Q}_{32}^{(1)}] [\hat{R}_4^0]$ via QL.
 2. $\langle \hat{Q} \rangle_3$

$$= \left[\begin{array}{c|c} \hat{Q}_{32}^{(1)} & \hat{Q}_{31}^{(1)} \\ \hline \hat{Q}_{32}^{(2)} & \hat{Q}_{31}^{(2)} \end{array} \right].$$

3. $\langle A_1 \rangle_3$

$$= \left[\begin{array}{c|c} \begin{bmatrix} \hat{Q}_{32}^{(1)} & 0 \\ V_3^H \hat{Q}_{32}^{(2)} & W_3 \end{bmatrix} & \begin{bmatrix} \hat{Q}_{31}^{(1)} \\ V_3^H \hat{Q}_{31}^{(2)} \end{bmatrix} \\ \hline [P_3 \hat{R}_3^H \hat{Q}_{32}^{(1)} + D_3 \hat{Q}_{32}^{(2)} \quad U_3] & P_3 \hat{R}_3^H \hat{Q}_{31}^{(1)} + D_3 \hat{Q}_{31}^{(2)} \end{array} \right].$$

On completing this step, we are able to construct the first three block columns of A_1 :

$$(8) \quad A_1 = \left[\begin{array}{c} A_1^{(2)} \left[\begin{array}{c} [D_1 \hat{Q}_{12} \quad U_1] \begin{bmatrix} \hat{Q}_{21}^{(1)} & 0 \\ V_2^H \hat{Q}_{21}^{(2)} & W_2 \end{bmatrix} \begin{bmatrix} \hat{Q}_{32}^{(1)} \\ V_3^H \hat{Q}_{32}^{(2)} \end{bmatrix} \\ [P_2 \hat{R}_2^H \hat{Q}_{22}^{(1)} + D_2 \hat{Q}_{22}^{(2)} \quad U_2] \begin{bmatrix} \hat{Q}_{32}^{(1)} \\ V_3^H \hat{Q}_{32}^{(2)} \end{bmatrix} \\ P_3 \hat{R}_3^H \hat{Q}_{31}^{(1)} + D_3 \hat{Q}_{31}^{(2)} \\ 0 \end{array} \right] \begin{bmatrix} ? \\ ? \\ ? \\ ? \end{bmatrix} \end{array} \right] = \left[\begin{array}{c} A_1^{(3)} \begin{bmatrix} ? \\ ? \\ ? \end{bmatrix} \end{array} \right].$$

4 This is the last step for our 4×4 -block example, and there is nothing left to be done except the calculation of the last block column:

$$\langle A_1 \rangle_4 = \left[\begin{array}{c|c} \cdot & \begin{bmatrix} I & 0 \\ 0 & V_4^H \end{bmatrix} \\ \hline \cdot & [P_4 \hat{R}_4^H \quad D_4] \end{array} \right].$$

And, finally, we have the complete block upper triangular matrix A_1 :

$$(9) \quad A_1 = \left[\begin{array}{c} A_1^{(3)} \left[\begin{array}{c} [D_1 \hat{Q}_{12} \quad U_1] \begin{bmatrix} \hat{Q}_{22}^{(1)} & 0 \\ V_2^H \hat{Q}_{22}^{(2)} & W_2 \end{bmatrix} \begin{bmatrix} \hat{Q}_{32}^{(1)} & 0 \\ V_3^H \hat{Q}_{32}^{(2)} & W_3 \end{bmatrix} \begin{bmatrix} I & 0 \\ 0 & V_4^H \end{bmatrix} \\ [P_2 \hat{R}_2^H \hat{Q}_{22}^{(1)} + D_2 \hat{Q}_{22}^{(2)} \quad U_2] \begin{bmatrix} \hat{Q}_{32}^{(1)} & 0 \\ V_3^H \hat{Q}_{32}^{(2)} & W_3 \end{bmatrix} \begin{bmatrix} I & 0 \\ 0 & V_4^H \end{bmatrix} \\ [P_3 \hat{R}_3^H \hat{Q}_{32}^{(1)} + D_3 \hat{Q}_{32}^{(2)} \quad U_3] \begin{bmatrix} I & 0 \\ 0 & V_4^H \end{bmatrix} \\ [P_4 \hat{R}_4^H \quad D_4] \end{array} \right] \end{array} \right].$$

For the simplicity of notation in the following steps, we denote the component matrices as follows:

$$\hat{U}_i = [P_i \hat{R}_i^H \hat{Q}_{i2}^{(1)} + D_i \hat{Q}_{i2}^{(2)} \quad U_i],$$

$$\hat{V}_i^H = \begin{bmatrix} \hat{Q}_{i2}^{(1)} \\ V_i^H \hat{Q}_{i2}^{(2)} \end{bmatrix}, \text{ except for } \hat{V}_4^H = \begin{bmatrix} I & 0 \\ 0 & V_4^H \end{bmatrix},$$

$$\hat{W}_i = \begin{bmatrix} \hat{Q}_{i2}^{(1)} & 0 \\ V_i^H \hat{Q}_{i2}^{(2)} & W_i \end{bmatrix},$$

and

$$\hat{D}_i = P_i \hat{R}_i^H \hat{Q}_{i1}^{(1)} + D_i \hat{Q}_{i1}^{(2)}, \text{ except for } \hat{D}_4 = [P_4 \hat{R}_4^H \quad D_4].$$

With these notations,

$$A_1 = A\hat{Q} = \begin{bmatrix} \hat{D}_1 & \hat{U}_1 \hat{V}_2^H & \hat{U}_1 \hat{W}_2 \hat{V}_3^H & \hat{U}_1 \hat{W}_2 \hat{W}_3 \hat{V}_4^H \\ 0 & \hat{D}_2 & \hat{U}_2 \hat{V}_3^H & \hat{U}_2 \hat{W}_3 \hat{V}_4^H \\ 0 & 0 & \hat{D}_3 & \hat{U}_3 \hat{V}_4^H \\ 0 & 0 & 0 & \hat{D}_4 \end{bmatrix}.$$

2.3. Left multiplication—Converting to linearly independent rows.

Step 1 QR decomposition on $[\hat{D}_1 \quad \hat{U}_1]$:

$$\begin{bmatrix} \tilde{Q}_1^{(1)} & \tilde{Q}_1^{(2)} & \tilde{Q}_1^{(3)} \end{bmatrix}^H [\hat{D}_1 \quad \hat{U}_1] = \begin{bmatrix} 0 & 0 \\ \hat{D}_{o1} & \hat{U}_{o1} \\ 0 & Y_2 \end{bmatrix},$$

where $\tilde{Q}_1 = [\tilde{Q}_1^{(1)} \quad \tilde{Q}_1^{(2)} \quad \tilde{Q}_1^{(3)}]$ is the “Q” matrix of the QR decomposition (for later convenience and consistency we have put the bottom row of zeros in the “R” factor on top), and $\hat{D}_{o1}$ and Y_2 have linearly independent rows by construction. (They can be in echelon form.) The result of Step 1 applied to the whole matrix A_1 by left multiplying the elementary transformation $\tilde{Q}_1^H$ is

$$\begin{bmatrix} 0 & 0 & 0 & 0 \\ \hat{D}_{o1} & \hat{U}_{o1}\hat{V}_2^H & \hat{U}_{o1}\hat{W}_2\hat{V}_3^H & \hat{U}_{o1}\hat{W}_2\hat{W}_3\hat{V}_4^H \\ 0 & Y_2\hat{V}_2^H & Y_2\hat{W}_2\hat{V}_3^H & Y_2\hat{W}_2\hat{W}_3\hat{V}_4^H \\ \hline 0 & \hat{D}_2 & \hat{U}_2\hat{V}_3^H & \hat{U}_2\hat{W}_3\hat{V}_4^H \\ 0 & 0 & \hat{D}_3 & \hat{U}_3\hat{V}_4^H \\ \hline 0 & 0 & 0 & \hat{D}_4 \end{bmatrix},$$

and the block row containing Y_2 goes to the next stage while the first two block rows will remain untouched in the next steps.

Step 2 This step consists again in finding a QR factorization,

$$\tilde{Q}_2^H \begin{bmatrix} Y_2\hat{V}_2^H & Y_2\hat{W}_2 \\ \hat{D}_2 & \hat{U}_2 \end{bmatrix} = \begin{bmatrix} 0 & 0 \\ \hat{D}_{o2} & \hat{U}_{o2} \\ 0 & Y_3 \end{bmatrix},$$

in which $\tilde{Q}_2$ has the block decomposition

$$\tilde{Q}_2 = \begin{bmatrix} \tilde{Q}_{21}^{(1)} & \tilde{Q}_{21}^{(2)} & \tilde{Q}_{21}^{(3)} \\ \tilde{Q}_{22}^{(1)} & \tilde{Q}_{22}^{(2)} & \tilde{Q}_{22}^{(3)} \end{bmatrix}.$$

Left multiplying $\tilde{Q}_2^H$ produces a result for the second step:

$$\begin{bmatrix} 0 & 0 & 0 & 0 \\ \hat{D}_{o1} & \hat{U}_{o1}\hat{V}_2^H & \hat{U}_{o1}\hat{W}_2\hat{V}_3^H & \hat{U}_{o1}\hat{W}_2\hat{W}_3\hat{V}_4^H \\ 0 & 0 & 0 & 0 \\ 0 & \hat{D}_{o2} & \hat{U}_{o2}\hat{V}_3^H & \hat{U}_{o2}\hat{W}_3\hat{V}_4^H \\ 0 & 0 & Y_3\hat{V}_3^H & Y_3\hat{W}_3\hat{V}_4^H \\ \hline 0 & 0 & \hat{D}_3 & \hat{U}_3\hat{V}_4^H \\ 0 & 0 & 0 & \hat{D}_4 \end{bmatrix}.$$

Step 3 Again, we perform a QR factorization,

$$\tilde{Q}_3^H \begin{bmatrix} Y_3\hat{V}_3^H & Y_3\hat{W}_3 \\ \hat{D}_3 & \hat{U}_3 \end{bmatrix} = \begin{bmatrix} 0 & 0 \\ \hat{D}_{o3} & \hat{U}_{o3} \\ 0 & Y_4 \end{bmatrix},$$

in which $\tilde{Q}_3$ has the block decomposition

$$\tilde{Q}_2 = \begin{bmatrix} \tilde{Q}_{31}^{(1)} & \tilde{Q}_{31}^{(2)} & \tilde{Q}_{31}^{(3)} \\ \tilde{Q}_{32}^{(1)} & \tilde{Q}_{32}^{(2)} & \tilde{Q}_{32}^{(3)} \end{bmatrix}.$$

Left multiplying $\tilde{Q}_3^H$ we obtain

$$\begin{bmatrix} 0 & 0 & 0 & 0 \\ \hat{D}_{o1} & \hat{U}_{o1}\hat{V}_2^H & \hat{U}_{o1}\hat{W}_2\hat{V}_3^H & \hat{U}_{o1}\hat{W}_2\hat{W}_3\hat{V}_4^H \\ 0 & 0 & 0 & 0 \\ 0 & \hat{D}_{o2} & \hat{U}_{o2}\hat{V}_3^H & \hat{U}_{o2}\hat{W}_3\hat{V}_4^H \\ 0 & 0 & 0 & 0 \\ 0 & 0 & \hat{D}_{o3} & \hat{U}_{o3}\hat{V}_4^H \\ 0 & 0 & 0 & Y_4\hat{V}_4^H \\ \hline 0 & 0 & 0 & \hat{D}_4 \end{bmatrix}.$$

4 In this final step, we QR factorize $\begin{bmatrix} Y_4\hat{V}_4^H \\ \hat{D}_4 \end{bmatrix}$:

$$\tilde{Q}_4^H \begin{bmatrix} Y_4\hat{V}_4^H \\ \hat{D}_4 \end{bmatrix} = \begin{bmatrix} 0 \\ \hat{D}_{o4} \end{bmatrix},$$

in which the last block consists of linearly independent rows to yield the final result:

$$\begin{bmatrix} 0 & 0 & 0 & 0 \\ \hat{D}_{o1} & \hat{U}_{o1}\hat{V}_2^H & \hat{U}_{o1}\hat{W}_2\hat{V}_3^H & \hat{U}_{o1}\hat{W}_2\hat{W}_3\hat{V}_4^H \\ 0 & 0 & 0 & 0 \\ 0 & \hat{D}_{o2} & \hat{U}_{o2}\hat{V}_3^H & \hat{U}_{o2}\hat{W}_3\hat{V}_4^H \\ 0 & 0 & 0 & 0 \\ 0 & 0 & \hat{D}_{o3} & \hat{U}_{o3}\hat{V}_4^H \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & \hat{D}_{o4} \end{bmatrix}.$$

If we left multiply the $\tilde{Q}_i^H$'s to b also, we will have

$$\tilde{Q}_4^H \tilde{Q}_3^H \tilde{Q}_2^H \tilde{Q}_1^H \begin{bmatrix} b_1 \\ b_2 \\ b_3 \\ b_4 \end{bmatrix} = \begin{bmatrix} b_{o1}^{(1)} \\ b_{o1}^{(2)} \\ \hline b_{o2}^{(1)} \\ b_{o2}^{(2)} \\ \hline b_{o3}^{(1)} \\ b_{o3}^{(2)} \\ \hline b_{o4}^{(1)} \\ b_{o4}^{(2)} \end{bmatrix}.$$

Hence, the quantities $b_{oi}^{(1)}$'s remain to produce the Moore–Penrose square error:

$$\epsilon_{MPI}^2 = \|b_{o1}^{(1)}\|^2 + \|b_{o3}^{(1)}\|^2 + \|b_{o3}^{(1)}\|^2 + \|b_{o4}^{(1)}\|^2,$$

TABLE 2
The dimensions of the component matrices used in the reduction procedure.

Matrix	D_i	U_i	V_i	W_i	P_i	Q_i	R_i
Dimensions	$m_i \times n_i$	$m_i \times k_{i+1}$	$n_i \times k_i$	$k_i \times k_{i+1}$	$m_i \times l_i$	$n_i \times l_{i+1}$	$l_{i+1} \times l_i$
Matrix	$\hat{R}_i$	$\hat{Q}_{i1}^{(1)}$		$\hat{Q}_{i1}^{(2)}$		$\hat{Q}_{i2}^{(1)}$	$\hat{Q}_{i2}^{(2)}$
Dimensions	$l_i \times l_i$	$l_i \times (n_i + l_i - l_{i+1})$		$n_i \times (n_i + l_i - l_{i+1})$		$l_i \times l_{i+1}$	$n_i \times l_{i+1}$

while Y_i 's will now be uniquely determined by solving a unilateral semiseparable system given by

$$\begin{bmatrix} \hat{D}_{o1} & \hat{U}_{o1}\hat{V}_2^H & \hat{U}_{o1}\hat{W}_2\hat{V}_3^H & \hat{U}_{o1}\hat{W}_2\hat{W}_3\hat{V}_4^H \\ 0 & \hat{D}_{o2} & \hat{U}_{o2}\hat{V}_3^H & \hat{U}_{o2}\hat{W}_3\hat{V}_4^H \\ 0 & 0 & \hat{D}_{o3} & \hat{U}_{o3}\hat{V}_4^H \\ 0 & 0 & 0 & \hat{D}_{o4} \end{bmatrix} \begin{bmatrix} Y_1 \\ Y_2 \\ Y_3 \\ Y_4 \end{bmatrix} = \begin{bmatrix} b_{o1}^{(2)} \\ b_{o2}^{(2)} \\ b_{o3}^{(2)} \\ b_{o4}^{(2)} \end{bmatrix},$$

and this system can now be solved by the direct method of [6].

It should now be clear that the two steps of the algorithm just described can be done in a top-down, left-right fashion with “lazy” execution, and only the time-varying state space description has to be handled, of course. In fact, one can say, alternatively, that the state space descriptions are treated in increasing index order.

2.4. Flop counts. We list the dimensions of each component matrix in Table 2. We have $\sum_{i=1}^K m_i = M$ and $\sum_{i=1}^K n_i = N$.

In each step of the one-pass, top-down algorithm, the following tasks must be performed, and the flops required for each task are listed:

1. Multiply a triangular matrix to a general matrix $\hat{R}_i R_i^H$: $l_i^2 l_{i+1}$ flops required.
2. QR factorize a general matrix $\begin{bmatrix} \hat{R}_i R_i^H \\ \hat{Q}_i \end{bmatrix}$ via Householder triangularization: $2l_{i+1}^2(l_i + n_i) - \frac{2}{3}l_{i+1}^3$ flops required.
3. Left multiply a triangular matrix to a general matrix $P_i \hat{R}_i$: $m_i l_i^2$ flops.
4. Right multiply $\hat{Q}_i$ generated in the above step to

$$\begin{bmatrix} I & 0 \\ 0 & V_i^H \\ P_i & D_i \end{bmatrix},$$

using the elementary reflectors: $[4(l_i + n_i)l_{i+1} - 2l_{i+1}^2](l_i + k_i + m_i)$ flops.

5. Left multiply a triangular matrix to a general matrix $Y_i[\hat{V}_i^H \hat{W}_i]$: $(l_i + k_i)^2(l_i + n_i + k_{i+1})$ flops.
6. QR factorize a general matrix $\begin{bmatrix} Y_i \hat{V}_i^H & \hat{W}_i \\ \hat{D}_i & \hat{U}_i \end{bmatrix}$ via Householder triangularization: $2(l_i + n_i + k_{i+1})^2(l_i + k_i + m_i) - \frac{2}{3}(l_i + n_i + k_{i+1})^3$ flops.

Assuming $m_i = m = M/K$, $n_i = n/K$, and $l_i = k_i = p$ for all $i = 1, 2, \dots, K$,¹ we have the approximate total flops

$$(10) \quad 25p^3K + 22p^2N + 11p^2M + 12pM/K + 2N^2M/K - \frac{2}{3}N^2/K^2.$$

We now approximately estimate an optimal number of blocks K , which gives an upper bound of the minimal flops needed, under two conditions:

¹There is an underlying assumption that M and N have a common divider K with moderate size.

1. $M \approx N$. (10) is approximated by

$$f_1(K) = 25p^3K + 33p^2M + 12pM^2/K + \frac{4}{3}M^3/K^2.$$

Take the derivative with respect to K and set the derivative to 0:

$$25p^3 - 12pM^2/K^2 - \frac{8}{3}M^3/K^3 = 0.$$

$$f_1^{\text{opt}} < (33 + \frac{25}{9} + 12\sqrt{3})p^2M \approx 56.56p^2M, \text{ with } K^{\text{opt}} \approx \frac{2\sqrt{3}M}{5p} \approx 0.6928M/p.$$

2. $\alpha = \frac{N}{M} \ll 1$. Then (10) is approximated by

$$f_2(K) = 25p^3K + 11p^2M + 12\alpha pM^2/K + 2\alpha^2M^3/K^2.$$

$$f_2^{\text{opt}} < (11 + 12\sqrt[3]{25\alpha})p^2M, \text{ with } K^{\text{opt}} \approx \sqrt[3]{\frac{\alpha^2}{25}} \frac{M}{p} \approx 0.3420\sqrt[3]{\alpha^2}M/p.$$

2.5. Experimental run-times. We have written a Fortran 90 program to implement our fast solver for the least squares problem. Our experiments were run on a Sun Fire v880 Server with four 900 MHz UltraSPARC-III processors, 8 gigabytes of memory, and the Solaris 2.8 operating system. The code was compiled using Sun WorkShop Fortran 95 compiler and the Sun Performance Library, with options “-fast -dalign.” The machine precision is $\epsilon = 2^{-54} \approx 1.1 \times 10^{-16}$. We compared our method with the LAPACK routine DGELS [3], which solves the problem as a general least squares problem.

We used the results in [26] to measure the backward errors in the solutions produced by both methods. Let $\hat{x}$ be any approximate solution to the least squares problem; and let $A = Q \begin{pmatrix} D \\ 0 \end{pmatrix} W^T$ be the SVD of A . Rewrite

$$b = Q \begin{pmatrix} b_1 \\ b_2 \end{pmatrix} \quad \text{and} \quad r = b - A\hat{x} = Q \begin{pmatrix} r_1 \\ r_2 \end{pmatrix}.$$

We compute the backward error in $\hat{x}$ as

$$\mathcal{E} \equiv \begin{cases} \frac{\|D r_1\|_2}{\|r\|_2} & \text{if } \hat{x} = 0, \\ \min(\eta, \tilde{\sigma}) & \text{otherwise,} \end{cases}$$

where $\eta = \frac{\|r\|_2}{\|\hat{x}\|_2}$, $\gamma = \|A\hat{x} - b\|_2$, and

$$\tilde{\sigma} = \sqrt{\frac{r_1^T D^2 (D^2 + \eta^2 I)^{-1} r_1}{\gamma^2 / \eta^2 + \eta^2 r_1^T (D^2 + \eta^2 I)^{-2} r_1}}.$$

It is known that this backward error is within a factor of 2 from the smallest possible backward error [26].

In our experiments, we let the number of blocks increase but keep the block sizes and off-diagonal block ranks constant. We tested the following types of matrices:

- $\bullet, \bullet$: Well-conditioned random matrices with $m = 30$, $n = 20$, and $k = l = 5$.

- , II: Well-conditioned random matrices with $m = 30$, $n = 20$, and $k = l = 10$.
- , III: Ill-conditioned random matrices with $m = 30$, $n = 20$, and $k = l = 5$.
- , IV: Ill-conditioned random matrices with $m = 30$, $n = 20$, and $k = l = 10$.

Table 3 summarizes our numerical experiments. Our scheme (**NEW**) was consistently and significantly faster than LAPACK (**DGELS**), sometimes by factors exceeding 5000. On the other hand, it produced backward errors that are as small as those of **DGELS**.

3. Superfast solution of $AX = B$. We now consider the fast solution of the linear system of equations $AX = B$, where A and B are given in SSS form, and we desire to find the SSS form of X . One possible approach to the problem is to first compute explicitly the SSS form of U , L , and V in the decomposition $A = ULV^H$ and then use the fast multiplication algorithm for matrices in SSS form. Such an approach has already been described in [16]. Here we show how to extend the implicit decomposition ideas [6] in the same direction.

We will assume that A , X , and B are conformally partitioned. It is useful to allow B and X to be nonsquare matrices. To facilitate this description we will assume that the row and column partitions of the matrix A are indexed from 1 to n . So the SSS representation of A is given by $\{D_i(A)\}_{i=1}^n$, $\{U_i(A)\}_{i=1}^n$, and so on. We will assume that the row partitions of both X and B are also indexed from 1 to n . However, we will assume that the column partitions of X and B are indexed from m to r . Hence the SSS representation for B , for example, will be given by $\{D_i(B)\}_{i=1}^n$, $\{U_i(B)\}_{i=1}^n$, $\{W_i(B)\}_{i=1}^r$, $\{V_i(B)\}_{i=1}^r$, $\{P_i(B)\}_{i=1}^n$, $\{R_i(B)\}_{i=m}^n$, and $\{Q_i(B)\}_{i=m}^n$. Similar considerations hold for X . Throughout this presentation we will pretend that $m \leq 1 \leq n \leq r$. However, these assumptions can be removed.

To ease the presentation we will assume that the initial equation $AX = B$ is modified to read

$$AX = B - \begin{pmatrix} 0 \\ P_2(A) \\ P_3(A)R_2(A) \\ \vdots \\ P_n(A)R_{n-1}(A) \cdots R_2(A) \end{pmatrix} \begin{pmatrix} Q_m(B)R_{m+1}^H(B) \cdots R_0^H(B)P_1^H(\tau) \\ \vdots \\ Q_0(B)P_1^H(\tau) \\ D_1^H(\tau) \\ V_2(B)U_1^H(\tau) \\ \vdots \\ V_r(B)W_{r-1}^H(B) \cdots W_2^H(B)U_1^H(\tau) \end{pmatrix}^H, \quad (11)$$

where $D_1(\tau) = 0$, $P_1(\tau) = 0$, and $U_1(\tau) = 0$. The reason for this peculiar form will become clear shortly.

This fast solver is essentially of the same form as the fast solver presented in [6]. This fast solver can also be presented in a recursive fashion, which we proceed to do.

3.1. Case of $n > 1$ and $k_1(A) < m_1$: Elimination. First we perform orthogonal eliminations by computing QL and LQ factorizations

$$U_1(A) = q_1 \begin{pmatrix} 0 \\ \hat{U}_1(A) \end{pmatrix} \begin{pmatrix} m_1 - k_1(A) \\ k_1(A) \end{pmatrix}$$

TABLE 3
Execution times and scaled backward errors.

		Number of blocks				
		10	20	40	80	160
Execution time (seconds)						
NEW	TYPE I	1.0×10^{-2}	1.6×10^{-2}	2.7×10^{-2}	5.3×10^{-2}	1.2×10^{-1}
	TYPE II	1.5×10^{-2}	2.3×10^{-2}	4.5×10^{-2}	9.9×10^{-2}	2.2×10^{-1}
	TYPE III	1.1×10^{-2}	1.7×10^{-2}	3.2×10^{-2}	5.3×10^{-2}	9.4×10^{-2}
	TYPE IV	1.4×10^{-2}	2.4×10^{-2}	4.4×10^{-2}	8.6×10^{-2}	1.8×10^{-1}
DGELS	TYPE I	3.8×10^{-2}	4.8×10^{-1}	2.3×10^0	7.2×10^1	5.7×10^2
	TYPE II	5.3×10^{-2}	3.3×10^{-1}	2.4×10^0	5.2×10^1	6.2×10^2
	TYPE III	3.9×10^{-2}	2.5×10^{-1}	2.5×10^0	4.4×10^1	4.9×10^2
	TYPE IV	3.8×10^{-2}	2.8×10^{-1}	2.1×10^0	3.9×10^1	5.2×10^2
Scaled backward errors $\frac{\mathcal{E}}{\ A\ _2 \epsilon}$						
NEW	TYPE I	4.0×10^{-2}	4.3×10^{-2}	5.4×10^{-2}	2.8×10^{-2}	1.6×10^{-2}
	TYPE II	3.6×10^{-2}	5.1×10^{-2}	4.1×10^{-2}	2.9×10^{-2}	3.1×10^{-2}
	TYPE III	2.4×10^{-2}	5.4×10^{-2}	2.4×10^{-2}	1.2×10^{-2}	2.4×10^{-2}
	TYPE IV	5.5×10^{-2}	4.8×10^{-2}	3.3×10^{-2}	3.1×10^{-2}	1.7×10^{-2}
DGELS	TYPE I	5.3×10^{-2}	3.4×10^{-2}	2.7×10^{-2}	1.7×10^{-2}	1.3×10^{-2}
	TYPE II	5.3×10^{-2}	3.3×10^{-2}	2.9×10^{-2}	2.6×10^{-2}	1.8×10^{-2}
	TYPE III	3.9×10^{-2}	2.2×10^{-2}	2.3×10^{-2}	1.7×10^{-2}	1.2×10^{-2}
	TYPE IV	5.1×10^{-2}	3.3×10^{-2}	2.3×10^{-2}	2.0×10^{-2}	1.3×10^{-2}

and

$$(q_1^H D_1(A)) = \frac{m_1 - k_1(A)}{k_1(A)} \begin{pmatrix} m_1 - k_1(A) & k_1(A) \\ D_{11}(A) & 0 \\ D_{21}(A) & D_{22}(A) \end{pmatrix} w_1,$$

where q_1 and w_1 are unitary matrices. We now need to apply q_1^H to the first m_1 rows of (11). Nothing needs to be done for A , X , and the second term involving τ on the right-hand side. Applying q_1^H to the first m_1 rows of B we modify $P_1(B)$, $D_1(B)$, and $U_1(B)$ as follows:

$$q_1^H P_1(B) = \frac{m_1 - k_1(A)}{k_1(A)} \begin{pmatrix} \tilde{P}_1(B) \\ \hat{P}_1(B) \end{pmatrix},$$

$$q_1^H D_1(B) = \frac{m_1 - k_1(A)}{k_1(A)} \begin{pmatrix} \tilde{D}_1(B) \\ \tilde{D}_2(B) \end{pmatrix},$$

$$q_1^H U_1(B) = \frac{m_1 - k_1(A)}{k_1(A)} \begin{pmatrix} \tilde{U}_1(B) \\ \hat{U}_1(B) \end{pmatrix}.$$

Next we need to apply w_1 to the first m_1 rows of X . Of course this is a purely formal process since the first m_1 rows of X are unknown at this stage. Here are the quantities that are modified:

$$w_1 P_1(X) = \frac{m_1 - k_1(A)}{k_1(A)} \begin{pmatrix} \tilde{P}_1(X) & 0 \\ \hat{P}_{11}(X) & \hat{P}_{12}(X) \end{pmatrix},$$

$$w_1 D_1(X) = \frac{m_1 - k_1(A)}{k_1(A)} \begin{pmatrix} \tilde{D}_1(X) \\ \hat{D}_1(X) \end{pmatrix},$$

$$w_1 U_1(X) = \frac{m_1 - k_1(A)}{k_1(A)} \begin{pmatrix} \tilde{U}_1(X) & 0 \\ \hat{U}_{11}(X) & \hat{U}_{12}(X) \end{pmatrix}.$$

The zeros in the modified $w_1 U_1(X)$ and $w P_1(X)$ are not a restriction since the number of columns in either of these matrices has not been fixed yet.

We also need to apply w_1^H to the first m_1 columns of the coefficient matrix. Since it has already been applied to $D_1(A)$, we need only to compute

$$w_1 Q_1(A) = \frac{m_1 - k_1(A)}{k_1(A)} \begin{pmatrix} \tilde{Q}_1(A) \\ \hat{Q}_1(A) \end{pmatrix}.$$

We now observe that the first $m_1 - k_1(A)$ equations of the transformed system

are of the form

$$D_{11}(A) \begin{pmatrix} Q_m^H(X) R_{m+1}^H(X) \cdots R_0^H(X) (\tilde{P}_1(X) & 0)^H \\ \vdots \\ Q_0^H(X) (\tilde{P}_1(X) & 0)^H \\ \tilde{D}_1^H(X) \\ V_2(X) (\tilde{U}_1(X) & 0)^H \\ \vdots \\ V_r(X) W_{r-1}^H(X) \cdots W_2^H(X) (\tilde{U}_1(X) & 0)^H \end{pmatrix}^H$$

$$= \begin{pmatrix} Q_m^H(B) R_{m+1}^H(B) \cdots R_0^H(B) \tilde{P}_1^H(B) \\ \vdots \\ Q_0^H(B) \tilde{P}_1^H(B) \\ \tilde{D}_1^H(B) \\ V_2(B) \tilde{U}_1^H(B) \\ \vdots \\ V_r(B) W_{r-1}^H(B) \cdots W_2^H(B) \tilde{U}_1^H(B) \end{pmatrix}^H.$$

Solving this we obtain

$$\begin{aligned} \tilde{D}_1(X) &= D_{11}^{-1}(A) \tilde{D}_1(B), \\ \tilde{P}_1(X) &= D_{11}^{-1}(A) \tilde{P}_1(B), \\ Q_i(X) &= (Q_i(B) \quad \vec{Q}_i), \\ R_i(X) &= \begin{pmatrix} R_i(B) & 0 \\ (\vec{R}_i)_1 & (\vec{R}_i)_2 \end{pmatrix}, \\ \tilde{U}_1(X) &= D_{11}^{-1}(A) \tilde{U}_1(B), \\ V_i(X) &= (V_i(B) \quad \vec{V}_i), \\ W_i(X) &= \begin{pmatrix} W_i(B) & 0 \\ (\vec{W}_i)_1 & (\vec{W}_i)_2 \end{pmatrix}. \end{aligned}$$

Notice that all $Q_i(X)$, $V_i(X)$, $R_i(X)$, and $W_i(X)$ have constrained forms for their solutions for all i . These forms will be maintained consistently throughout the algorithm.

We now need to subtract from the right-hand side the first $m_1 - k_1(A)$ columns of the coefficient matrix multiplied by the first $m_1 - k_1(A)$ rows of the transformed unknowns. We first subtract $D_{21}(A)$ times the first $m_1 - k_1(A)$ rows of the unknowns from the corresponding rows of B . We observe that it leads to the following changes:

$$\begin{aligned} \hat{D}_1(B) &= \tilde{D}_2(B) - D_{21}(A) \tilde{D}_1(X), \\ \hat{U}_1(B) &= \tilde{U}_2(B) - D_{21}(A) \tilde{U}_1(X), \\ \hat{P}_1(B) &= \tilde{P}_2(B) - D_{21}(A) \tilde{P}_1(X). \end{aligned}$$

To subtract the remaining rows of the first $m_1 - k_1(A)$ columns of the coefficient matrix from the right-hand side, we observe that it can be merged with the τ terms as follows:

$$\begin{aligned} \hat{D}_1(\tau) &= D_1(\tau) + \tilde{Q}_1^H(A) \tilde{D}_1(X), \\ \hat{P}_1(\tau) &= P_1(\tau) + \tilde{Q}_1^H(A) \tilde{P}_1(X), \\ \hat{U}_1(\tau) &= U_1(\tau) + \tilde{Q}_1^H(A) \tilde{U}_1(X). \end{aligned}$$

If we now discard the first $m_1 - k_1(A)$ rows and columns, we are left with a set of equations that are identical in structure to the system (11) that we started with. In particular we replace in the SSS form for A , X , and B all terms with their corresponding hatted forms. The resulting hatted system $\hat{A}\hat{X} = \hat{B} - (\tau \text{ terms})$ looks exactly like the (11) we started with and can be solved recursively for $\hat{X}$. Once we have $\hat{X}$ we need to recover X . The right formulas are

$$\begin{aligned} D_1(X) &= w_1^H \begin{pmatrix} \tilde{D}_1(X) \\ \hat{D}_1(X) \end{pmatrix}, \\ \hat{U}_1(X) &= (\hat{U}_{11}(X) \quad \hat{U}_{12}(X)), \\ U_1(X) &= w_1^H \begin{pmatrix} \tilde{U}_1(X) & 0 \\ \hat{U}_{11}(X) & \hat{U}_{12}(X) \end{pmatrix}, \\ \hat{P}_1(X) &= (\hat{P}_{11}(X) \quad \hat{P}_{12}(X)), \\ P_1(X) &= w_1^H \begin{pmatrix} \tilde{P}_1(X) & 0 \\ \hat{P}_{11}(X) & \hat{P}_{12}(X) \end{pmatrix}. \end{aligned}$$

These formulas can be verified only by looking at the next two cases.

3.2. Case for $n > 1$ and $k_1(A) \geq m_1$: Merge. The second possibility is that $k_1(A)$ is not large enough to permit efficient elimination. In this case we merge the first two block rows of the equation. We also merge the first two block columns of A . However, for X and B we do not merge any block columns. Rather we move the diagonal block over by one position to the right. To merge the first two block rows and columns of A , we can use the formulas in section 3.6.

Next we need to merge the first two block rows of B and move the diagonal block one position over to the right:

$$\begin{aligned} \tilde{D}_1(B) &= \begin{pmatrix} U_1(B)V_2^H(B) \\ D_2(B) \end{pmatrix}, \\ \hat{D}_i(B) &= D_{i+1}(B), \quad i = 2, \dots, n-1, \\ \begin{pmatrix} \tilde{P}_1(B) \\ \hat{R}_1(B) \\ \tilde{P}_1(\tau) \end{pmatrix} (\hat{R}_0(B) \quad \hat{Q}_0^H(B)) &= \begin{pmatrix} \begin{pmatrix} P_1(B) \\ P_2(B)R_1(B) \\ R_2(B)R_1(B) \\ P_1(\tau) \end{pmatrix} & \begin{pmatrix} D_1(B) \\ P_2(B)Q_1^H(B) \\ R_2(B)Q_1^H(B) \\ D_1(\tau) \end{pmatrix} \end{pmatrix}, \\ \hat{P}_i(B) &= P_{i+1}(B), \quad i = 2, \dots, n-1, \\ \hat{R}_i(B) &= R_{i+1}(B), \quad i = m-1, \dots, -1, 2, \dots, r-1, \\ \hat{Q}_i(B) &= Q_{i+1}(B), \quad i = m-1, \dots, -1, 1, \dots, r-1, \\ \tilde{U}_1(B) &= \begin{pmatrix} U_1(B)W_2(B) \\ U_2(B) \end{pmatrix}, \\ \hat{U}_i(B) &= U_{i+1}(B), \quad i = 2, \dots, n-1, \\ \hat{W}_i(B) &= W_{i+1}(B), \quad i = 2, \dots, r-1, \\ \hat{V}_i(B) &= V_{i+1}(B), \quad i = 2, \dots, r-1. \end{aligned}$$

The implied factorization in the above formulas must be carried out as follows:

$$\tilde{P}_1(B) = \begin{pmatrix} 0 & I & 0 \\ P_2(B) & 0 & 0 \end{pmatrix},$$

$$\begin{aligned}
\hat{R}_1(B) &= (R_2(B) \quad 0 \quad 0), \\
\tilde{P}_1(\tau) &= (0 \quad 0 \quad I), \\
\hat{R}_0(B) &= \begin{pmatrix} R_1(B) \\ P_1(B) \\ P_1(\tau) \end{pmatrix}, \\
\hat{Q}_0(B) &= (Q_1(B) \quad D_1^H(B) \quad D_1^H(\tau)).
\end{aligned}$$

The formal expressions for X are similar to B , so we skip them.

Some quantities in the SSS representation for B still do not wear hats since we still need to subtract the second block row of the τ terms from B . To do that we must first move the diagonal τ block one position over to the right. Observe that $\tilde{P}_1(\tau)$ has already been defined. The remaining changes are

$$\begin{aligned}
\tilde{D}_1(\tau) &= U_1(\tau)V_2^H(B), \\
\tilde{U}_1(\tau) &= U_1(\tau)W_2(B).
\end{aligned}$$

Now we can subtract the second block row of the τ terms from the new first block of B . The corresponding changes to B are

$$\begin{aligned}
\hat{P}_1(B) &= \tilde{P}_1(B) - \begin{pmatrix} 0 \\ P_2(A) \end{pmatrix} \tilde{P}_1(\tau), \\
\hat{U}_1(B) &= \tilde{U}_1(B) - \begin{pmatrix} 0 \\ P_2(A) \end{pmatrix} \tilde{U}_1(\tau), \\
\hat{D}_1(B) &= \tilde{D}_1(B) - \begin{pmatrix} 0 \\ P_2(A) \end{pmatrix} \tilde{D}_1(\tau).
\end{aligned}$$

We must now return the τ terms to canonical form with the first two block rows being merged to zero. The changes are

$$\begin{aligned}
\hat{P}_1(\tau) &= R_2(A)\tilde{P}_1(\tau), \\
\hat{D}_1(\tau) &= R_2(A)\tilde{D}_1(\tau), \\
\hat{U}_1(\tau) &= R_2(A)\tilde{U}_1(\tau).
\end{aligned}$$

Now the hatted sequences represent an SSS system with $n - 1$ block rows. This can be solved recursively for the hatted SSS representation for X . From that we must recover the original SSS representation for X involving n row partitions. To do that we observe that we need to split the first block row of the hatted representation and shift the first diagonal block one position to the left. We begin by first splitting the first block row:

$$\begin{aligned}
\hat{D}_1(X) &= \begin{pmatrix} \tilde{D}_1(X) \\ D_2(X) \end{pmatrix}, \\
\hat{U}_1(X) &= \begin{pmatrix} \tilde{U}_1(X) \\ U_2(X) \end{pmatrix}, \\
\hat{P}_1(X) &= \begin{pmatrix} \tilde{P}_1(X) \\ P_2(X) \end{pmatrix}.
\end{aligned}$$

The rest of the changes to X are

$$D_1(X) = \tilde{P}_1(X)\hat{Q}_0^H(X),$$

$$\begin{aligned}
P_1(X) &= \tilde{P}_1(X) \hat{R}_0(X), \\
P_i(X) &= \hat{P}_{i-1}(X), \quad i = 3, \dots, n, \\
R_i(X) &= \hat{R}_{i-1}(X), \quad i = m, \dots, n, \\
Q_i(X) &= \hat{Q}_{i-1}(X), \quad i = m, \dots, n, \\
U_1(X) \begin{pmatrix} W_2(X) & V_2^H(X) \end{pmatrix} &= \begin{pmatrix} \tilde{U}_1(X) & \tilde{D}_1(X) \end{pmatrix}, \\
W_i(X) &= \hat{W}_{i-1}(X), \quad i = 3, \dots, n, \\
V_i(X) &= \hat{V}_{i-1}(X), \quad i = 3, \dots, n,
\end{aligned}$$

where the implied factorization must be solved as follows:

$$\begin{aligned}
U_1(X) &= \begin{pmatrix} 0 & I \end{pmatrix}, \\
\tilde{U}_1(X) &= \begin{pmatrix} \tilde{U}_{11}(X) & \tilde{U}_{12}(X) \end{pmatrix}, \\
W_2(X) &= \begin{pmatrix} W_2(B) & 0 \\ \tilde{U}_{11}(X) & \tilde{U}_{12}(X) \end{pmatrix}, \\
V_2(X) &= \begin{pmatrix} V_2(B) & \tilde{D}_1^H(X) \end{pmatrix}.
\end{aligned}$$

3.3. Case for $n = 1$: Base. In this case the equations can be solved directly to determine the SSS form for X as follows:

$$\begin{aligned}
D_1(X) &= D_1^{-1}(A) D_1(B), \\
P_1(X) &= D_1^{-1}(A) P_1(B), \\
R_i(X) &= R_i(B), \\
Q_i(X) &= Q_i(B), \\
U_1(X) &= D_1^{-1}(A) U_1(B), \\
W_i(X) &= W_i(B), \\
V_i(X) &= V_i(B).
\end{aligned}$$

Note that the τ terms are just zeros.

3.4. Experimental run-times. We now report on the CPU run-times of the superfast solver on a PowerBook 400 MHz G4 computer with 1 MB of L2 cache, 1 GB of RAM, and a 100 MHz bus. The CPU run-times in seconds are reported in Table 4. We assume that we are solving systems of the form $AX = B$, with B a square matrix. Tests were done on systems where the number of columns of B ranged from 256 to 32,768. This is the size column in Table 4. For the matrix A we chose SSS representations such that $m_i(A) = l_i(A) = k_i(A)$, and $m_i(A)$ was either 16 or 32. For the SSS representation of B we chose $l_i(B) = k_i(B)$, and $k_i(B)$ was either 1 or 4. Of course we chose $m_i(B) = m_i(A)$.

For comparison we also report in Table 4 the CPU run-times of the fast solver of [6] for the same system of equations. Of course the fast solver does not exploit the SSS representation of the right-hand side matrix B , so it is just passed a standard dense matrix representation.

As can be seen the fast solver behaves like an $O(n^2)$ algorithm, where n is the order of the system. While this is an order of magnitude better than the standard Gaussian elimination solvers, it is an order of magnitude slower than the superfast solver. We also observe that the superfast solver is reasonably insensitive to the ranks of the off-diagonal blocks of B .

TABLE 4

CPU run-times in seconds for the superfast solver. Numbers prefixed with an asterisk denote estimated run-times.

		Size							
$m_i(A)/l_i(A)/k_i(A)$	$k_i(B)/l_i(B)$	256	512	1024	2048	4096	8192	16,384	32,768
16	1	0.16	0.30	0.60	1.24	2.52	5.52	11.10	24.22
16	4	0.17	0.32	0.64	1.30	2.69	5.57	11.68	25.42
32	1	0.35	0.72	1.56	3.27	6.74	14.13	30.21	69.97
32	4	0.35	0.75	1.63	3.33	6.93	14.41	30.99	70.88
16	fast solver	0.33	1.92	7.87	31.19	136.24	544.46	*2.2E3	*8.7E3
32	fast solver	0.49	2.39	9.79	38.65	163.36	659.58	*2.6E3	*1.1E4

TABLE 5

One-norm normalized residual errors $\|AX - B\|_1/(\epsilon_{mach}(\|A\|_1\|X\|_1 + \|B\|_1))$ of the superfast solver in double precision for stable SSS representations $\|W_i(A)\|_1 = \|R_i(A)\|_1 = 1 = \|W_i(B)\|_1 = \|R_i(B)\|_1$. Entries much larger than one indicate potential lack of stability.

		Size				
$m_i(A)/l_i(A)/k_i(A)$	$k_i(B)/l_i(B)$	256	512	1024	2048	4096
16	1	1.67	1.07	1.56	3.67	3.43
16	4	0.87	3.66	4.10	3.49	3.01
32	1	1.10	1.04	7.56	6.94	3.82
32	4	1.78	2.96	5.30	9.27	7.83
64	1	0.64	1.21	1.59	5.43	9.25
64	4	0.60	0.89	2.11	3.80	4.30
128	1	0.36	0.58	1.12	1.65	5.32
128	4	0.36	0.60	2.06	2.00	9.45

3.5. Stability. An unstable superfast solver is not very useful. However, even the standard slow solvers are not technically backward stable when there are multiple right-hand sides. For that matter, there is no proof that even ordinary matrix-matrix multiplication is backward stable [23]. Numerical analysts generally attribute this phenomenon to an insufficient number of degrees of freedom in the problem for attaching the backward error when there is more than one right-hand side. Hence there is no simple way to measure how stable such solvers are. What we have found useful is to use the standard normalized measure of backward error [23] for the single right-hand side case, with a straightforward modification:

$$\frac{\|AX - B\|_1}{\epsilon_{mach}(\|A\|_1\|X\|_1 + \|B\|_1)}.$$

This is not equivalent to measuring the backward error column by column. In fact it is weaker. The justification is that it can be viewed as a normalized residual.

In Table 5 we present the normalized residual error for a series of random experiments where we used the superfast solver to solve systems of the form $AX = B$, where both A and B were chosen to be random stable SSS forms. As can be seen from the table, the superfast solver seems to behave well from a numerical point of view. However, a more thorough error analysis is needed. This is outside the scope of this paper, so it will be reported separately.

3.6. Merging and splitting SSS blocks. In the presentation of the superfast solver for $AX = B$, we assumed that A and B had SSS representations that were conformally partitioned. If that is not the case, the situation can be remedied by

suitable merging and/or splitting partitions in A and B . For that purpose we present some simple formulas in this section.

We first consider block merging. In the following example, we have merged the second and third blocks in a 4×4 SSS matrix into one single block:

$$A = \begin{pmatrix} D_1 & U_1 V_2^H & U_1 W_2 V_3^H & U_1 W_2 W_3 V_4^H \\ P_2 Q_1^H & D_2 & U_2 V_3^H & U_2 W_3 V_4^H \\ P_3 R_2 Q_1^H & P_3 Q_2^H & D_3 & U_3 V_4^H \\ P_4 R_3 R_2 Q_1^H & P_4 R_3 Q_2^H & P_4 Q_3^H & D_4 \end{pmatrix} \\ = \begin{pmatrix} D_1 & U_1 (V_2^H & W_2 V_3^H) & U_1 (W_2 W_3) V_4^H \\ \begin{pmatrix} P_2 \\ P_3 R_2 \end{pmatrix} Q_1^H & \begin{pmatrix} D_2 & U_2 V_3^H \\ P_3 Q_2^H & D_3 \end{pmatrix} & \begin{pmatrix} U_2 W_3 \\ U_3 \end{pmatrix} V_4^H \\ P_4 (R_3 R_2) Q_1^H & P_4 (R_3 Q_2^H & Q_3^H) & D_4 \end{pmatrix}.$$

In general, to merge blocks i and $i+1$ in a given SSS representation, we keep all other blocks unchanged and use the following matrices to represent the new i th block:

$$D_i^{\text{new}} = \begin{pmatrix} D_i & U_i V_{i+1}^H \\ P_{i+1} Q_i^H & D_{i+1} \end{pmatrix}, \quad U_i^{\text{new}} = \begin{pmatrix} U_i W_{i+1} \\ U_{i+1} \end{pmatrix}, \quad P_i^{\text{new}} = \begin{pmatrix} P_i \\ P_{i+1} R_i \end{pmatrix}, \\ (V_i^{\text{new}})^H = (V_i^H \quad W_i V_{i+1}^H), \quad W_i^{\text{new}} = W_i W_{i+1}, \\ R_i^{\text{new}} = R_{i+1} R_i, \quad (Q_i^{\text{new}})^H = (R_{i+1} Q_i^H \quad Q_{i+1}^H).$$

Now we consider the issue of splitting one block into two. To split block i into two blocks for a given SSS representation, we can simply keep all other blocks unchanged and use the above merging formulas reversely to get the equations for the new i and $i+1$ blocks:

$$\begin{pmatrix} D_i^{\text{new}} & U_i^{\text{new}} (V_{i+1}^{\text{new}})^H \\ P_{i+1}^{\text{new}} (Q_i^{\text{new}})^H & D_{i+1}^{\text{new}} \end{pmatrix} = D_i, \quad \begin{pmatrix} U_i^{\text{new}} W_{i+1}^{\text{new}} \\ U_{i+1}^{\text{new}} \end{pmatrix} = U_i, \\ (P_i^{\text{new}} P_{i+1}^{\text{new}} R_i^{\text{new}}) = P_i, \quad ((V_i^{\text{new}})^H \quad W_i^{\text{new}} (V_{i+1}^{\text{new}})^H) = V_i^H, \quad W_i^{\text{new}} W_{i+1}^{\text{new}} = W_i, \\ R_{i+1}^{\text{new}} R_i^{\text{new}} = R_i, \quad (R_{i+1}^{\text{new}} (Q_i^{\text{new}})^H \quad (Q_{i+1}^{\text{new}})^H) = Q_i^H.$$

To solve these equations, we partition the matrices for the old i th block conformally with the two new blocks as

$$D_i = \begin{pmatrix} D_i^{11} & D_i^{12} \\ D_i^{21} & D_i^{22} \end{pmatrix}, \quad U_i = \begin{pmatrix} U_i^1 \\ U_i^2 \end{pmatrix}, \quad P_i = \begin{pmatrix} P_i^1 \\ P_i^2 \end{pmatrix}, \\ V_i^H = ((V_i^1)^H \quad (V_i^2)^H), \quad Q_i^H = ((Q_i^1)^H \quad (Q_i^2)^H).$$

These equations allow us to identify

$$D_i^{\text{new}} = D_i^{11}, \quad D_{i+1}^{\text{new}} = D_i^{22}, \quad U_{i+1}^{\text{new}} = U_i^2, \quad P_i^{\text{new}} = P_i^1, \quad V_i^{\text{new}} = V_i^1, \quad Q_{i+1}^{\text{new}} = Q_i^2.$$

The remaining matrices satisfy

$$\begin{pmatrix} (V_i^2)^H & W_i \\ D_i^{12} & U_i^1 \end{pmatrix} = \begin{pmatrix} W_i^{\text{new}} \\ U_{i+1}^{\text{new}} \end{pmatrix} ((V_{i+1}^{\text{new}})^H \quad W_{i+1}^{\text{new}}), \\ \begin{pmatrix} P_i^2 & D_i^{21} \\ R_i & (Q_i)^H \end{pmatrix} = \begin{pmatrix} P_{i+1}^{\text{new}} \\ R_{i+1}^{\text{new}} \end{pmatrix} (R_i^{\text{new}} \quad (Q_i^{\text{new}})^H).$$

By factorizing the left-hand side matrices using numerical tools such as the SVD and rank-revealing QR factorizations [13, 27], these two equations allow us to compute an effective representation of those remaining matrices for the new blocks.

3.7. Fast model reduction. As became evident in the presentation of the superfast solver, sometimes we produce SSS representations that are not as compact as possible (to some specified tolerance τ). Dewilde and van der Veen [16] present a technique to find the optimal reduced-order model. Here we present a simple, efficient, and numerically stable method to compress a given SSS representation to a prespecified tolerance τ . This method is closely related to the algorithms presented in [16] for computing minimal realizations in input and output normal forms.

We will present the technique using SVDs. In particular we will call the SVD from which all singular values that are less than τ have been discarded as a τ -accurate SVD. It is a simple matter to replace the SVD in these calculations with rank-revealing QR factorizations, or rank-revealing LU factorizations, or even LU factorization with complete pivoting. It is likely that this will lead to considerable speedup for a small loss in compression.

Our algorithm for model reduction can be viewed as a natural extension of the algorithm presented in [6] for constructing SSS representations. More formally, suppose we have a nominal SSS representation for the matrix A . Given a tolerance τ , we would like to compute a new SSS representation for A , accurate to this new tolerance, but more rapidly than the algorithm presented in [6].

First, we observe that it is enough to specify the method for U_i , W_i , and V_i , since the same technique can then be applied to P_i , R_i , and Q_i . We split the method into two stages. In the first stage we convert the representation into left proper form we mean that all the column bases C_i of the Hankel-blocks,² where

$$\begin{aligned} C_1 &= U_1, \\ C_i &= \begin{pmatrix} C_{i-1}W_i \\ U_i \end{pmatrix}, \end{aligned}$$

should have orthonormal columns. In the second stage we convert the representation into right proper form; that is, now all the row bases G_i of the Hankel-blocks, where

$$\begin{aligned} G_n &= V_n, \\ G_i &= \begin{pmatrix} V_i \\ G_{i+1}W_i^H \end{pmatrix}, \end{aligned}$$

will have orthonormal columns. The second stage recursions will essentially be first-stage recursions in the opposite order. Note that $H_i = C_i G_{i+1}^H$.

Note that the method of [6] already produces SSS representations in left proper form. However, it is likely that updating operations will destroy proper form. So we begin by indicating how the left proper form can be restored efficiently. For convenience, we use hats to denote the representation in left proper form.

Consider the following recursions:

$$\begin{aligned} \begin{pmatrix} \tilde{W}_i \\ U_i \end{pmatrix} &\approx \begin{pmatrix} \hat{W}_i \\ \hat{U}_i \end{pmatrix} \Sigma_i F_i^H, & \tau\text{-accurate SVD factorization,} \\ \tilde{W}_{i+1} &= \Sigma_i F_i^H W_{i+1}, \\ \hat{V}_{i+1} &= V_{i+1} F_i \Sigma_i^H, \end{aligned}$$

²The term Hankel-block is taken from [16]. It denotes the off-diagonal blocks that extend from the diagonal to the northeast corner (for the upper case) or to the southwest corner (for the lower case).

with the understanding that $\tilde{W}_1$ and $\hat{W}_1$ are empty matrices. Then it is easy to check that the new column bases

$$\begin{aligned}\hat{C}_1 &= \hat{U}_1, \\ \hat{C}_i &= \begin{pmatrix} \hat{C}_{i-1}\hat{W}_i \\ \hat{U}_i \end{pmatrix}\end{aligned}$$

have orthonormal columns and that the hatted sequences form a valid SSS representation for the given matrix. The hatted SSS representation will be accurate to τ , provided we assume that the two-norm of the G_i 's is bounded by a small multiple of the norm of the original matrix. We will call such SSS representations τ -accurate. Also note that the recursions depend only linearly on the matrix size. If by some chance the SSS representation is unstable, then τ must be set to zero in the first stage. It should then be restored to its proper value in the second stage.

In the next stage we take an SSS representation in left proper form and further compress it to the given tolerance τ efficiently by converting it into right proper form. For simplicity we assume that the given SSS representation is already in left proper form and denote it using unhatted quantities. We use hatted quantities to denote terms in the compressed right proper form representation. In the second stage it is sufficient to concentrate on the row bases G_i since $H_i = C_i G_{i+1}^H$, and C_i has orthonormal columns by our assumption of left proper form.

However, this time we must run the recursions backward in time. Here they are

$$\begin{aligned}\begin{pmatrix} V_i \\ \tilde{W}_i^H \end{pmatrix} &\approx \begin{pmatrix} \hat{V}_i \\ \hat{W}_i^H \end{pmatrix} \Sigma_i F_i^H, & \tau\text{-accurate SVD factorization,} \\ \tilde{W}_{i-1} &= W_{i-1} F_i \Sigma_i^H, \\ \hat{U}_{i-1} &= U_{i-1} F_i \Sigma_i^H,\end{aligned}$$

with the understanding that $\tilde{W}_n$ and $\hat{W}_n$ are empty matrices. It can be seen that the hatted sequences form a valid SSS representation in right proper form and that the approximations are τ accurate. We draw attention to the fact that in each stage the W_i matrices are transformed twice, first to $\tilde{W}$ and then to $\hat{W}_i$.

4. Related and future work. The algebraic formalism of the structure that is used in this paper appears in work of Dewilde and van der Veen [16] and Eidelman and Gohberg [18, 19, 20, 21]. Other structures that would work just as well appear in the works of Starr [38], Starr and Rokhlin [39], Hackbusch [29], Hackbusch and Khoromskij [30, 31], Hackbusch, Khoromskij, and Sauter [32], and Chandrasekaran and Gu [9]. Fast, direct, but not necessarily stable algorithms for such matrices were first presented in Starr's thesis [38, 39]. They also appear in the works of Eidelman and Gohberg [18, 19] and Hackbusch [29] and Hackbusch and Khoromskij [30, 31]. Fast and stable direct solvers appeared first in the works of Dewilde and van der Veen [16] and independently later in the work of Chandrasekaran and Gu [8]. Various generalizations have been carried out by various authors, including us. Here is a brief list of such works: [10, 11, 12, 30, 31, 32, 34, 36].

The computational electromagnetics literature has also looked at this problem independently. Relevant work includes that of Gurel and Chew [28], Canning and Rogovin [4], and Gope and Jandhyala [24].

In this paper we presented a fast and stable direct method for solving a linear least squares problem whose coefficient matrix satisfies the SSS matrix structure as

defined in (1). We also presented a fast solver for handling multiple right-hand sides that are also—in addition to the coefficient matrix—in SSS form.

The fast multipole method (FMM) of Carrier, Greengard, and Rokhlin [5], Greengard and Rokhlin [25], and Rokhlin [37] has evolved into a major workhorse for many computationally intensive problems in a wide area of applications. It turns out that the matrix structure exploited by the FMM to speed up the computation is closely related to the SSS structure exploited in this paper. Future work includes developing fast direct solvers for linear systems of equations where the coefficient matrix has the FMM matrix structure.

We will also explore the applications of the fast SSS algorithms to speed up Kress's global discretization technique for solving the integral equations of scattering theory [14, 15].

Acknowledgments. The authors are grateful to Nathan Champagne and Robert Sharpe of Lawrence Livermore National Laboratory for their suggestions and interest in this research and to Jianlin Xia and Jiang Zhu of the University of California at Berkeley for their help in preparing portions of this paper.

REFERENCES

- [1] D. ALPAY AND P. DEWILDE, *Time-varying signal approximation and estimation*, in Signal Processing, Scattering, and Operation Theory, and Numerical Methods, Birkhäuser Boston, Boston, MA, 1990, pp. 1–22.
- [2] D. ALPAY, P. DEWILDE, AND H. DYM, *Lossless inverse scattering and reproducing kernels for upper triangular matrices*, in Extension and Interpolation of Linear Operators and Matrix Functions, Birkhäuser, Basel, 1990, pp. 61–135.
- [3] E. ANDERSON, Z. BAI, C. BISCHOF, J. DEMMEL, J. DONGARRA, J. DU CROZ, A. GREENBAUM, S. HAMMARLING, A. MCKENNEY, S. OSTROUCHOV, AND D. SORESENSEN, *LAPACK Users' Guide*, 2nd ed., SIAM, Philadelphia, 1994.
- [4] F. X. CANNING AND K. ROGOVIN, *Fast direct solution of moment-method matrices*, IEEE Antennas and Propagation Magazine, 40 (1998), pp. 15–26.
- [5] J. CARRIER, L. GREENGARD, AND V. ROKHLIN, *A fast adaptive multipole algorithm for particle simulations*, SIAM J. Sci. Statist. Comput., 9 (1988), pp. 669–686.
- [6] S. CHANDRASEKARAN, P. DEWILDE, M. GU, T. PALS, AND A.-J. VAN DER VEEN, *Fast stable solver for sequentially semi-separable linear systems of equations*, in High Performance Computing—HiPC 2002: 9th International Conference, Lecture Notes in Comput. Sci. 2552, Springer-Verlag, Heidelberg, 2002, pp. 545–554.
- [7] S. CHANDRASEKARAN, P. DEWILDE, M. GU, T. PALS, X. SUN, A.-J. VAN DER VEEN, AND D. WHITE, *Fast Stable Solvers for Sequentially Semi-separable Linear Systems of Equations and Least Squares Problems*, Technical report, University of California, Berkeley, CA, 2003.
- [8] S. CHANDRASEKARAN AND M. GU, *Fast and stable algorithms for banded plus semiseparable systems of linear equations*, SIAM J. Matrix Anal. Appl., 25 (2003), pp. 373–384.
- [9] S. CHANDRASEKARAN AND M. GU, *A fast and stable solver for recursively semi-separable systems of equations*, in Structured Matrices in Mathematics, Computer Science and Engineering, II, V. Olshevsky, ed., Contemp. Math. 281, AMS, Providence, RI, 2001, pp. 39–53.
- [10] S. CHANDRASEKARAN AND M. GU, *Fast and stable eigendecomposition of symmetric banded plus semi-separable matrices*, Linear Algebra Appl., 313 (2000), pp. 107–114.
- [11] S. CHANDRASEKARAN AND M. GU, *A divide-and-conquer algorithm for the eigendecomposition of symmetric block-diagonal plus semiseparable matrices*, Numer. Math., 96 (2004), pp. 723–731.
- [12] S. CHANDRASEKARAN, M. GU, AND T. PALS, *A Fast and Stable Solver for Smooth Recursively Semi-Separable Systems*, paper presented at the SIAM Annual Conference, San Diego, CA, 2001, and SIAM Conference of Linear Algebra in Controls, Signals and Systems, Boston, MA, 2001.
- [13] S. CHANDRASEKARAN AND I. C. F. IPSSEN, *On rank-revealing QR factorisations*, SIAM J. Matrix Anal. Appl., 15 (1994), pp. 592–622.
- [14] D. COLTON AND R. KRESS, *Integral Equation Methods in Scattering Theory*, Wiley, New York, 1983.

- [15] D. COLTON AND R. KRESS, *Inverse Acoustic and Electromagnetic Scattering Theory*, Appl. Math. Sci. 93, Springer-Verlag, Berlin, 1992.
- [16] P. DEWILDE AND A. VAN DER VEEN, *Time-Varying Systems and Computations*, Kluwer Academic Publishers, Boston, MA, 1998.
- [17] P. DEWILDE AND A. J. VAN DER VEEN, *Inner-outer factorization and the inversion of locally finite systems of equations*, Linear Algebra Appl., 313 (2000), pp. 53–100.
- [18] Y. EIDELMAN AND I. GOHBERG, *Inversion formulas and linear complexity algorithm for diagonal plus semiseparable matrices*, Comput. Math. Appl., 33 (1997), Elsevier, pp. 69–79.
- [19] Y. EIDELMAN AND I. GOHBERG, *A look-ahead block Schur algorithm for diagonal plus semiseparable matrices*, Comput. Math. Appl., 35 (1998), pp. 25–34.
- [20] Y. EIDELMAN AND I. GOHBERG, *On a new class of structured matrices*, Integral Equations Operator Theory, 34 (1999), pp. 293–324.
- [21] Y. EIDELMAN AND I. GOHBERG, *A modification of the Dewilde van der Veen method for inversion of finite structured matrices*, Linear Algebra Appl., 343/344 (2002), pp. 419–450.
- [22] I. GOHBERG, T. KAILATH, AND I. KOLTRACHT, *Linear complexity algorithms for semiseparable matrices*, Integral Equations Operator Theory, 8 (1985), pp. 780–804.
- [23] G. GOLUB AND C. VAN LOAN, *Matrix Computations*, 3rd ed., The Johns Hopkins University Press, Baltimore, MD, 1996.
- [24] D. GOPE AND V. JANDHYALA, *An iteration-free fast multilevel solver for dense method of moment systems*, in Proceedings of the IEEE meeting on Electrical Performance of Electronic Packaging, 2001, Boston, MA, pp. 177–180.
- [25] L. GREENGARD AND V. ROKHLIN, *A fast algorithm for particle simulations*, J. Comput. Phys., 73 (1987), pp. 325–348.
- [26] M. GU, *Backward perturbation bounds for linear least squares problems*, SIAM J. Matrix Anal. Appl., 20 (1998), pp. 363–372.
- [27] M. GU AND S. C. EISENSTAT, *Efficient algorithms for computing a strong rank-revealing QR factorization*, SIAM J. Sci. Comput., 17 (1996), pp. 848–869.
- [28] L. GUREL AND W. C. CHEW, *Fast direct (non-iterative) solvers for integral-equation formulations of scattering problems*, in Proceedings of the IEEE Antennas and Propagation Society International Symposium, 1998 Digest, Antennas: Gateways to the Global Network, Vol. 1., Atlanta, GA, 1998, pp. 298–301.
- [29] W. HACKBUSCH, *A sparse arithmetic based on $\mathcal{H}$ -matrices. Part-I: Introduction to $\mathcal{H}$ -matrices*, Computing, 62 (1999), pp. 89–108.
- [30] W. HACKBUSCH AND B. N. KHOROMSKIJ, *A sparse $\mathcal{H}$ -matrix arithmetic. Part-II: Application to multi-dimensional problems*, Computing, 64 (2000), pp. 21–47.
- [31] W. HACKBUSCH AND B. N. KHOROMSKIJ, *A sparse $\mathcal{H}$ -matrix arithmetic: General complexity estimates*, J. Comput. Appl. Math., 125 (2000), pp. 79–501.
- [32] W. HACKBUSCH, B. N. KHOROMSKIJ, AND S. SAUTER, *On H^2 -Matrices*, preprint 50, MPI, Leipzig, 1999.
- [33] T. KAILATH, *Fredholm resolvents, Wiener-Hopf equations, and Riccati differential equations*, IEEE Trans. Inform. Theory, IT-15 (1969), pp. 655–672.
- [34] N. MASTRONARDI, S. CHANDRASEKARAN, AND S. VAN HUFFEL, *Fast and stable two-way chasing algorithm for diagonal plus semi-separable systems of linear equations*, Numer. Linear Algebra Appl., 38 (2000), pp. 7–12.
- [35] A. P. MULLHAUPT AND K. S. RIEDEL, *Low grade matrices and matrix fraction representations*, Linear Algebra Appl., 342 (2002), pp. 187–201.
- [36] N. MASTRONARDI, S. CHANDRASEKARAN, AND S. VAN HUFFEL, *Fast and stable algorithms for reducing diagonal plus semi-separable matrices to tridiagonal and bidiagonal form*, BIT, 41 (2001), pp. 149–157.
- [37] V. ROKHLIN, *Applications of volume integrals to the solution of PDEs*, J. Comput. Phys., 86 (1990), pp. 414–439.
- [38] P. STARR, *On the Numerical Solution of One-Dimensional Integral and Differential Equations*, Research report YALEU/DCS/RR-888, Yale University, New Haven, CT, 1991.
- [39] P. STARR AND V. ROKHLIN, *On the numerical solution of 2-point boundary value problem. II.*, Comm. Pure Appl. Math., 47 (1994), pp. 1117–1159.
- [40] A.-J. VAN DER VEEN, *Time-varying lossless systems and the inversion of large structured matrices*, Archiv f. Elektrotechnik u. Übertragungstechnik, 49 (1995) pp. 372–382.
- [41] N. YARVIN AND V. ROKHLIN, *A generalized one-dimensional fast multipole method with application to filtering of spherical harmonics*, J. Comput. Phys., 147 (1998), pp. 594–609.

RELAXATIONS FOR ROBUST LINEAR MATRIX INEQUALITY PROBLEMS WITH VERIFICATIONS FOR EXACTNESS*

C. W. SCHERER†

Abstract. Robust semidefinite programming problems with rational dependence on uncertainties are known to have a wide range of applications, in particular in robust control. It is well established how to systematically construct relaxations on the basis of the full block S-procedure. In general, such relaxations are expected to be conservative, but for concrete problem instances they are often observed to be tight. The main purpose of this paper is to suggest novel computationally verifiable conditions for when general classes of linear matrix inequality relaxations do not involve any conservatism. If the convex set of uncertainties is finitely generated, we suggest a novel sequence of relaxations which can be proved to be asymptotically exact. Finally, our results are applied to the particularly relevant robustness analysis problem for linear time-invariant dynamical systems affected by uncertainties that are full ellipsoidal or repeated and contained in intersections of disks or circles. This leads to extensions of known results on relaxation exactness for small block structures, including an elementary proof for tightness of standard structured singular value computations for three full complex uncertainty blocks.

Key words. robust semidefinite programming, higher-order relaxations, exactness tests, robust performance analysis, structured singular value, linear matrix inequalities

AMS subject classifications. 90C22, 62G35, 93B52, 93D09, 58E25

DOI. 10.1137/S0895479803430953

1. Introduction. It is a fundamental insight that a whole variety of robust analysis and synthesis questions in control can be translated into so-called robust linear matrix inequality (LMI) problems. These are variants of standard LMI problems in which the data matrices are subject to bounded structured uncertainties. As a typical instance, consider the semidefinite program (SDP) to infimize, for some given real vector c and Hermitian matrices $F_0, F_1, \dots, F_n$, the functional $\langle c, x \rangle = c_1 x_1 + \dots + c_n x_n$ over all $x \in \mathbb{R}^n$ which satisfy the LMI $F_0 + x_1 F_1 + \dots + x_n F_n < 0$. If the data matrices $F_0(\delta), F_1(\delta), \dots, F_n(\delta)$ are functions of some real or complex parameter δ that is only known to be contained in some set $\mathcal{D}$, and the goal is to solve the robust counterpart, then infimize $\langle c, x \rangle$ over all $x \in \mathbb{R}^n$ such that

$$(1.1) \quad F_0(\delta) + x_1 F_1(\delta) + \dots + x_n F_n(\delta) < 0 \quad \text{for all } \delta \in \mathcal{D}.$$

The modeling power of this framework, in particular for robust optimization and robust linear algebra, has been known for a long time and has recently been summarized, e.g., in [13, 3]. We are mainly interested in robust control where this problem formulation captures robust performance analysis, such as in standard structured singular value theory and considerable generalizations thereof [29]. Similarly for linear-parameter-varying systems, it covers the problem of constructing quadratic-in-the-state Lyapunov functions which are constant [2], quadratic [44, 26], or even polynomial or rational in the parameters [12]. Finally, the synthesis of robust state

*Received by the editors July 2, 2003; accepted for publication (in revised form) by L. Vandenberghe December 20, 2004; published electronically September 15, 2005. This research was supported by the Technology Foundation STW, the Applied Science Division of NWO, and the Technology Programme of the Ministry of Economic Affairs.

<http://www.siam.org/journals/simax/27-2/43095.html>

†Delft Center for Systems and Control, Delft University of Technology, Mekelweg 2, 2628 CD Delft, The Netherlands (c.w.scherer@dcsc.tudelft.nl).

feedback and observer gains or of linear parameter-varying controllers is also easily molded into this framework.

If $F_j(\delta)$, $j = 0, 1, \dots, n$ depend affinely on δ and $\boldsymbol{\delta}$ is the convex hull of (a moderate number of) finitely many generators, the described robust semidefinite program can be readily reduced to a standard LMI problem. If the number of extreme points to describe $\boldsymbol{\delta}$ is large, one can construct an efficiently computable relaxation with beautiful a priori guarantees on the relaxation error [4]. The situation drastically differs in the case where the dependence is nonlinear in δ . If an explicit parametrization of $\boldsymbol{\delta}$ requires a large number of parameters and only probabilistic guarantees are required, the algorithms in [33] provide a viable and computationally attractive approach. Since these algorithms are relying on sampling (and hence on an inner approximation of) the set $\boldsymbol{\delta}$, they typically generate ϵ -suboptimal solutions on the exact optimal value of the robust LMI problem.

For guarantees with certainty, as are often required for stability and performance in control, the interest is in computing upper bounds by multiplier relaxations. For a large variety of interesting problem classes it can be assumed that the data matrices are rational functions of δ without pole at zero. This implies [42, 13] that one can construct a linear fractional representation

$$\frac{1}{2} \begin{pmatrix} F_0(\delta) \\ F_1(\delta) \\ \vdots \\ F_n(\delta) \end{pmatrix} = \begin{pmatrix} C_0 \\ C_1 \\ \vdots \\ C_n \end{pmatrix} \Delta(\delta) (I - A\Delta(\delta))^{-1} B + \begin{pmatrix} D_0 \\ D_1 \\ \vdots \\ D_n \end{pmatrix}$$

with $\Delta(\delta)$ being block-diagonal and depending linearly on δ . Although not explicitly reflected in the description, we stress that the blocks $C_j \Delta(\delta) (I - A\Delta(\delta))^{-1} B + D_j$ are guaranteed to be Hermitian. If we abbreviate

$$W(x) = \begin{pmatrix} 0 & I \\ C_0 & D_0 \\ C_1 & D_1 \\ \vdots & \vdots \\ C_n & D_n \end{pmatrix}' \begin{pmatrix} 0 & I & x_1 I & \cdots & x_n I \\ I & 0 & 0 & \cdots & 0 \\ x_1 I & 0 & 0 & \cdots & 0 \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ x_n I & 0 & 0 & \cdots & 0 \end{pmatrix} \begin{pmatrix} 0 & I \\ C_0 & D_0 \\ C_1 & D_1 \\ \vdots & \vdots \\ C_n & D_n \end{pmatrix},$$

we infer

$$F_0(\delta) + \sum_{j=1}^n x_j F_j(\delta) = \begin{pmatrix} \Delta(\delta)(I - A\Delta(\delta))^{-1} B \\ I \end{pmatrix}' W(x) \begin{pmatrix} \Delta(\delta)(I - A\Delta(\delta))^{-1} B \\ I \end{pmatrix}.$$

If we hence define $\boldsymbol{\Delta} := \{\Delta(\delta) : \delta \in \boldsymbol{\delta}\}$, the robust semidefinite program amounts to minimizing $\langle c, x \rangle$ over all $x \in \mathbb{R}^n$ with

$$(1.2) \quad \begin{pmatrix} \Delta(I - A\Delta)^{-1} B \\ I \end{pmatrix}' W(x) \begin{pmatrix} \Delta(I - A\Delta)^{-1} B \\ I \end{pmatrix} < 0 \quad \text{for all } \Delta \in \boldsymbol{\Delta}.$$

We obtain the seemingly specialized, but in fact rather general, problem formulation as considered throughout this paper, with further illustrations provided in section 2.

Let us now recall how one can construct relaxations for computing upper bounds that are based on the so-called full-block S-procedure as proved independently in [22, 26] and [37, 38]. If $\boldsymbol{\Delta}$ is compact and $I - A\Delta$ is nonsingular for all $\Delta \in \boldsymbol{\Delta}$, the

considered robust SDP is equivalent to minimizing $\langle c, x \rangle$ over $x \in \mathbb{R}^n$ and Hermitian matrices P with

$$(1.3) \quad \begin{pmatrix} \Delta \\ I \end{pmatrix}' P \begin{pmatrix} \Delta \\ I \end{pmatrix} > 0 \quad \text{for all } \Delta \in \Delta$$

such that

$$(1.4) \quad \begin{pmatrix} I & 0 \\ A & B \end{pmatrix}' P \begin{pmatrix} I & 0 \\ A & B \end{pmatrix} + W(x) < 0.$$

This step turns the rational dependence of the original semi-infinite semidefinite programming problem into quadratic dependence on the uncertain parameters at the expense of introducing the so-called multiplier variable P . The actual difficulty in its numerical implementation results from the implicit characterization of the set of all feasible multipliers in (1.3) in terms of infinitely many inequalities. In practice the goal is to find, or approximate, this multiplier set to obtain relaxations for computing, or bounding, as required for guarantees with certainty.

We will recall various such approximations that have been suggested in the literature, ranging from diagonal multipliers for diagonal uncertainties up to full block multipliers where (1.3) is guaranteed with convexity arguments. In general it cannot be expected that these relaxations are exact, and it is typically not known how to gradually reduce the gap between the precise value and the computed approximation.

Although it is well known that the efficient reduction of the relaxation gap is not possible, [9, 43, 15, 16], computational experience reveals that this is indeed possible for [9, 43, 15, 16]. This raises the issue of developing reliable tools that allow us to numerically verify whether a chosen relaxation is exact, and hence does not involve any conservatism.

We are led to the main contributions of this paper. In section 2 we formulate the precise problem together with a generic family of multiplier relaxations, complemented by illustrations with concrete examples. In section 3 we prove a novel exactness characterization in terms of the relaxation's Lagrange dual, while including an explicit discussion of the relation to known rank-one exactness principles. The brief section 4 is devoted to the verification of a fundamental well-posedness hypothesis in the problem formulation by solving a phase I problem. In section 5 we consider in somewhat more detail extensions of known relaxations in robust control, with unifying elementary proofs for exactness for small block structures. Section 6 comprises a detailed analysis of under which conditions the suggested exactness tests are indeed computationally verifiable by standard SDP solvers. Finally, for finitely generated uncertainty sets and based on a generalization of Pólya's theorem to matrix-valued polynomials, we present in section 7 a new family of relaxations which can be proven to be asymptotically exact.

2. Relaxations of robust LMI problems. With real or complex data matrices A and B and a set Δ of real or complex structured matrices, define the rational function

$$F(\Delta) := \Delta(I - A\Delta)^{-1}B \quad \text{for } \Delta \in \Delta.$$

Moreover, for Hermitian matrices $W_0, W_1, \dots, W_n$ define the affine mapping

$$W(x) := W_0 + x_1 W_1 + \dots + x_n W_n \quad \text{for } x \in \mathbb{R}^n.$$

For compactness of notation, let us introduce the abbreviation

$$L(x, \Delta) := \begin{pmatrix} F(\Delta) \\ I \end{pmatrix}' W(x) \begin{pmatrix} F(\Delta) \\ I \end{pmatrix}.$$

If c is a real cost vector, this paper revolves around the following problem.

$$(2.1) \quad \begin{array}{ll} \text{minimize} & \langle c, x \rangle \text{ over all } x \in \mathbb{R}^n \text{ such that} \\ & L(x, \Delta) < 0 \text{ for all } \Delta \in \mathbf{\Delta}. \end{array}$$

The optimal value is denoted by $\gamma_{\text{opt}} \in [-\infty, \infty]$.

Throughout this paper we make the following assumption, which guarantees that $F(\Delta)$ is well-defined on $\mathbf{\Delta}$. Section 4 reveals how to numerically verify this property.

$$I - A\Delta \text{ is nonsingular for all } \Delta \in \mathbf{\Delta}.$$

Along with (ROB), we consider approximation problems that are constructed as follows. We choose linear Hermitian-valued mappings $G(y)$ and $H(y)$, defined in the variable y that lives in some finite-dimensional inner product space, such that

$$(2.2) \quad G(y) \leq 0 \text{ implies } \begin{pmatrix} \Delta \\ I \end{pmatrix}' H(y) \begin{pmatrix} \Delta \\ I \end{pmatrix} \geq 0 \text{ for all } \Delta \in \mathbf{\Delta}.$$

It causes no loss of generality [7, section 2.5] to assume that

$$(2.3) \quad G(y) < 0 \text{ is feasible.}$$

With any such pair of mappings G, H let us now consider the following relaxation.

$$(2.4) \quad \begin{array}{ll} \text{minimize} & \langle c, x \rangle \text{ over all } x \text{ and } y \text{ with} \\ & G(y) < 0, \begin{pmatrix} I & 0 \\ A & B \end{pmatrix}' H(y) \begin{pmatrix} I & 0 \\ A & B \end{pmatrix} + W(x) < 0. \end{array}$$

Computing its optimal value denoted by $\gamma_{\text{rel}} \in [-\infty, \infty]$ (while suppressing the dependence on G, H) clearly amounts to solving a standard linear SDP.

Recall that any Hermitian matrix P which satisfies (1.3) is said to be a multiplier for $\mathbf{\Delta}$. Obviously, $\{H(y) : G(y) < 0\}$ and its closure $\{H(y) : G(y) \leq 0\}$ are subsets of the set of all multipliers that admit an explicit description in terms of LMIs. It is worthwhile to stress that any set with an LMI representation can be parameterized (with suitable mappings G, H) in this fashion [3]. If the multiplier sets of two relaxations with values $\gamma_{\text{rel},1}, \gamma_{\text{rel},2}$ satisfy $\{H_1(y) : G_1(y) < 0\} \subset \{H_2(y) : G_2(y) < 0\}$, it is clear that $\gamma_{\text{rel},1} \geq \gamma_{\text{rel},2}$. Therefore the larger the multiplier set (in the partial order induced by inclusion) the better the relaxation. $\square$

It is easy to establish the following general relation of (ROB) and (REL), which also clarifies in which sense (REL) is actually a relaxation of (ROB).

LEMMA 2.1.

$\gamma_{\text{opt}} = \pm\infty \iff \gamma_{\text{rel}} = \pm\infty$. We just exploit the easily verified relation

$$(2.5) \quad \begin{pmatrix} I & 0 \\ A & B \end{pmatrix} \begin{pmatrix} F(\Delta) \\ I \end{pmatrix} = \begin{pmatrix} \Delta \\ I \end{pmatrix} (I - A\Delta)^{-1} B \text{ for all } \Delta \in \mathbf{\Delta}.$$

If (REL) has no feasible solution, we infer $\gamma_{\text{rel}} = \infty$, and there is nothing to prove. Now suppose (x, y) is feasible for (REL). For any $\Delta \in \mathbf{\Delta}$, we left-multiply the second inequality in (2.4) with $\text{col}(F(\Delta), I)'$ and right-multiply with $\text{col}(F(\Delta), I)$ to obtain

$$[(I - A\Delta)^{-1} B]' \begin{pmatrix} \Delta \\ I \end{pmatrix}' H(y) \begin{pmatrix} \Delta \\ I \end{pmatrix} (I - A\Delta)^{-1} B + L(x, \Delta) < 0.$$

Since $G(y) < 0$, we infer from (2.2) that $L(x, \Delta) < 0$. Since $\Delta \in \mathbf{\Delta}$ was arbitrary, we can conclude that x is actually feasible for (ROB), which in turn implies that $\gamma_{\text{opt}} \leq \gamma_{\text{rel}}$. $\square$

If $\gamma_{\text{rel}} = -\infty$, we can hence infer that $\gamma_{\text{opt}} = -\infty$ and the relaxation is exact. In all other cases it is impossible to make further a priori statements on the

$$\text{relaxation gap } \gamma_{\text{rel}} - \gamma_{\text{opt}} \geq 0$$

in the generality as discussed here. One of the central goals of this paper is to develop numerically verifiable conditions for the gap to vanish, which means that the relaxation is actually exact. Before continuing we intend to illustrate the power of (ROB) and (REL) through various examples.

2.1. Robust performance analysis for system interconnections. An interconnection of linear time-invariant uncertain systems can be easily represented in linear fractional form as [46]

$$\Delta(s) \star \begin{pmatrix} A(s) & B(s) \\ C(s) & D(s) \end{pmatrix} := D(s) + C(s)\Delta(s)(I - A(s)\Delta(s))^{-1}B(s),$$

where $A(s)$, $B(s)$, $C(s)$, and $D(s)$ are given proper and stable transfer matrices, and the proper and stable uncertainty transfer matrix $\Delta(s)$ admits some specific (often block-diagonal) structure with its size bounded in terms of its frequency response [29, 46]. Both structure and size of Δ can be specified with a suitable set $\mathbf{\Delta}$ of complex matrices by requiring $\Delta(s) \in \mathbf{\Delta}$ for all $s \in i\mathbb{R} \cup \{\infty\}$. If $\mathbf{\Delta}$ is star-shaped with star-center zero, the question of whether the system interconnection remains stable for all perturbation can be reduced to the algebraic problem of whether $\det(I - A(s)\Delta) \neq 0$ for all $s \in i\mathbb{R} \cup \{\infty\}$, $\Delta \in \mathbf{\Delta}$. With the spectral norm $\|\cdot\|$, the worst-case H_∞ -performance level is smaller than γ if

$$(2.6) \quad \|D(s) + C(s)\Delta(I - A(s)\Delta)^{-1}B(s)\| < \gamma \quad \text{for all } s \in i\mathbb{R} \cup \{\infty\}, \Delta \in \mathbf{\Delta}.$$

After fixing $s \in i\mathbb{R} \cup \{\infty\}$, the robust stability test amounts to checking (A), while the inequality in (2.6) can be easily reformulated, after the substitution $x = \gamma^2$, as

$$(2.7) \quad \begin{pmatrix} \Delta(I - A(s)\Delta)^{-1}B(s) \\ I \end{pmatrix}' W(x) \begin{pmatrix} \Delta(I - A(s)\Delta)^{-1}B(s) \\ I \end{pmatrix} < 0$$

with the abbreviation

$$W(x) = \begin{pmatrix} 0 & I \\ C(s) & D(s) \end{pmatrix}' \begin{pmatrix} -xI & 0 \\ 0 & I \end{pmatrix} \begin{pmatrix} 0 & I \\ C(s) & D(s) \end{pmatrix}.$$

Therefore (ROB) defines the worst gain of the transfer matrix at s with respect to the uncertainty set $\mathbf{\Delta}$. It is well known how to handle a whole variety of other practically relevant performance criteria, such as robust passivity, by straightforward modifications of $W(x)$.

2.2. A typical uncertainty structure and relaxation. For a general $W(x)$, let us now recall block structures as they appear in robust control [29, 46, 23, 36, 24, 13, 32]. With the column partitions $I = (E_1 \cdots E_m)$ and $I = (F_1 \cdots F_m)$ of the identity matrix and the index set $K := \{1, \dots, m\}$ suppose that

$$\mathbf{\Delta} := \left\{ \Delta = \sum_{k \in K} E_k \Delta_k F_k' : \Delta_k \in \mathbf{\Delta}_k \right\},$$

where Δ_k imposes itself additional structural and size constraints on the subblock Δ_k . If $G_k(y_k)$, $H_k(y_k)$ are linear maps in y_k such that

$$(2.8) \quad G_k(y_k) \leq 0 \text{ implies } \begin{pmatrix} \Delta_k \\ I \end{pmatrix}' H_k(y_k) \begin{pmatrix} \Delta_k \\ I \end{pmatrix} \geq 0 \text{ for all } \Delta_k \in \Delta_k,$$

we can then define for $y = (y_1, \dots, y_m)$ the linear maps

$$H(y) := \sum_{k \in K} \begin{pmatrix} E_k & 0 \\ 0 & F_k \end{pmatrix} H_k(y_k) \begin{pmatrix} E_k & 0 \\ 0 & F_k \end{pmatrix}', \quad G(y) := \text{diag}_{k \in K}(G_k(y_k))$$

to guarantee (2.2), simply since

$$\left(\sum_{k \in K} \begin{pmatrix} E_k \Delta_k F_k' \\ I \end{pmatrix}' \right) H(y) \left(\sum_{k \in K} \begin{pmatrix} E_k \Delta_k F_k' \\ I \end{pmatrix} \right) = \sum_{k \in K} F_k \begin{pmatrix} \Delta_k \\ I \end{pmatrix}' H_k(y_k) \begin{pmatrix} \Delta_k \\ I \end{pmatrix} F_k'.$$

Let us illustrate the construction for the following three concrete standard block types with index sets K_f , K_r , and K_h forming a partition of K .

For $k \in K_f$ assume that

$$\Delta_k = \left\{ \Delta_k \in \mathbb{C}^{n_k \times m_k} : \begin{pmatrix} \Delta_k \\ I \end{pmatrix}' P_j^k \begin{pmatrix} \Delta_k \\ I \end{pmatrix} \geq 0, \quad j = 1, \dots, b_k \right\}$$

with fixed Hermitian matrices P_j^k . This includes spectral norm bounds $\|\Delta_k\| \leq 1$, positive realness constraints $\Delta_k' + \Delta_k \geq 0$, and mixtures thereof as the most important special cases. Note that Δ_k admits an LMI description if the left-upper blocks of all P_j^k are negative semidefinite, and it is compact if at least one is negative definite. We can trivially guarantee (2.8) with

$$y_k = (y_1^k, \dots, y_{b_k}^k) \in \mathbb{R}^{b_k}, \quad H_k(y_k) := \sum_{j=1}^{b_k} y_j^k P_j^k, \quad G_k(y_k) = -\text{diag}(y_1^k, \dots, y_{b_k}^k).$$

For $k \in K_r$ suppose that

$$\Delta_k = \left\{ \delta_k I_{n_k} : \delta_k \in \mathbb{C}, \begin{pmatrix} \delta_k \\ 1 \end{pmatrix}' P_0^k \begin{pmatrix} \delta_k \\ 1 \end{pmatrix} = 0, \begin{pmatrix} \delta_k \\ 1 \end{pmatrix}' P_j^k \begin{pmatrix} \delta_k \\ 1 \end{pmatrix} \geq 0, \quad j = 1, \dots, b_k \right\}$$

with fixed Hermitian matrices

$$P_j^k = \begin{pmatrix} q_j^k & s_j^k \\ (s_j^k)' & r_j^k \end{pmatrix} \in \mathbb{C}^{2 \times 2} \text{ satisfying } q_j^k r_j^k - |s_j^k|^2 < 0 \text{ or } P_j^k = 0.$$

Lemma A.1 implies that $\delta_k \in \mathbb{C}$ is contained in a finite intersection of nontrivial disks and half-planes for $P_0^k = 0$ and that it is additionally constrained to some line or circle for $P_0^k \neq 0$. It is elementary to verify that (2.8) holds for $y_k = (Y_0^k, Y_1^k, \dots, Y_{b_k}^k)$ with Hermitian $Y_0^k, Y_1^k, \dots, Y_{b_k}^k$ and

$$(2.9) \quad H_k(y_k) = \sum_{j=0}^{b_k} \begin{pmatrix} q_j^k Y_j^k & s_j^k Y_j^k \\ (s_j^k)' Y_j^k & r_j^k Y_j^k \end{pmatrix}, \quad G_k(y_k) = -\text{diag}(Y_1^k, \dots, Y_{b_k}^k),$$

just because

$$\begin{pmatrix} \delta_k I \\ I \end{pmatrix}' \begin{pmatrix} q_j^k Y_j^k & s_j^k Y_j^k \\ (s_j^k)' Y_j^k & r_j^k Y_j^k \end{pmatrix} \begin{pmatrix} \delta_k I \\ I \end{pmatrix} = Y_j^k \begin{pmatrix} \delta_k \\ 1 \end{pmatrix}' \begin{pmatrix} q_j^k & s_j^k \\ (s_j^k)' & r_j^k \end{pmatrix} \begin{pmatrix} \delta_k \\ 1 \end{pmatrix}.$$

Observe that Δ_k has an LMI representation if $q_0^k = 0$ (whenever $P_0^k \neq 0$) and $q_j^k \leq 0$ for all $j = 1, \dots, b_k$, and it is compact if strict inequality holds for at least one index. For $k \in K_h$, we finally assume that Δ_k is explicitly constrained as

$$\Delta_k \in \Delta_k := \text{con}\{\Delta_1^k, \dots, \Delta_{b_k}^k\}.$$

This includes general polytopes of real or complex full matrices or of structured matrices with scalar elements that are confined to polytopes in real or complex spaces. We now take $y_k = Y^k$ with Hermitian Y^k , $H_k(y_k) = Y^k$, and

$$(2.10) \quad G_k(y_k) = \text{diag} \left(\begin{pmatrix} I \\ 0 \end{pmatrix}' Y^k \begin{pmatrix} I \\ 0 \end{pmatrix}, - \begin{pmatrix} \Delta_j^k \\ I \end{pmatrix}' Y^k \begin{pmatrix} \Delta_j^k \\ I \end{pmatrix}, j = 1, \dots, b_k \right)$$

to fulfill (2.8), as seen by an elementary concavity argument. Clearly, Δ_k is compact and admits an LMI description.

We have actually constructed extension of relaxations that are well known in μ -theory, and we illustrate its additional flexibility with the following concrete example.

Suppose some dynamic interconnection is affected by an uncertainty

$$\Delta(s) = \text{diag}(\Delta_1(s), \delta_2(s), \delta_3 I, \delta_4 I)$$

with a full block Δ_1 satisfying $\|\Delta_1\|_\infty \leq 1$, a complex repeated block δ_2 of gain at most one and being positive real, and two real parametric uncertainties δ_3, δ_4 of absolute value at most one. Viewing each of the last two real block as a complex repeated block in the unit disk intersected with the real axis leads to a relaxation with

$$P^1 = \begin{pmatrix} -I & 0 \\ 0 & I \end{pmatrix}, \quad P_1^2 = P_1^3 = P_1^4 = \begin{pmatrix} -1 & 0 \\ 0 & 1 \end{pmatrix},$$

$$P_2^2 = \begin{pmatrix} 0 & 1 \\ 1 & 0 \end{pmatrix}, \quad P_0^3 = P_0^4 = \begin{pmatrix} 0 & i \\ -i & 0 \end{pmatrix}$$

implying $y = (y_1, Y_1^2, Y_2^2, Y_0^3, Y_1^3, Y_0^4, Y_1^4)$, $G(y) = -\text{diag}(y_1, Y_1^2, Y_2^2, Y_1^3, Y_1^4)$ and

$$H(y) = \begin{pmatrix} \text{diag}(-y_1 I, -Y_1^2, -Y_1^3, -Y_1^4) & \text{diag}(0, Y_2^2, iY_0^3, iY_0^4) \\ \text{diag}(0, (Y_2^2)', -i(Y_0^3)', -i(Y_0^4)') & \text{diag}(y_1, Y_1^2, Y_1^3, Y_1^4) \end{pmatrix}.$$

Instead, we can combine the last two real repeated blocks as $\Delta_3 = \text{diag}(\delta_3 I, \delta_4 I)$ and work with the relaxation $y = (y_1, Y_1^2, Y_2^2, Y^3)$,

$$G(y) = -\text{diag} \left(y_1, Y_1^2, Y_2^2, - \begin{pmatrix} I \\ 0 \end{pmatrix}' Y^3 \begin{pmatrix} I \\ 0 \end{pmatrix}, \begin{pmatrix} \pm I \\ I \end{pmatrix}' Y^3 \begin{pmatrix} \pm I \\ I \end{pmatrix} \right),$$

$$H(y) = \begin{pmatrix} \text{diag} \left(-y_1 I, -Y_1^2, \begin{pmatrix} I \\ 0 \end{pmatrix}' Y^3 \begin{pmatrix} I \\ 0 \end{pmatrix} \right) & \text{diag} \left(0, Y_2^2, \begin{pmatrix} I \\ 0 \end{pmatrix}' Y^3 \begin{pmatrix} 0 \\ I \end{pmatrix} \right) \\ \text{diag} \left(0, (Y_2^2)', \begin{pmatrix} 0 \\ I \end{pmatrix}' Y^3 \begin{pmatrix} I \\ 0 \end{pmatrix} \right) & \text{diag} \left(y_1, Y_1^2, \begin{pmatrix} 0 \\ I \end{pmatrix}' Y^3 \begin{pmatrix} 0 \\ I \end{pmatrix} \right) \end{pmatrix}.$$

It is easy to check that the latter relaxation is expected to be less conservative than the former since the set of multipliers is larger. This example illustrates how one can combine individual blocks in order to increase the set of multipliers and to potentially improve the relaxation at the expense of a larger LMI system. For example, if the frequency response of $\delta_2(s)$ is contained in some polytope in $\mathbb{C}$, we could view $\text{diag}(\delta_2(s)I, \delta_3I, \delta_4I)$ as one convex hull block, which might improve the relaxation even further. In contrast to standard μ -theory, note that (δ_3, δ_4) can vary in an arbitrary polytope of $\mathbb{R}^2$. An explicit numerical example can be found in [39].

2.3. Robust linear semidefinite programming. In robust linear algebra or in control, decision variables typically live in some real subspace $\mathbf{X}$ of structured matrices, and one minimizes $\mathcal{C}(X)$ over $\mathcal{L}(X, \delta)' + \mathcal{L}(X, \delta) < 0$ for all $\delta \in \boldsymbol{\delta}$, where $\mathcal{L}(\cdot, \delta)$ and $\mathcal{C}(\cdot)$ are matrix- and scalar-valued affine functions while $\mathcal{L}(X, \cdot)$ is rational without pole in zero. It is possible to construct a linear fractional representation

$$\mathcal{L}(X, \delta) = \Delta(\delta) \star \begin{pmatrix} A & B \\ C(X) & D(X) \end{pmatrix} := D(X) + C(X)\Delta(\delta)(I - A\Delta(\delta))^{-1}B$$

with fixed A, B , affine $C(X), D(X)$, and linear $\Delta(\delta)$. With $\boldsymbol{\Delta} = \{\Delta(\delta) : \delta \in \boldsymbol{\delta}\}$, we have to minimize $\mathcal{C}(X)$ over $X \in \mathbf{X}$ such that

$$\begin{pmatrix} F(\Delta) \\ I \end{pmatrix}' \begin{pmatrix} 0 & C(X)' \\ C(X) & D(X) + D(X)' \end{pmatrix} \begin{pmatrix} F(\Delta) \\ I \end{pmatrix} < 0 \quad \text{for all } \Delta \in \boldsymbol{\Delta},$$

and the corresponding relaxation involves the constraints

$$G(y) < 0, \quad \begin{pmatrix} I & 0 \\ A & B \end{pmatrix}' H(y) \begin{pmatrix} I & 0 \\ A & B \end{pmatrix} + \begin{pmatrix} 0 & C(X)' \\ C(X) & D(X) + D(X)' \end{pmatrix} < 0.$$

This clarifies why our seemingly particular formulation of (ROB) widely generalizes the scenarios for linear semidefinite programming, conic quadratic programming, or standard linear programming from the usually considered affine to possibly polynomial or even rational dependence on the uncertain parameters [3]. This little section is included to stress that we can exploit a particular structural dependence on X within the relaxation in order to arrive at computationally more efficient implementations than those based on a translation into the standard form (1.1). In contrast to [13], our approach allows full flexibility in the choice of the employed class of multipliers. Finally, if the data matrices depend affinely on the uncertainties, one can choose $A = 0$ in the linear fractional representation, and then it is well known how to keep the size of $\Delta(\delta)$ minimal [46].

2.4. Robust least-squares. Among the numerous practically interesting instances of the latter scenario, we would like to briefly address a matrix version of the classical least-squares approximation problem for the purpose of illustration. The nominal least-squares problem can be formulated as

$$\min_X \|XM - N\|_F^2,$$

where M, N are data matrices and $\|\cdot\|_F$ denotes the Frobenius norm. If M, N have l columns and $e_1, \dots, e_l$ denote the standard unit vector vectors, this is equivalent to

infimizing γ over all (γ, X) satisfying the LMI constraint

$$(2.11) \quad \begin{pmatrix} -\gamma & (XM e_1 - N e_1)' & \cdots & (XM e_l - N e_l)' \\ XM e_1 - N e_1 & -\gamma I & & 0 \\ \vdots & & \ddots & \\ XM e_l - N e_l & 0 & & -\gamma I \end{pmatrix} < 0.$$

If the data matrices $M(\delta)$, $N(\delta)$ depend rationally (without pole in zero) on the uncertain parameter $\delta \in \mathcal{D}$, the robust counterpart requires us to minimize the worst-case residual error

$$\sup_{\delta \in \mathcal{D}} \|XM(\delta) - N(\delta)\|_F^2.$$

This problem is subsumed to (ROB) by redescribing the robust counterpart of (2.11) as in the previous section with a linear fractional representation

$$\begin{pmatrix} M(\delta)e_1 \\ N(\delta)e_1 \\ \vdots \\ M(\delta)e_l \\ N(\delta)e_l \end{pmatrix} = \Delta(\delta) \star \left(\begin{array}{c|c} A & B_0 \\ \hline C_1^M & D_1^M \\ C_1^N & D_1^N \\ \vdots & \vdots \\ C_l^M & D_l^M \\ C_l^N & D_l^N \end{array} \right).$$

Obviously, the same technique applies if replacing $\|\cdot\|_F^2$ with other functionals whose sublevel sets admit LMI representations.

2.5. Inclusion of nominal and multiple robustness constraints. If it is desired to include the nominal LMI constraint $U(x) < 0$ (with Hermitian-valued affine $U(x)$), one just needs to replace $F(\Delta)$ and $W(x)$ by $F_e(\Delta) = (F(\Delta) \ 0)$ and $\text{diag}(W(x), U(x))$, respectively. Similarly, multiple robustness constraints

$$\begin{pmatrix} F_j(\Delta) \\ I \end{pmatrix}' W_j(x) \begin{pmatrix} F_j(\Delta) \\ I \end{pmatrix} < 0 \quad \text{for all } \Delta \in \mathbf{\Delta}, \quad j = 1, \dots, l,$$

can be easily combined into one single constraint by standard diagonal augmentation. Therefore multiple nominal or robustness constraints can be assumed to be already incorporated in the formulation of (ROB).

3. Exactness tests. Our main result on the exactness of the relaxation (REL) is expressed in terms of its Lagrange dual. Let us hence start by explicitly formulating these duals [1] for the two situations that (REL) is infeasible and feasible, where G^* , H^* denote the adjoint mappings of G , H , respectively.

LEMMA 3.1.

$$(1) \quad \begin{array}{l} M \neq 0 \quad N \neq 0 \\ M \geq 0 \quad N \geq 0, \quad \langle W_0, M \rangle \geq 0 \quad \langle W_j, M \rangle = 0 \quad j = 1, \dots, n \end{array}$$

$$(3.1) \quad H^* \left(\begin{pmatrix} I & 0 \\ A & B \end{pmatrix} M \begin{pmatrix} I & 0 \\ A & B \end{pmatrix}' \right) + G^*(N) = 0.$$

$$(2) \quad \begin{array}{l} \gamma_{rel} \geq 0 \\ N \geq 0, \quad \langle W_j, M \rangle + c_j = 0 \quad j = 1, \dots, n \end{array} \quad (3.1)$$

The infeasibility characterization for (REL) is obtained by Lagrange dualization of the following (obviously feasible) linear SDP:

$$\inf \left\{ t \in \mathbb{R} : G(y) < 0, \begin{pmatrix} I & 0 \\ A & B \end{pmatrix}' H(y) \begin{pmatrix} I & 0 \\ A & B \end{pmatrix} + W(x) - tI < 0 \right\}.$$

Therefore it is possible to determine infeasibility certificates or dual optimal solutions by primal-dual LMI algorithms. $\square$

We are now ready to formulate one of the main results of this paper, a novel, necessary, and sufficient condition for the existence of a worst-case perturbation and the absence of a relaxation gap, formulated in terms of dual optimal solutions or infeasibility certificates of (REL) as appearing in Lemma 3.1.

THEOREM 3.2.

$$(1) \quad \Delta_0 \in \Delta$$

$$(3.2) \quad L(x, \Delta_0) < 0$$

$$\Delta_0 \in \Delta \quad (3.2) \quad M \neq 0 \quad N$$

$$(3.3) \quad \begin{pmatrix} I & -\Delta_0 \end{pmatrix} \begin{pmatrix} I & 0 \\ A & B \end{pmatrix} M = 0.$$

$$(2) \quad \Delta_0 \in \Delta \quad (3.3) \quad (M, N)$$

$$(3.4) \quad \inf \{ \langle c, x \rangle : L(x, \Delta_0) < 0 \} = \gamma_{opt} \quad \gamma_{opt} = \gamma_{rel}.$$

$$\Delta_0 \in \Delta \quad (3.4) \quad (M, N)$$

Let us first prove (1). The first statement is obvious from the definition of (ROB). Observe that, for any fixed $\Delta_0 \in \Delta$, (3.3) is equivalent to $(I - \Delta_0 A - \Delta_0 B)M = 0$ or (since $I - A\Delta_0$ is nonsingular by assumption (A)) to

$$(3.5) \quad \begin{pmatrix} I & -F(\Delta_0) \end{pmatrix} M = 0.$$

Let us now assume that $\Delta_0 \in \Delta$ satisfies (3.3) for some infeasibility certificate $M \neq 0$, N of (REL). Due to (3.5) and since $M \geq 0$ does not vanish, there exists some nonzero $Z \geq 0$ such that

$$(3.6) \quad M = \begin{pmatrix} F(\Delta_0) \\ I \end{pmatrix} Z \begin{pmatrix} F(\Delta_0) \\ I \end{pmatrix}'.$$

Since

$$\left\langle \begin{pmatrix} F(\Delta_0) \\ I \end{pmatrix}' W_j \begin{pmatrix} F(\Delta_0) \\ I \end{pmatrix}, Z \right\rangle = \langle W_j, M \rangle \quad \text{for } j = 0, 1, \dots, n,$$

we infer for all $j = 1, \dots, n$ that

$$(3.7) \quad \left\langle \begin{pmatrix} F(\Delta_0) \\ I \end{pmatrix}' W_0 \begin{pmatrix} F(\Delta_0) \\ I \end{pmatrix}, Z \right\rangle \geq 0, \quad \left\langle \begin{pmatrix} F(\Delta_0) \\ I \end{pmatrix}' W_j \begin{pmatrix} F(\Delta_0) \\ I \end{pmatrix}, Z \right\rangle = 0.$$

Therefore Z is an infeasibility certificate for the LMI $L(x, \Delta_0) < 0$, which proves (3.2).

Let us now assume that (3.2) holds. Then there exists a nonzero infeasibility certificate $Z \geq 0$ satisfying (3.7). If we define M by (3.6), this matrix obviously satisfies (3.5), and hence also (3.3). To finish the proof it remains to construct some N with (3.1). With the abbreviation

$$J(Z) = \begin{pmatrix} I & 0 \\ A & B \end{pmatrix} \begin{pmatrix} F(\Delta_0) \\ I \end{pmatrix} Z \begin{pmatrix} F(\Delta_0) \\ I \end{pmatrix}' \begin{pmatrix} I & 0 \\ A & B \end{pmatrix}',$$

we need to show the existence of $N \geq 0$ with $H^*(J(Z)) + G^*(N) = 0$. Without any constraint on N , the equation is solvable iff $\text{im}(H^* \circ J) \subset \text{im}(G^*)$, which is equivalent to the fact that $G(y) = 0$ implies $J^* \circ H(y) = J^*(H(y)) = 0$. To show that this is indeed true we choose y with $G(y) = 0$ and hence $G(\pm y) \leq 0$ by linearity. Due to (2.2) we infer $\pm \text{col}(\Delta_0, I)' H(y) \text{col}(\Delta_0, I) = \text{col}(\Delta_0, I)' H(\pm y) \text{col}(\Delta_0, I) \geq 0$, and hence $\text{col}(\Delta_0, I)' H(y) \text{col}(\Delta_0, I) = 0$. Again due to (2.5) we have

$$J(Z) = \begin{pmatrix} \Delta_0 \\ I \end{pmatrix} (I - A\Delta_0)^{-1} B Z B' [(I - A\Delta_0)^{-1}]' \begin{pmatrix} \Delta_0 \\ I \end{pmatrix}',$$

and therefore

$$(3.8) \quad J^*(H(y)) = B' [(I - A\Delta_0)^{-1}]' \begin{pmatrix} \Delta_0 \\ I \end{pmatrix}' H(y) \begin{pmatrix} \Delta_0 \\ I \end{pmatrix} (I - A\Delta_0)^{-1} B = 0.$$

Finally, we show that there even exists some $N \geq 0$ with $H^*(J(Z)) + G^*(N) = 0$. This is true if $\inf\{t \in \mathbb{R} : H^*(J(Z)) + G^*(N) = 0, N + tI \geq 0\}$ is attained and is nonpositive. As just seen this program is strictly feasible. Its dual is to maximize $-\langle H(y), J(Z) \rangle$ over y with $G(y) \leq 0$ and $\langle G(y), I \rangle = -1$. Since this dual is as well strictly feasible by (2.3), the primal optimal value is attained. Moreover, $G(y) \leq 0$ implies $-\langle H(y), J(Z) \rangle = -\langle J^*(H(y)), Z \rangle \leq 0$ due to $Z \geq 0$, (2.2), and (3.8). Therefore the optimal value of the dual, and hence that of the primal, is nonpositive.

Let us now turn to the proof of (2). From the proof of Lemma 2.1 we can extract the following important insight: If (x, y) is feasible for (REL), then x is feasible for (ROB), and for any fixed $\Delta \in \mathbf{\Delta}$ the following chain of inequalities holds true:

$$(3.9) \quad \inf\{\langle c, x \rangle : L(x, \Delta) < 0\} = \inf\{\langle c, x \rangle : L(x, \Delta) \leq 0\} \leq \gamma_{\text{opt}} \leq \gamma_{\text{rel}}.$$

Let us now assume that $\Delta_0 \in \mathbf{\Delta}$ satisfies (3.3) for some optimal solution (M, N) of the dual of (REL). If we simply follow the steps in the proof of (1), we infer that there exists some $Z \geq 0$ with

$$(3.10) \quad \left\langle \begin{pmatrix} F(\Delta_0) \\ I \end{pmatrix}' W_0 \begin{pmatrix} F(\Delta_0) \\ I \end{pmatrix}, Z \right\rangle = \gamma_{\text{rel}}, \quad \left\langle \begin{pmatrix} F(\Delta_0) \\ I \end{pmatrix}' W_j \begin{pmatrix} F(\Delta_0) \\ I \end{pmatrix}, Z \right\rangle + c_j = 0$$

for $j = 1, \dots, n$. By standard weak duality we infer $\inf\{\langle c, x \rangle : L(x, \Delta_0) < 0\} \geq \gamma_{\text{rel}}$. Due to (3.9) for $\Delta = \Delta_0$ we arrive at (3.4).

Let us now assume that (3.4) holds for some $\Delta_0 \in \mathbf{\Delta}$. If we dualize $\inf\{\langle c, x \rangle : L(x, \Delta_0) < 0\}$ (without duality gap due to strict feasibility), we conclude that there exists some $Z \geq 0$ satisfying (3.10). As in the proof of part (1), we define M by (3.6) and observe that it satisfies (3.5), and hence also (3.3). Moreover, we have seen

that there exists some $N \geq 0$ with $H^*(J(Z)) + G^*(N) = 0$. Due to (3.10) we infer $\langle W_0, M \rangle = \gamma_{\text{rel}}$ and $\langle W_j, M \rangle + c_j = 0$ for $j = 1, \dots, n$. All this proves that (M, N) is an optimal solution of (REL)'s dual. $\square$

Theorem 3.2 is used in practice by just computing an optimal solution of the relaxation's dual and by verifying the existence of some $\Delta_0 \in \Delta$ with (3.3). As seen in our examples it is typically not restrictive to assume that Δ admits an LMI representation, and then this test just amounts to solving a linear SDP. Under an additional hypothesis on the relaxation, one can circumvent this numerical verification and verify exactness by a direct inspection of M , which leads us to the following so-called rank-one principle.

COROLLARY 3.3. *Let (M, N) be a dual optimal solution for (REL) with $M = mm'$ for some vector m . Partition $m = \text{col}(w, \xi)$ and define $z = Aw + B\xi$. Then (3.1) implies $H^*(\text{col}(w, z)\text{col}(w, z)') + G^*(N) = 0$. For all y with $G(y) \leq 0$ we infer $\text{col}(w, z)'H(y)\text{col}(w, z) = \langle y, H^*(\text{col}(w, z)\text{col}(w, z)') \rangle = -\langle y, G^*(N) \rangle = -\langle G(y), N \rangle \geq 0$. By hypothesis there exists $\Delta_0 \in \Delta$ with $w = \Delta_0 z = \Delta_0(Aw + B\xi)$, and therefore*

$$\begin{pmatrix} w \\ z \end{pmatrix}' H(y) \begin{pmatrix} w \\ z \end{pmatrix} \geq 0, \quad \text{whenever } G(y) \leq 0, \quad \text{whenever } w = \Delta z, \quad \Delta \in \Delta.$$

$$\Delta_0 \in \Delta, \quad (3.2) \quad \text{whenever } M = mm', \quad \text{whenever } M = mm', \quad \Delta_0 \in \Delta, \quad (3.4)$$

Let (M, N) denote either an infeasibility certificate or a dual optimal solution for (REL) with $M = mm'$ for some vector m . Partition $m = \text{col}(w, \xi)$ and define $z = Aw + B\xi$. Then (3.1) implies $H^*(\text{col}(w, z)\text{col}(w, z)') + G^*(N) = 0$. For all y with $G(y) \leq 0$ we infer $\text{col}(w, z)'H(y)\text{col}(w, z) = \langle y, H^*(\text{col}(w, z)\text{col}(w, z)') \rangle = -\langle y, G^*(N) \rangle = -\langle G(y), N \rangle \geq 0$. By hypothesis there exists $\Delta_0 \in \Delta$ with $w = \Delta_0 z = \Delta_0(Aw + B\xi)$, and therefore

$$\begin{pmatrix} I & -\Delta_0 \end{pmatrix} \begin{pmatrix} I & 0 \\ A & B \end{pmatrix} M = \begin{pmatrix} I & -\Delta_0 \end{pmatrix} \begin{pmatrix} w \\ Aw + B\xi \end{pmatrix} \begin{pmatrix} w \\ \xi \end{pmatrix}' = 0.$$

Hence Δ_0 satisfies (3.3), and the proof is finished due to Theorem 3.2. $\square$

On one hand, Corollary 3.3 relates the exactness test in Theorem 3.2 to a variety of other rank-one principles in the literature [29, 28, 20, 30, 27]. On the other hand, we conclude that the rank-one principle can hold only if the robust LMI problem under consideration admits a worst-case perturbation. It is believed to be a particularly appealing feature of the given tests that they are formulated and proved within one common framework for a whole family of relaxations parameterized by G, H , capturing the full variety of examples as in section 2 and many more. It is finally relevant to stress that Theorem 3.2 is indeed more general than Corollary 3.3, as illustrated by means of the following example.

EXAMPLE 3.4. Consider the problem (ROB) of infimizing $x \in \mathbb{R}$ such that

$$\begin{pmatrix} 1 - \delta_1^2 - \delta_2^2 & 0 \\ 0 & 1/(1 + \delta_1^2 + \delta_2^2) \end{pmatrix} < xI_2 \quad \text{for all } (\delta_1, \delta_2) \in [-1, 1] \times [-1, 1].$$

If we solve the convex hull block relaxation as defined in section 2.2, we obtain the optimal value $\gamma_{\text{rel}} = 1$, which obviously equals $\gamma_{\text{opt}} = 1$. The computed M has rank two (such that we cannot confirm exactness with Corollary 3.3), while solving (3.3) leads to the correct worst-case uncertainty $(\delta_1, \delta_2) = (0, 0)$.

4. On the verification of Assumption (A). It is rather typical in concrete problems, as for our examples in section 2, that the left-upper block of $W(x)$ is positive semidefinite for all first components x of (REL)-feasible points (x, y) (or even for all

$x \in \mathbb{R}^n$). The following result indicates that Assumption (A) is then automatically guaranteed by $\gamma_{\text{rel}} < \infty$.

LEMMA 4.1. Fix any $\Delta \in \Delta$; suppose $(I - A\Delta)z = 0$ for $z \neq 0$; define $w = \Delta z$ to get $z = Aw$, $w \neq 0$; left-, right-multiply the second inequality in (2.4) with $\text{col}(w, 0)'$, $\text{col}(w, 0)$ to infer with the extra hypothesis on $W(x)$ that $\text{col}(w, z)'H(y)\text{col}(w, z) < 0$; on the other hand, left-, right-multiply the second inequality in (2.2) with z' , z to infer that $\text{col}(w, z)'H(y)\text{col}(w, z) \geq 0$, a contradiction. $\square$

A direct verification of Assumption (A) can be based on the feasibility problem

$$(4.1) \quad G(y) < 0, \quad \begin{pmatrix} I \\ A \end{pmatrix}' H(y) \begin{pmatrix} I \\ A \end{pmatrix} < 0.$$

By standard duality, (4.1) is infeasible iff there exist $M \neq 0$, N with

$$M \geq 0 \quad N \geq 0, \quad H^* \left(\begin{pmatrix} I \\ A \end{pmatrix}' M \begin{pmatrix} I \\ A \end{pmatrix} \right) + G^*(N) = 0.$$

COROLLARY 4.2. “If” is obvious. For the other direction assume that Assumption (A) does not hold, and choose $\Delta_0 \in \Delta$ and a vector $w \neq 0$ with $(I - \Delta_0 A)w = 0$. With $Z = ww'$ and $J(Z) = \text{col}(I, A)Z\text{col}(I, A)'$ it suffices to show that there exists some $N \geq 0$ with $H^*(J(Z)) + G^*(N) = 0$. The proof is completed as in the proof of Theorem 3.2 since $w = \Delta_0 Aw$, and therefore $J(Z) = \text{col}(\Delta_0, I)(Aw)(Aw)'\text{col}(\Delta_0, I)'$. $\square$

In summary, if (4.1) is feasible, then (A) holds. If (4.1) is not feasible, one computes an infeasibility certificate and tries to solve $(I - \Delta_0 A)M = 0$ for some $\Delta_0 \in \Delta$. If such a Δ_0 exists, then Assumption (A) is not true, while Assumption (A) remains undecided if the equation cannot be solved. If the relaxation satisfies the hypothesis in Corollary 3.3, the existence of an infeasibility certificate with M of rank one guarantees that Assumption (A) does not hold. All this indicates how one can numerically verify Assumption (A) by solving a phase I problem.

5. Robust performance analysis. In this section we intend to address in somewhat more detail the particularly relevant block structure and the corresponding relaxation (REL) of section 2.2.

For each of the full blocks $k \in K_f$ let us suppose that only one constraint is imposed ($b_k = 1$), where P_k^1 is assumed to be nonsingular, to have a negative definite left-upper block and a positive semidefinite right-lower block.

Moreover, let us extend Δ to Δ^∞ by allowing for each unbounded repeated block $\Delta_k = \delta_k I$, $k \in K_r$, as well the value $\delta_k = \infty$. We need to modify Assumption (A) and the definition of $F(\cdot)$ as follows. Suppose that $K_\infty \subset \{k \in K_r : q_k^j \geq 0, j = 1, \dots, b_k\}$. If $\Delta \in \Delta$ satisfies $\Delta_k = \delta_k I \neq 0$ for all $k \in K_\infty$, observe that $\Delta = \Psi_{K_\infty}(\Delta)^{-1} \Phi_{K_\infty}(\Delta)$ with

$$\Phi_{K_\infty}(\Delta) = \sum_{k \in K \setminus K_\infty} E_k \Delta_k F_k' + \sum_{k \in K_\infty} E_k F_k' \quad \text{and} \quad \Psi_{K_\infty}(\Delta) = \sum_{k \in K \setminus K_\infty} E_k E_k' + \sum_{k \in K_\infty} \frac{1}{\delta_k} E_k E_k'.$$

This allows us to rerepresent $F(\Delta) = \Delta(I - A\Delta)^{-1}B = (I - \Delta A)^{-1}\Delta B$ as

$$(5.1) \quad F(\Delta) = (\Psi_{K_\infty}(\Delta) - \Phi_{K_\infty}(\Delta)A)^{-1} \Phi_{K_\infty}(\Delta)B.$$

As the crucial point, $\Phi_{K_\infty}(\Delta)$ and $\Psi_{K_\infty}(\Delta)$ can be continuously extended (with finite complex values) to $\delta_k = \infty$ for $k \in K_\infty$. For such Δ , Assumption (A) holds if $\Psi_{K_\infty}(\Delta) - \Phi_{K_\infty}(\Delta)A$ is nonsingular, and $F(\Delta)$ is defined by (5.1). We comment on the verification of this extended version of Assumption (A) at the end of section 5.1.

5.1. Rank-one principle. Let us now reveal that the rank-one exactness principle does apply, which extends available results for the structured singular value [29, 28, 6, 31] to more general descriptions of full blocks, to multiple constraints on repeated blocks (with intersections of possibly unbounded circles, disks or lines, and half-spaces as value-sets), and to convex hull blocks.

COROLLARY 5.1. $\Delta_0 \in \Delta_\infty$ (3.2) $\Delta_0 \in \Delta_\infty$ (3.4)

It is straightforward to calculate G^* and H^* . Introduce the abbreviations $U_k = (E'_k \ 0)$, $V_k = (F'_k A \ F'_k B)$. For arbitrary Hermitian M it is then easy to verify that (3.1) holds for some $N \geq 0$ iff there exist $\hat{N}_k \geq 0$ and $N_j^k \geq 0$ with

$$(5.2) \quad \left\langle P^k, \begin{pmatrix} U_k \\ V_k \end{pmatrix} M \begin{pmatrix} U_k \\ V_k \end{pmatrix}' \right\rangle \geq 0, \quad k \in K_f,$$

$$(5.3) \quad \begin{pmatrix} U_k & V_k \end{pmatrix} \begin{pmatrix} q_j^k M & (s_j^k)' M \\ s_j^k M & r_j^k M \end{pmatrix} \begin{pmatrix} U'_k \\ V'_k \end{pmatrix} \begin{cases} = 0 & \text{for } k \in K_r, \ j = 0, \\ \geq 0 & \text{for } k \in K_r, \ j = 1, \dots, b_k, \end{cases}$$

$$(5.4) \quad \begin{pmatrix} U_k \\ V_k \end{pmatrix} M \begin{pmatrix} U_k \\ V_k \end{pmatrix}' + \begin{pmatrix} I \\ 0 \end{pmatrix} \hat{N}_k \begin{pmatrix} I \\ 0 \end{pmatrix}' - \sum_{j=1}^{b_k} \begin{pmatrix} \Delta_j^k \\ I \end{pmatrix} N_j^k \begin{pmatrix} \Delta_j^k \\ I \end{pmatrix}' = 0, \quad k \in K_h.$$

Since M has rank one, $M = mm'$ for $m \neq 0$. Partition $m = \text{col}(w, \xi)$, define $z = (A \ B)m = Aw + B\xi$, $w_k = E'_k w$, $z_k = F'_k z$, and note that $\text{col}(U_k, V_k)M\text{col}(U_k, V_k)' = \text{col}(w_k, z_k)\text{col}(w_k, z_k)'$.

For $k \in K_f$ we infer from (5.2) that $\text{col}(w_k, z_k)' P_1^k \text{col}(w_k, z_k) \geq 0$, and Lemma A.2 implies (due to our hypotheses on P_1^k) that $w_k = \Delta_k z_k$ for some $\Delta_k \in \Delta_k$.

For $k \in K_r$ we obtain from (5.3) that

$$\begin{pmatrix} w'_k \\ z'_k \end{pmatrix}' \begin{pmatrix} q_0^k & (s_0^k)' \\ s_0^k & r_0^k \end{pmatrix} \begin{pmatrix} w'_k \\ z'_k \end{pmatrix} = 0,$$

$$\begin{pmatrix} w'_k \\ z'_k \end{pmatrix}' \begin{pmatrix} q_j^k & (s_j^k)' \\ s_j^k & r_j^k \end{pmatrix} \begin{pmatrix} w'_k \\ z'_k \end{pmatrix} \geq 0, \quad j = 1, \dots, b_k.$$

If $z_k \neq 0$, we can apply Lemma A.3 to see that $w_k = z_k \delta_k = [\delta_k I] z_k = \Delta_k z_k$ with $\Delta_k \in \Delta_k$. If $z_k = 0$ and $w_k = 0$, we can choose any $\Delta_k \in \Delta_k$ with $z_k = \Delta_k w_k$. If $z_k = 0$ and $w_k \neq 0$, we infer $q_0^k = 0$, $q_j^k \geq 0$ for all $j = 1, \dots, b_k$; then Δ_k is unbounded, and we set $\delta_k = \infty$. Let us collect all $k \in K_r$ with $\delta_k = \infty$ in K_∞ .

For $k \in K_h$, (5.4) implies

$$w_k w'_k + \hat{N}_k - \sum_{j=1}^{b_k} \Delta_j^k N_j^k (\Delta_j^k)' = 0, \quad w_k z'_k z_k - \sum_{j=1}^{b_k} \Delta_j^k N_j^k z_k = 0, \quad z_k z'_k - \sum_{j=1}^{b_k} N_j^k = 0.$$

Due to the third relation, $\ker(z'_k) \subset \ker(N_j^k)$ for all j such that there exist α_j^k with $N_j^k = z_k \alpha_j^k z'_k$. If $z_k \neq 0$, we have $\alpha_j^k \geq 0$ and $\sum_{j=1}^{b_k} \alpha_j^k = 1$, and we can divide by $z'_k z_k$

in the second relation to get $w_k = \sum_{j=1}^{b_k} \Delta_j^k z_k \alpha_j^k = \Delta_k z_k$ for $\Delta_k = \sum_{j=1}^{b_k} \alpha_j^k \Delta_j^k \in \Delta_k$. If $z_k = 0$, we infer $N_j^k = 0$, and hence $w_k = 0$ from the first relation, which allows us to choose any $\Delta_k \in \Delta_k$ in order to realize $w_k = \Delta_k z_k$.

Collect all subblocks in $\Delta_0 \in \Delta^\infty$. Observe that $\Psi_{K_\infty}(\Delta_0)w = \Phi_{K_\infty}(\Delta_0)z$, and hence $(\Psi_{K_\infty}(\Delta_0) - \Phi_{K_\infty}(\Delta_0)A)w = \Phi_{K_\infty}(\Delta_0)B\xi$ or $w - F(\Delta_0)\xi = 0$ (using the extended Assumption (A)) and therefore (3.5). The proof is completed in exactly the same fashion as the proof of Theorem 3.2. $\square$

It is remarkable that the verification of the extended version of Assumption (A) can be again based on checking the feasibility of (4.1) as in section 4.

COROLLARY 5.2. (4.1) Δ^∞ (4.1)

Let (4.1) be feasible. Choose $\Delta \in \Delta^\infty$ and collect all $k \in K_r$ with $\delta_k = \infty$ in K_∞ ; suppose $(\Psi_{K_\infty}(\Delta) - \Phi_{K_\infty}(\Delta)A)w = 0$ for $w \neq 0$; define $z = Aw$ to get $\Psi_{K_\infty}(\Delta)w = \Phi_{K_\infty}(\Delta)z$; with $w_k = E_k'w$, $z_k = F_k'z$ infer $w_k = \Delta_k z_k$ for $k \in K \setminus K_\infty$ and $z_k = 0$ for $k \in K_\infty$; this implies $\text{col}(w_k, z_k)' H_k(y_k) \text{col}(w_k, z_k) \geq 0$ for $k \in K$, and hence $\text{col}(w, z)' H(y) \text{col}(w, z) \geq 0$; left- and right-multiply the second inequality in (4.1) with w' and w to infer $\text{col}(w, z)' H(y) \text{col}(w, z) < 0$, a contradiction. If (4.1) admits an infeasibility certificate with M of rank 1, literally as in the proof of Corollary 5.1, one constructs $\Delta_0 \in \Delta^\infty$ and $w \neq 0$ with $(\Psi_{K_\infty}(\Delta) - \Phi_{K_\infty}(\Delta)A)w = 0$. $\square$

5.2. Simple block structures. Let us now reveal that the relaxations are always exact for so-called simple block structures, which extends and unifies particular results for the structured singular value in [29, 28].

THEOREM 5.3 (two full blocks). $|K_f| = 2$ $|K_r| = 0$ $|K_h| = 0$ $\gamma_{opt} = \gamma_{rel}$

Recall from the proof of Corollary 5.1 and for general M and K with $|K_h| = 0$ that M satisfies (3.1) for some $N \geq 0$ iff M satisfies (5.2)–(5.3).

If $\gamma_{rel} = \infty$, we have to show $\gamma_{opt} = \infty$. Let us assume the contrary and fix $x_0 \in \mathbb{R}^n$, which is feasible for (ROB). Moreover, choose a (REL)-infeasibility certificate (M, N) as in Lemma 3.1. Note that $\langle W(x_0), M \rangle = \langle W_0, M \rangle \geq 0$. Since M satisfies (5.2), Lemma A.6 implies that there exists $m \neq 0$ with $\langle W(x_0), mm' \rangle \geq 0$ such that (5.2) persists to hold for M replaced by mm' . Corollary 5.1, applied for c replaced by 0 and $W(x)$ replaced by $W(x_0)$, implies that there exists some $\Delta_0 \in \Delta^\infty$ with $L(x_0, \Delta_0) \not\leq 0$, a contradiction.

Now suppose $\gamma_{rel} < \infty$ and fix any (ROB)-feasible $x_0 \in \mathbb{R}^n$. If we can show that $\langle c, x_0 \rangle \geq \gamma_{rel}$, we infer that $\gamma_{opt} \geq \gamma_{rel}$, and the proof is complete. Let us assume the contrary $\langle c, x_0 \rangle < \gamma_{rel}$. If (M, N) denotes a dual optimal solution as in Lemma 3.1, we conclude that $\langle W(x_0), M \rangle = \langle W_0, M \rangle - \langle c, x_0 \rangle > 0$. As proved in the first part there exists some $\Delta_0 \in \Delta^\infty$ with $L(x_0, \Delta_0) \not\leq 0$, a contradiction to the choice of x_0 . $\square$

THEOREM 5.4 (one repeated block). $|K_f| = 0$ $K_r = \{1\}$ $|K_h| = 0$

- $q_0^1 r_0^1 - |s_0^1|^2 < 0$ $q_1^1 r_1^1 - |s_1^1|^2 < 0$ $\gamma_{opt} = \gamma_{rel}$
- $(q_0^1, r_0^1, s_0^1) = 0$ $q_1^1 r_1^1 - |s_1^1|^2 < 0$ $W(x_0)$ x_0 $\gamma_{opt} = \gamma_{rel}$

The proof for $q_0^1 r_0^1 - |s_0^1|^2 < 0$ literally follows that of Theorem 5.3 by just relying on Lemma A.7. If $(q_0^1, r_0^1, s_0^1) = 0$, a difficulty arises since Lemma A.7 does

not apply directly. Observe that (5.3) reads as

$$L(M) := \begin{pmatrix} I & 0 \\ A & B \end{pmatrix} \begin{pmatrix} q_1^1 M & (s_1^1)' M \\ s_1^1 M & r_1^1 M \end{pmatrix} \begin{pmatrix} I & 0 \\ A & B \end{pmatrix}' \geq 0.$$

Due to the extended version of Assumption (A), we can apply Lemma A.1 to conclude the existence of a Hermitian X with

$$X \geq 0 \quad \text{with} \quad \begin{pmatrix} I & A \end{pmatrix} \begin{pmatrix} q_1^1 X & (s_1^1)' X \\ s_1^1 X & r_1^1 X \end{pmatrix} \begin{pmatrix} I & A \end{pmatrix}' + L(M) = 0.$$

Then $\hat{M} := M + (I \ 0)' X (I \ 0) \geq M$ actually satisfies $L(\hat{M}) = 0$ by linearity. Moreover, for any (ROB)-feasible x_0 the additional hypothesis on $W(x_0)$ allows us to conclude $\langle W(x_0), \hat{M} \rangle = \langle W(x_0), M \rangle + \langle (I \ 0) W(x_0) (I \ 0)', X \rangle \geq \langle W(x_0), M \rangle$. Hence we can replace M by $\hat{M}$ to complete the proof as in the first case. $\square$

In contrast to [28, 20] we managed to handle possibly unbounded families of disks and circles for one repeated complex block, which leads to a complete common treatment of variants of the KYP Lemma for continuous and discrete time [35, 1]. After the submission of this paper we became aware of an independent proof of the first item in Theorem 5.4 in [25].

Concerning the proof of Theorem 5.3, we would like to stress that the general philosophy of constructing rank-one dual optimal solutions is well established [29, 41] but that the suggested approach via Lemma A.6 is applicable for all dimensions and is completely elementary [14]. Lemma A.6 does lead as well to a straightforward proof of the standard S-procedure, formulated as follows. With Hermitian $P_0, P_1, P_2 \in \mathbb{C}^{n \times n}$ consider the indefinite quadratic program

$$\gamma_1 = \inf \{ x' P_0 x : x \in \mathbb{C}^n, x(1) = 1, x' P_1 x \leq 0, x' P_2 x \leq 0 \}.$$

If there exists some $x_0 \in \mathbb{C}^n$ with $x_0(1) = 1$ and $x_0' P_1 x_0 < 0$, $x_0' P_2 x_0 < 0$, then the value of this program equals that of the linear SDP

$$\gamma_2 = \inf \{ \langle X, P_0 \rangle : X \in \mathbb{C}^{n \times n}, X \geq 0, X(1, 1) = 1, \langle X, P_1 \rangle \leq 0, \langle X, P_2 \rangle \leq 0 \}.$$

If the vector x satisfies $x(1) = 1$, then $X = xx' \geq 0$ and $X(1, 1) = 1$. Therefore $\gamma_1 \geq \gamma_2$. Now suppose $\gamma_1 > \gamma_2$ and choose $\gamma \in (\gamma_2, \gamma_1)$. There exists $X \geq 0$ with $X(1, 1) = 1$, $\langle X, P_1 \rangle \leq 0$, $\langle X, P_2 \rangle \leq 0$, and $\langle X, P_0 \rangle < \gamma$ or $\langle X, P_0 - \gamma e_1 e_1' \rangle < 0$ with the standard first unit vector e_1 . Therefore $\langle X, P_0 - \gamma e_1 e_1' + \delta I \rangle < 0$ for some $\delta > 0$. By Lemma A.6 there exists a vector $x \neq 0$ with $x' P_1 x \leq 0$, $x' P_2 x \leq 0$, and $x'(P_0 - \gamma e_1 e_1' + \delta I)x \leq 0$ or $x' P_0 x < \gamma |x(1)|^2$.

If $x(1) \neq 0$, we take $\tilde{x} = x/x(1)$ to infer $\tilde{x}(1) = 1$ and $\tilde{x}' P_0 \tilde{x} < \gamma$, $\tilde{x}' P_1 \tilde{x} \leq 0$, $\tilde{x}' P_2 \tilde{x} \leq 0$, which in turn implies $\gamma_1 < \gamma$, a contradiction.

If $x(1) = 0$, we choose $\xi \in \mathbb{C}$ with $|\xi| = 1$ such that $\operatorname{Re}(\xi x_0' P_j x) \leq 0$ for $j = 1, 2$. Then $x_t = x_0 + t\xi x$ satisfies $x_t(1) = 1$ and $x_t' P_j x_t = x_0' P_j x_0 + 2t \operatorname{Re}(\xi x_0' P_j x) + t^2 x' P_j x$. For $j = 1, 2$ and all $t \geq 0$ we conclude $x_t' P_j x_t < 0$. For $j = 0$ we exploit $x' P_0 x < 0$ to infer $\lim_{t \rightarrow \infty} x_t' P_0 x_t = -\infty$, and hence $\gamma_1 = -\infty$, a contradiction. $\square$

6. Computationally verifiable exactness. The purpose of this section is to investigate under which conditions the exactness test (2) of Theorem 3.2 is indeed applicable in practice. For this purpose we assume that Δ is compact and admits an LMI description. Moreover, let us suppose that (ROB) is feasible and that the (obviously closed and convex) set

$$X = \{x \in \mathbb{R}^n : L(x, \Delta) \leq 0 \text{ for all } \Delta \in \Delta\}$$

is bounded, and hence compact. In practice one would always impose a priori bounds on x as described in section 2.5 such that the latter hypothesis is not restrictive. As a consequence, $\langle c, x \rangle$ attains its minimum on X , and the set of minimizers X_{opt} is convex and compact. To avoid trivialities suppose $c \neq 0$.

Let us now consider a relaxation (REL) which is feasible, and denote by $\mathcal{M}$ the set of all matrices M such that (M, N) is optimal for the relaxation's dual. In order to verify exactness, one computes some element $M \in \mathcal{M}$ and determines

$$(6.1) \quad \nu_{\text{rel}}(M) = \inf_{\Delta \in \Delta} \left\| \begin{pmatrix} I & -\Delta \end{pmatrix} \begin{pmatrix} I & 0 \\ A & B \end{pmatrix} M \right\|.$$

Since $M \geq 0$ and since Δ has an LMI description, the computation of $\nu_{\text{rel}}(M)$ can be easily translated into a linear SDP. If $\nu_{\text{rel}}(M)$ vanishes, the relaxation is exact, and any minimizer defines a worst-case uncertainty as in Theorem 3.2. If $\nu_{\text{rel}}(M) \leq \nu_0$ for some small ν_0 , we can actually infer approximate exactness, as made precise with explicit error bounds in the following result whose proof can be found in Appendix B.

THEOREM 6.1. *Let $\Delta \in \Delta$ and $\nu_0 \geq 0$. Suppose $\nu_{\text{rel}}(M) \leq \nu_0$. Then, for any $x \in \mathbb{R}^n$ such that $\|x\|_1 \leq \xi$, we have $\langle c, x \rangle \geq \gamma_{\text{opt}} - \nu_0(1 + \xi)r$.*

$$(6.2) \quad \gamma_{\text{opt}} \leq \gamma_{\text{rel}} \leq \gamma_{\text{opt}} + \nu_0(1 + \xi)r.$$

$$\|x\|_1 \leq \eta, \quad x \in \mathbb{R}^n, \quad L(x, \Delta_0) < 0$$

$$(6.3) \quad \gamma_{\text{rel}} - \nu_0(1 + \eta)r \leq \inf\{\langle c, x \rangle : L(x, \Delta_0) < 0\} \leq \gamma_{\text{opt}}.$$

Note that r and ξ are solely determined by (ROB), and (rough) estimates can be obtained by solving (rough) relaxations. In fact Theorem 6.1 can be viewed as an approximate version of (2) in Theorem 3.2 for small ν_0 , and the latter is recovered with $\nu_0 = 0$. Finally, observe that $\nu_{\text{rel}}(M)$ is continuous in M since Δ is compact. Hence, if $\nu_{\text{rel}}(M)$ vanishes or is small, it is guaranteed that $\nu_{\text{rel}}(\tilde{M})$ is small as well for all $\tilde{M}$ close to M , which guarantees stability of the suggested approximate exactness test against small errors in the computation of M . $\square$

From now on let us fix (existence assumed) a worst-case uncertainty $\Delta_0 \in \Delta$ which satisfies $\gamma_{\text{opt}} = \inf\{\langle c, x \rangle : L(x, \Delta_0) < 0\}$. Moreover, recall that (3.3) is equivalent to (3.5) or to

$$(6.4) \quad \langle \hat{W}, M \rangle = 0 \quad \text{if defining} \quad \hat{W} := \begin{pmatrix} I \\ -F(\Delta_0)' \end{pmatrix} \begin{pmatrix} I & -F(\Delta_0) \end{pmatrix} \geq 0.$$

The remaining discussion in this section is devoted to the difficulty that (6.4) might not be valid for some computed $M \in \mathcal{M}$, even if the relaxation is exact. This is indeed relevant since SDP solvers of a different nature return different points in $\mathcal{M}$, and $\mathcal{M}$ is in general not a singleton. This motivates the concept of verifiable exactness, which implies that we can check exactness irrespective of which particular element of $\mathcal{M}$ is computed.

(REL) is verifiably exact if $\langle \hat{W}, M \rangle = 0$ for all $M \in \mathcal{M}$.

If (REL)'s dual is strictly feasible, central-path-following primal-dual interior-point algorithms determine an element $\hat{M} \in \mathcal{M}$ satisfying $\text{im}(M) \subset \text{im}(\hat{M})$ for all $M \in \mathcal{M}$ [18]. Therefore (6.4) holds for $\hat{M}$ iff it holds for all $M \in \mathcal{M}$. We

conclude that we can check exactness in terms of $\hat{M}$ only if the relaxation is verifiable exact in the sense of our definition. $\square$

Let us first relate verifiable exactness to the following perturbation of (REL) with some fixed $t \in \mathbb{R}$, denoted as $(\text{REL})_t$:

$$\gamma_{\text{rel}}(t) = \inf \left\{ \langle c, x \rangle : G(y) < 0, \begin{pmatrix} I & 0 \\ A & B \end{pmatrix}' H(y) \begin{pmatrix} I & 0 \\ A & B \end{pmatrix} + W(x) + t\hat{W} < 0 \right\}.$$

For all t in a neighborhood of $t = 0$, $\gamma_{\text{rel}}(t)$ is finite (since finite at $t = 0$ by assumption), nondecreasing (since $\hat{W} \geq 0$), and convex. Therefore its right derivative $\gamma'_{\text{rel}}(0)$ at zero does exist and is nonnegative. By a rather standard result in convex analysis $\gamma'_{\text{rel}}(0)$ equals $\sup\{\langle \hat{W}, M \rangle : M \in \mathcal{M}\}$ [5]. This leads to the following alternative characterization of verifiable exactness.

LEMMA 6.2. $\gamma'_{\text{rel}}(0) = 0$

This reformulation opens the path for a relation to the perturbed version of (ROB), again obtained by just replacing $W(x)$ with $W(x) + t\hat{W}$ for $t \in \mathbb{R}$. Due to the particular structure of $\hat{W}$, $(\text{ROB})_t$ reads explicitly as

$$\gamma_{\text{opt}}(t) = \inf \{ \langle c, x \rangle : L(x, \Delta) + t[F(\Delta) - F(\Delta_0)]'[F(\Delta) - F(\Delta_0)] < 0 \text{ for all } \Delta \in \Delta \}.$$

Similarly as for $\gamma_{\text{rel}}(t)$, it is seen that $\gamma_{\text{opt}}(t)$ is finite, nondecreasing, and convex in a neighborhood of $t = 0$. Therefore its right derivative $\gamma'_{\text{opt}}(0)$ exists and is nonnegative.

As in Lemma 2.1 one shows that $\gamma_{\text{opt}}(t) \leq \gamma_{\text{rel}}(t)$. If the relaxation is verifiably exact, $\gamma_{\text{opt}}(0) = \gamma_{\text{rel}}(0)$ implies $0 \leq \gamma'_{\text{opt}}(0) \leq \gamma'_{\text{rel}}(0) = 0$, and thus $\gamma'_{\text{opt}}(0) = 0$. Hence we have identified $\gamma'_{\text{opt}}(0) = 0$ as a necessary condition for (ROB) to admit a verifiable exact relaxation.

THEOREM 6.3. $\gamma'_{\text{opt}}(0) > 0$

It is interesting to observe that $\gamma'_{\text{opt}}(0)$ is positive if there exists a worst-case uncertainty $\Delta_1 \in \Delta$ that is different from Δ_0 and for which $F(\Delta_1) - F(\Delta_0)$ has no kernel. This is a consequence of the following more refined condition.

THEOREM 6.4. $\gamma'_{\text{opt}}(0) > 0$ if and only if $\Delta_1 \in \Delta \setminus \{\Delta_0\}$ such that

$$(6.5) \quad \inf \{ \langle c, x \rangle : L(x, \Delta_1) < 0 \} = \gamma_{\text{opt}}$$

Since (ROB) is feasible, (6.5) is strictly feasible. Consider

$$\gamma(t) = \inf \{ \langle c, x \rangle : L(x, \Delta_1) + t[F(\Delta_1) - F(\Delta_0)]'[F(\Delta_1) - F(\Delta_0)] < 0 \}.$$

Clearly, $\gamma(t) \leq \gamma_{\text{opt}}(t)$. By hypothesis $\gamma(0) = \gamma_{\text{opt}}(0) = \gamma_{\text{opt}}$. Therefore $\gamma'(0) \leq \gamma'_{\text{opt}}(0)$. It hence suffices to show that $\gamma'(0) > 0$. This is true if there exists a dual optimal solution Z for (6.5) such that $\langle [F(\Delta_1) - F(\Delta_0)]'[F(\Delta_1) - F(\Delta_0)], Z \rangle > 0$ or, since $Z \geq 0$, equivalently $[F(\Delta_1) - F(\Delta_0)]Z \neq 0$. $\square$

It remains to clarify under which conditions $\gamma'_{\text{opt}}(0)$ vanishes such that we indeed have a chance to construct a verifiably exact relaxation. This is the contents of the following result whose nontrivial proof is found in Appendix B.

THEOREM 6.5. $\gamma'_{\text{opt}}(0) = 0$ if and only if $x_{\text{opt}} \in X_{\text{opt}}$ such that

$$\ker[L(x_{\text{opt}}, \Delta)] \subset \ker[F(\Delta) - F(\Delta_0)] \text{ for all } \Delta \in \Delta \setminus \{\Delta_0\}.$$

As an immediate consequence, $\gamma'_{\text{opt}}(0)$ vanishes if there exists some $x_{\text{opt}} \in X_{\text{opt}}$ such that $L(x_{\text{opt}}, \Delta) < 0$ for all $\Delta \in \Delta$ different from Δ_0 . The latter property requires Δ_0 to be unique as a worst-case uncertainty.

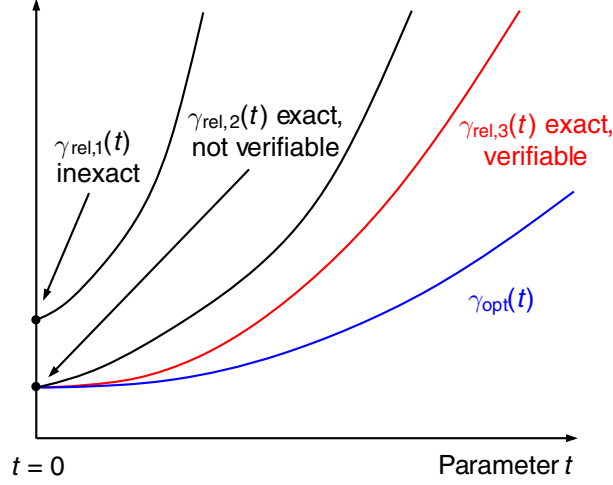


FIG. 1. Illustration of inexact, not verifiably exact, and verifiably exact relaxations.

Theorems 6.4 and 6.5 are not as unrelated as they look. Indeed, if the conditions in Theorem 6.4 hold for Δ_1 , one can prove that

$$(6.6) \quad \ker[L(x_{\text{opt}}, \Delta_1)] \not\subset \ker[F(\Delta_1) - F(\Delta_0)] \quad \text{for all } x_{\text{opt}} \in X_{\text{opt}},$$

which in turn implies that the sufficient condition in Theorem 6.5 is not satisfied.

Indeed, suppose Δ_1 satisfies (6.5) and $x_{\text{opt}} \in X_{\text{opt}}$ is arbitrary. Then x_{opt} is clearly optimal for (6.5). By complementary slackness any dual optimal solution Z of (6.5) satisfies $L(x_{\text{opt}}, \Delta_1)Z = 0$ or $\text{im}(Z) \subset \ker[L(x_{\text{opt}}, \Delta_1)]$. Since $[F(\Delta_1) - F(\Delta_0)]Z \neq 0$ just means $\text{im}(Z) \not\subset \ker[F(\Delta_1) - F(\Delta_0)]$, we infer (6.6). $\square$

To summarize, Theorem 6.4 formulates a sufficient condition for $\gamma'_{\text{opt}}(0)$ to be positive, which implies that there cannot exist a verifiably exact relaxation. Roughly speaking, this property holds if Δ_0 is not unique as a worst-case uncertainty. In contrast Theorem 6.5 contains a sufficient condition for $\gamma'_{\text{opt}}(0)$ to vanish, which makes it possible to construct verifiably exact relaxations. In rough terms this is true if the worst-case uncertainty Δ_0 is unique. Figure 1 and the subsequent example provide graphical and numerical illustrations.

Let us choose a one-parameter example in order to guarantee that the standard relaxations are exact (Theorem 5.4), although they might not be verifiably exact. Consider the problem of minimizing a nonconstant rational function $f(\delta)$ over $\delta \in [-1, 1]$ (without pole in this interval). Standard realization theory allows us to construct a representation $f(\delta) = C\Delta(\delta)(I - A\Delta(\delta))^{-1}B + D$ with $\Delta(\delta) = \delta I$ of minimal size and (A, B) in controllability canonical form. Our problem can be formulated as infimizing x such that $f(\delta) < x$ for all $\delta \in [-1, 1]$, which is subsumed to (ROB) as in section 2. Clearly, the unique minimizer of (ROB) is $x_{\text{opt}} = \max_{\delta \in [-1, 1]} f(\delta)$. We apply the (exact) relaxation as in Theorem 5.4 and conclude $\gamma'_{\text{opt}}(0) = \gamma'_{\text{rel}}(0)$.

If $f(\delta) = x_{\text{opt}}$ for exactly one $\delta_0 \in [-1, 1]$, then the relaxation is verifiably exact. Indeed, if $\delta \in [-1, 1] \setminus \{\delta_0\}$, we infer $f(\delta) - x_{\text{opt}} < f(\delta_0) - x_{\text{opt}}$, and hence $\ker(f(\delta) - x_{\text{opt}}) = \{0\}$; by Theorem 6.5 we have $\gamma'_{\text{opt}}(0) = 0$; therefore $\gamma'_{\text{rel}}(0) = 0$, and we can apply Lemma 6.2. We stress that this fact does not depend on the nature (order) of the zero δ_0 of the function $f(\delta) - x_{\text{opt}}$ as one might expect.

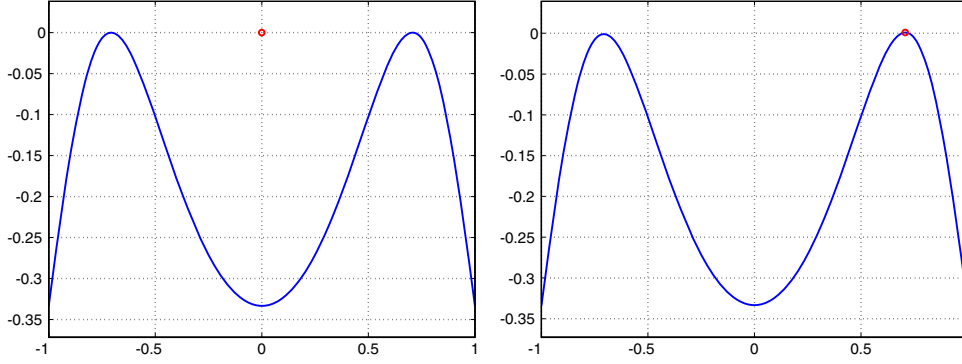


FIG. 2. Two maximizers (left) and one maximizer (right) with computed uncertainty (circle).

Now suppose that $f(\delta_0) = f(\delta_1) = x_{\text{opt}}$ for different $\delta_0, \delta_1 \in [-1, 1]$. Then neither the standard relaxation nor any relaxation can be verifiably exact. This follows from Theorem 6.3 since $\gamma'_{\text{opt}}(0) > 0$, which is a consequence of Theorem 6.4 and $\delta_0(I - \delta_0 A)^{-1}B \neq \delta_1(I - \delta_1 A)^{-1}B$. The latter is true since equality implies $\delta_0^j / \det(I - \delta_0 A) = \delta_1^j / \det(I - \delta_1 A)$ for $j = 1, 2$ because (A, B) is in controllability canonical form and A has at least dimension two. (Recall that $f(\delta) - x_{\text{opt}}$ has two zeros without being identically zero.) If $\delta_0 = 0$, we infer $\delta_1 = 0$ from $j = 1$, a contradiction; otherwise we have $\delta_0[\delta_0 / \det(I - \delta_0 A)] = \delta_1[\delta_1 / \det(I - \delta_1 A)] = \delta_1[\delta_0 / \det(I - \delta_0 A)]$, and hence $\delta_0 = \delta_1$, again a contradiction.

For a concrete numerical example consider $f(\delta) = (\delta^2 - \delta^4)/(1 - \delta^2 + \delta^4) - 1/3$ (Figure 2). The standard relaxation computes the correct maximal value, but it is not possible to confirm exactness since the function has two maximizers; the computed M has rank two and (6.1) admits the minimizer $\delta = 0$. If we slightly perturb the function such that the maximum is attained at one point only, we can indeed confirm exactness of the relaxation and compute the correct maximizer.

7. A new family of asymptotically exact relaxations. It can very well happen that the relaxations suggested in section 2 are not exact. Although there does exist a refinement of the convex hull block relaxation if Δ admits the standard real-repeated block structure with box constraints as in structured singular value theory [24, 45], no such schemes are known for general value sets that can be shown to approach exactness. In this last section we would like to construct a sequence of relaxations which can be proven to be asymptotically exact.

With $\Delta = \text{con}\{\Delta_1, \dots, \Delta_{n_{\text{ext}}}\}$ the actual goal is to approximate the class of all multipliers which satisfy (1.3). With the Hermitian-valued homogenous polynomial

$$\Lambda_d(P, \lambda) := (\lambda_1 + \dots + \lambda_{n_{\text{ext}}})^d \left[\sum_{j=1}^{n_{\text{ext}}} \lambda_j \begin{pmatrix} \Delta_j \\ I \end{pmatrix} \right]' P \left[\sum_{k=1}^{n_{\text{ext}}} \lambda_k \begin{pmatrix} \Delta_k \\ I \end{pmatrix} \right],$$

condition (1.3) is equivalent to

$$(7.1) \quad \Lambda_0(P, \lambda) > 0 \quad \text{for all } \lambda \in \Sigma,$$

where $\Sigma = \{(\lambda_1, \dots, \lambda_{n_{\text{ext}}}) \geq 0 : \lambda_1 + \dots + \lambda_{n_{\text{ext}}} = 1\}$ denotes the standard simplex in $\mathbb{R}^{n_{\text{ext}}}$. At this point we make contact with a classical question in real algebraic

geometry on the characterization of positive polynomials on simplices or polytopes [34] which has recently found renewed interest in control [8, 30]. As the only complication we have to deal with matrix-valued functions, and we need to guarantee positivity with respect to the semidefinite cone. We have introduced Λ_d for $d = 0, 1, 2, \dots$ in order to formulate the following generalization of Pólya's theorem [11, 34, 10] to Hermitian-valued functions. For this purpose let us denote by $C_{d,\nu}(P)$, $\nu = 1, \dots, n_d$, the nonzero coefficients of the monomials in a Taylor expansion of $\Lambda_d(P, \lambda)$ around $\lambda = 0$, and observe that they have the structure

$$C_{d,\nu}(P) = \sum_{j,k=1}^{n_{\text{ext}}} \alpha_{d,\nu}(j, k) \left(\begin{pmatrix} \Delta_j \\ I \end{pmatrix}' P \begin{pmatrix} \Delta_k \\ I \end{pmatrix} + \begin{pmatrix} \Delta_k \\ I \end{pmatrix}' P \begin{pmatrix} \Delta_j \\ I \end{pmatrix} \right)$$

with coefficients $\alpha_{d,\nu}(j, k)$ that can be easily constructed recursively.

THEOREM 7.1. *Let $P \in \mathbb{S}^n$ and $\Lambda_0(P, \lambda) \succ 0$ for all $\lambda \in \Sigma$. Then, for all $d \geq 0$, $\nu = 1, \dots, n_d$, $C_{d,\nu}(P) \succ 0$. Define*

$$u(P) := \max_{\nu=1, \dots, n_0} \lambda_{\max}(C_{0,\nu}(P)), \quad l(P) = \min_{\lambda \in \Sigma} \lambda_{\min}(\Lambda_0(P, \lambda))$$

and choose any integer d with $d \geq u(P)/l(P) - 2$. For all vectors v of norm one we infer $u(P) \geq v' C_{0,\nu}(P) v$ and $\min_{\lambda \in \Sigma} v' \Lambda_0(P, \lambda) v \geq l(P)$. Due to [11] we can hence conclude that all coefficients $v' C_{d,\nu}(P) v$ of the real-valued polynomial $v' \Lambda_d(P, \lambda) v$ are positive. Since v was arbitrary, the result follows. $\square$

This motivates us to introduce the sequence of relaxations

$$H(P) = P, \quad G_d(P) = -\text{diag}_{\nu=1, \dots, n_d}(C_{d,\nu}(P)) \quad \text{for } d = 0, 1, 2, \dots$$

with corresponding optimal values $\gamma_{\text{rel},d}$. For growing d the number of variables remains fixed, while the number of constraints grows exponentially. This conceptual advantage might lay ground for structure exploitation. Even for small d one should certainly try to numerically verify exactness as in section 6, which is also possible on the basis of the rank-one principle as shown next. Moreover, the sequence of relaxation can be proven to be asymptotically exact.

THEOREM 7.2. *Let $\gamma_{\text{opt}} = \inf_{\lambda \in \Sigma} \lambda_{\min}(\Lambda_0(P, \lambda))$. Then, $\lim_{d \rightarrow \infty} \gamma_{\text{rel},d} = \gamma_{\text{opt}}$.*

PROOF. Choose $\epsilon > 0$. Since Δ is compact, we can apply the full block S-procedure to infer that there exist (x, P) with (1.3), (1.4), and $\langle c, x \rangle < \gamma_{\text{opt}} + \epsilon$. By Theorem 7.1, there exists some d such that, for this particular P , we have $G_d(P) < 0$. This implies $\gamma_{\text{rel},d} \leq \langle c, x \rangle < \gamma_{\text{opt}} + \epsilon$, and hence convergence, since $\gamma_{\text{opt}} \leq \gamma_{\text{rel},d}$.

Let us now verify the hypothesis in Corollary 3.3. Suppose $\text{col}(w, z)' P \text{col}(w, z) \geq 0$ for all P with $C_{d,\nu}(P) \geq 0$, $\nu = 1, \dots, n_d$. Since the latter system of inequalities is strictly feasible we infer (Farkas) that there exists $Z_\nu \geq 0$ with

$$\begin{pmatrix} zz' & | & wz' & | & ww' \end{pmatrix} = \sum_{\nu=1}^{n_d} \sum_{j,k=1}^{n_{\text{ext}}} \alpha_{d,\nu}(j, k) \begin{pmatrix} 2Z_\nu & | & \Delta_k Z_\nu + \Delta_j Z_\nu & | & \Delta_k Z_\nu \Delta_j' + \Delta_j Z_\nu \Delta_k' \end{pmatrix}.$$

Since $\alpha_{d,\nu}(j, k) \geq 0$ and $Z_\nu \geq 0$ there exist $\beta_\nu \geq 0$ with $Z_\nu = z\beta_\nu z'$ (first relation). If $z = 0$, we infer $Z_\nu = 0$, and hence $w = 0$ (third relation), and hence $w = \Delta_0 z$ for any $\Delta_0 \in \Delta$. If $z \neq 0$, we conclude $\sum_{\nu=1}^{n_d} \sum_{j,k=1}^{n_{\text{ext}}} 2\alpha_{d,\nu}(j, k)\beta_\nu = 1$; by right-multiplying the second relation with z and dividing by $z'z \neq 0$ we arrive at $w = \Delta_0 z$ for the

matrix $\Delta_0 = \sum_{\nu=1}^{n_d} \sum_{j,k=1}^{n_{\text{ext}}} \alpha_{d,\nu}(j,k) \beta_{\nu}(\Delta_k + \Delta_j)$; it remains to observe $\Delta_0 \in \mathbf{\Delta}$ since it is a convex combination of elements in $\mathbf{\Delta}$. $\square$

Under the hypothesis in section 6, exactness is verifiable only if $\gamma'_{\text{opt}}(0)$ vanishes for some worst-case uncertainty. Let us assume this to be true. Due to Theorem 7.2 (applied to the perturbed problems) we observe that $\lim_{d \rightarrow \infty} \gamma'_{\text{rel},d}(0) = \gamma'_{\text{opt}}(0) = 0$. One can then prove that $\lim_{d \rightarrow \infty} \nu(M_d) = 0$, with $\nu(\cdot)$ as defined in (6.1) and (M_d, N_d) being an arbitrary dual optimal solution of the Pólya relaxation of degree d . Therefore $\nu(M_d)$ is small for large d , which implies that we can indeed computationally confirm approximate exactness by Theorem 6.1. $\square$

The suggested relaxation scheme is applicable for small numbers of extreme points n_{ext} in the description of $\mathbf{\Delta}$, and for small degrees d , which makes them particularly applicable for the analysis and synthesis of linear-parameter-varying systems with few parameters, and especially for the techniques in [38]. At this point we emphasize the need for constructing relaxations based on implicit descriptions of the uncertainty sets $\mathbf{\Delta}$ in order to keep the computational complexity moderate, with initial results to be found in [40]. Moreover, it is an interesting topic (left for future research) to investigate how the Pólya relaxations and the corresponding exactness characterization of section 6 are related to the moment approach for solving polynomial programs and the construction of optimal solutions as presented in [19]. Let us finally illustrate all our results for a variant of an example in [29].

Determine the maximum of

$$f(x, y) = 1 - 2 \frac{ax^2y}{2 - 2ay + ax^2y - x^2} - \frac{y(2ay + x^2 + ax^2y - 2)}{2 - 2a^2y^2 - x^2 + x^2a^2y^2}$$

on $\{(x, y) \in \mathbb{R}^2 : -0.8 \leq x \leq 0.7, -0.65 \leq y \leq 0.7\}$ for $a \in [0, 1]$. The function admits a linear fractional representation with $\Delta(x, y) = \text{diag}(xI_2, yI_2)$, which allows us to compute upper bounds on its maximum by applying the repeated block and the convex hull block relaxations of section 2.1 and the Pólya relaxation as described above. Even if (3.1) is not solvable, we can compute an optimizer $\Delta(x_0, y_0)$ of (6.1) and determine the lower bound $f(x_0, y_0)$. The results for repeated or convex hull block relaxations are displayed in Figure 3(left), where we observe that exactness is guaranteed for the former if $a \in [0, 0.37]$, while the latter leads to exactness if $a \in [0.37, 0.77]$. For $a \in [0.77, 0.88]$ or $a \in [0.99, 1]$, the Pólya relaxation with $d = 0$ is exact, while degree $d = 2$ is required for exactness if $a \in [0.88, 0.955] \cup [0.975, 0.99]$. The computed upper bounds for $a \in [0.955, 0.975]$ can be verified to be approximately exact with a relaxation of degree $d = 3$ (Figure 3(right)).

8. Conclusions. For general LMI relaxations of robust semidefinite programming problems with rational uncertainty dependence we have provided in this paper novel results on the verification of exactness, and we have investigated how far these tests are applicable in practice. This has been complemented with a sequence of asymptotically exact relaxations if the uncertainty set is finitely generated. Particular attention has been devoted to relaxations as they appear in robust control.

Appendix A. Auxiliary results.

The following slight generalization of standard Lyapunov or Stein equations is elementary to prove by suitable congruence transformations; see also [20].

LEMMA A.1. $q, r \in \mathbb{R}$, $s \in \mathbb{C}$

$$(A.1) \quad \delta = \left\{ \delta \in \mathbb{C} : \begin{pmatrix} \delta \\ 1 \end{pmatrix}' \begin{pmatrix} q & s \\ s' & r \end{pmatrix} \begin{pmatrix} \delta \\ 1 \end{pmatrix} \geq 0 \right\}$$

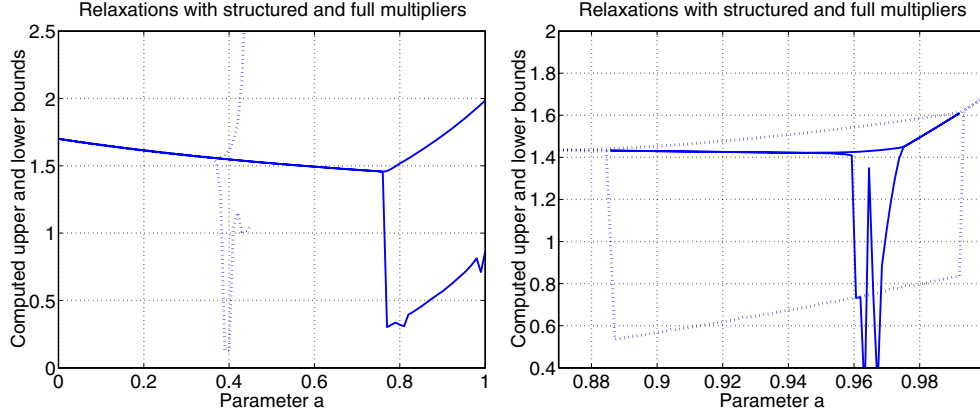


FIG. 3. Left: bounds with repeated (dotted) and convex hull block multipliers (full). Right: bounds with Pólya relaxations of degrees $d = 0$ (dotted) and $d = 2$ (full).

$$\delta = \begin{cases} \{ \delta \in \mathbb{C} : qr - |s|^2 < 0 \} & q > 0 \\ \{ \delta \in \mathbb{C} : q = 0 \} & q = 0 \\ \{ \delta \in \mathbb{C} : q < 0 \} & q < 0 \end{cases}$$

$$(A.2) \quad \begin{pmatrix} I \\ A \end{pmatrix}' \begin{pmatrix} qX & sX \\ s'X & rX \end{pmatrix} \begin{pmatrix} I \\ A \end{pmatrix} = P.$$

$$(A.2) \quad I - \delta A, \quad \delta \in \delta, \quad A, \quad X \geq 0, \quad P \leq 0$$

Consider first $q = 0$. Then $\delta = \{\delta \in \mathbb{C} : \delta's + s'\delta + r \geq 0\}$. If $s = 0$, then $r \geq 0$ (since δ is nonempty), and hence $\delta = \mathbb{C}$, a contradiction. Therefore $s \neq 0$, δ equals the genuine unbounded half-plane $\delta = \{\delta \in \mathbb{C} : (\delta + r/(2s'))'s + s'(\delta + r/(2s')) \geq 0\}$, and (A.2) is equivalent to

$$(A.3) \quad \begin{pmatrix} I + r/(2s')A \\ A \end{pmatrix}' \begin{pmatrix} 0 & sX \\ s'X & 0 \end{pmatrix} \begin{pmatrix} I + r/(2s')A \\ A \end{pmatrix} = P.$$

Since $-r/(2s') \in \delta$ we conclude that $I + r/(2s')A$ is nonsingular. Let us define

$$M = sA(I + r/(2s')A)^{-1} \quad \text{with} \quad \sigma(M) = \left\{ s\lambda(1 + r\lambda/(2s'))^{-1} : \lambda \in \sigma(A) \right\}.$$

Then (A.3) transforms to

$$(A.4) \quad M'X + XM = (I + r/(2s')A)^{-*}P(I + r/(2s')A)^{-1}.$$

If $z \in \sigma(M)$, then $z = s\lambda(1 + r\lambda/(2s'))^{-1}$ for some $\lambda \in \sigma(A)$. Since δ is unbounded, A is nonsingular, and hence $\lambda \neq 0$. Moreover, since $I - A/\lambda$ is singular, $1/\lambda$ is not contained in δ , and thus $(1/\lambda + r/(2s'))'s + s'(1/\lambda + r/(2s')) < 0$ or $[s\lambda(1 + r\lambda/(2s'))^{-1}] + [s\lambda(1 + r\lambda/(2s'))^{-1}]' < 0$, which just means that the real part of z is negative. Therefore M is Hurwitz, which allows us to conclude that (A.4) is uniquely solvable for X , where $X \geq 0$ if $P \leq 0$.

Now suppose $q \neq 0$. Then $\delta \in \delta$ iff $q(|\delta + s/q|^2 - t) \geq 0$ with $t := (|s|^2 - qr)/q^2$. If $t \leq 0$, then δ either consists of at most one point (for $q < 0$) or is the whole complex

plane (for $q > 0$), which are contradictions. Therefore t must be positive. Moreover, the matrix equation (A.2) now transforms into

$$(A.5) \quad \begin{pmatrix} I + As/q \\ A \end{pmatrix}' \begin{pmatrix} qX & 0 \\ 0 & -qtX \end{pmatrix} \begin{pmatrix} I + As/q \\ A \end{pmatrix} = P.$$

Suppose $q < 0$. Then δ is the (bounded) closed disk with center $-s/q$ and radius t . Hence $I + As/q$ is nonsingular, and (A.5) transforms into

$$X - M'XM = \frac{1}{q}((I + As/q)^{-1})'P(I + As/q)^{-1} \quad \text{with} \quad M = \sqrt{t}A(I + As/q)^{-1}.$$

If $z \in \sigma(M)$, there exists $\lambda \in \sigma(A)$ with $z = \sqrt{t}\lambda(1 + \lambda s/q)^{-1}$. If $\lambda \neq 0$, then $1/\lambda$ is not in δ , and hence $|1/\lambda + s/q|^2 > t$ or $|z|^2 = t|1/\lambda + s/q|^{-2} < 1$. If $\lambda = 0$, we have $z = 0$. Therefore M has all its eigenvalues in the open unit disk, and the desired result follows from those for the standard Stein equation.

Finally, let $q > 0$. Then δ is the (unbounded) exterior of the open disk with center $-s/q$ and radius t . Therefore A is nonsingular, and (A.5) transforms into

$$M'XM - X = \frac{1}{qt}(A^{-1})'PA^{-1} \quad \text{with} \quad M = \sqrt{1/t}(I + As/q)A^{-1}.$$

If $z \in \sigma(M)$, there exists a nonzero $\lambda \in \sigma(A)$ with $z = \sqrt{1/t}(1 + \lambda s/q)/\lambda = \sqrt{1/t}(1/\lambda + s/q)$. Since $1/\lambda$ is not in δ we infer $|1/\lambda + s/q|^2 < t$, and hence $|z|^2 < 1$. Therefore, $|\sigma(M)| < 1$, and the proof is completed as in the previous case. $\square$

The following simple but instrumental interpolation results are rather immediate consequences of a nonstrict version of the projection or elimination lemma [17, 38] and generalize various well-known concrete versions [35, 28, 32, 20].

LEMMA A.2. . . . $Q < 0$. . . $R \geq 0$. . . $R - SQ^{-1}S' > 0$. . .

$$\begin{pmatrix} w \\ z \end{pmatrix}' \begin{pmatrix} Q & S \\ S' & R \end{pmatrix} \begin{pmatrix} w \\ z \end{pmatrix} \geq 0 \Rightarrow w = \Delta z, \quad \Delta \bullet \bullet \bullet \begin{pmatrix} \Delta \\ I \end{pmatrix}' P \begin{pmatrix} \Delta \\ I \end{pmatrix} \geq 0,$$

. . . $w = \Delta z$. . . If $z = 0$, then $w'Qw \geq 0$, and hence $w = 0$; we can take $\Delta = 0$. Otherwise let us assume without loss of generality (w.l.o.g.) that z equals the first standard unit vector e_1 . Then we infer $w = (wX)z$ for an arbitrary X . With compatible partitions of S and R , the hypothesis and the inequality to solve read, respectively, as

$$\begin{pmatrix} w \\ 1 \end{pmatrix}' \begin{pmatrix} Q & S_1 \\ S'_1 & R_1 \end{pmatrix} \begin{pmatrix} w \\ 1 \end{pmatrix} \geq 0, \quad \begin{pmatrix} w & X \\ 1 & 0 \\ 0 & I \end{pmatrix}' \begin{pmatrix} Q & S_1 & S_2 \\ S'_1 & R_1 & R_{12} \\ S'_2 & R'_{12} & R_2 \end{pmatrix} \begin{pmatrix} w & X \\ 1 & 0 \\ 0 & I \end{pmatrix} \geq 0.$$

Both are equivalently translated into

$$\begin{pmatrix} -Q^{-1} & w + Q^{-1}S_1 \\ * & R_1 - S'_1Q^{-1}S_1 \end{pmatrix} \geq 0, \quad \begin{pmatrix} -Q^{-1} & w + Q^{-1}S_1 & X + Q^{-1}S_2 \\ * & R_1 - S'_1Q^{-1}S_1 & R_{12} - S'_1Q^{-1}S_2 \\ * & * & R_2 - S'_2Q^{-1}S_2 \end{pmatrix} \geq 0.$$

With $T = R - S'Q^{-1}S > 0$, partitioned suitably, the second one is equivalent to

$$\begin{pmatrix} -Q^{-1} & w + Q^{-1}S_1 & X + Q^{-1}S_2 - (w + Q^{-1}S)T_1^{-1}T_{12} \\ * & R_1 - S'_1Q^{-1}S_1 & 0 \\ * & 0 & T_2 - T'_{12}T_1^{-1}T_{12} \end{pmatrix} \geq 0.$$

It is now obvious that $X = -Q^{-1}S_2 + (w + Q^{-1}S)T_1^{-1}T_{12}$ is a solution. $\square$

LEMMA A.3. $q, r \in \mathbb{R}$, $s \in \mathbb{C}$, $w, z \in \mathbb{C}^n$, $w \neq z$, $w \neq 0$, $z \neq 0$.

$$(A.6) \quad qr - |s|^2 < 0 \quad \left(\begin{array}{c} w' \\ z' \end{array} \right)' \left(\begin{array}{cc} q & s' \\ s & r \end{array} \right) \left(\begin{array}{c} w' \\ z' \end{array} \right) \geq 0.$$

$z \neq 0$. If $w = z\delta$ for some $\delta \in \mathbb{C}$, then δ is contained in (A.1) since (A.6) implies

$$z' \left(\begin{array}{c} \delta' z' \\ z' \end{array} \right)' \left(\begin{array}{cc} q & s' \\ s & r \end{array} \right) \left(\begin{array}{c} \delta' z' \\ z' \end{array} \right) z = (z'z)^2 \left(\begin{array}{c} \delta' \\ 1 \end{array} \right)' \left(\begin{array}{cc} q & s' \\ s & r \end{array} \right) \left(\begin{array}{c} \delta' \\ 1 \end{array} \right) \geq 0,$$

and hence $q\delta\delta' + s'\delta + s\delta' + r \geq 0$.

Let us now prove the existence of δ . If $q \leq 0$, then $qw' + s'wz' + szw' + rzz' \geq 0$ implies $\ker(z') \subset \ker(w')$ (by making use of $z, s \neq 0$ if $q = 0$), and therefore δ exists. If $q > 0$ we infer from (A.6) that

$$\left(\begin{array}{c} w' + (s/q)z' \\ z' \end{array} \right)' \left(\begin{array}{cc} q & 0 \\ 0 & r - |s|^2/q \end{array} \right) \left(\begin{array}{c} w' + (s/q)z' \\ z' \end{array} \right) \geq 0.$$

Since $r - |s|^2/q < 0$ we conclude $\ker(w' + (s/q)z') \subset \ker(z')$, and there exists a nonzero $\delta \in \mathbb{C}$ with $z = (w + z(s/q)')\delta$. Hence $w = z(1/\delta - (s/q)')$. $\square$

If the trace of the complex matrix A is zero, it is a simple consequence of the convexity of the numerical range of A that there exists a nonzero vector x with $x'Ax = 0$ [21]. The following self-contained proof is short and fully elementary.

LEMMA A.4. $A \in \mathbb{C}^{n \times n}$, $\text{Trace}(A) = 0$, $x \in \mathbb{C}^n$, $x'Ax = 0$.

The proof is by induction on the dimension of A , where $n = 1$ is trivial. Suppose that x_0 minimizes $|x'Ax|$ on the compact unit ball $\{x \in \mathbb{C}^n : x'x = 1\}$. We need to prove that the minimum is zero and assume that it is positive. W.l.o.g. we can assume that x_0 is the first standard unit vector, and we partition A conformably as

$$A = \left(\begin{array}{cc} a & b' \\ c & D \end{array} \right) \quad \text{with} \quad |x_0'Ax_0| = |a| > 0.$$

Since $0 = \text{Trace}(A) = a + \text{Trace}(D) = \text{Trace}(a/(n-1)I + D)$ we infer by induction hypothesis that there exists a vector y with $y'y = 1$ and $y'(a/(n-1)I + D)y = 0$ or $y'Dy = -a/(n-1)$. Define $u = -b'y/a$ and $v = -y'c/a$. It is not difficult to see that there always exists some ϕ such that $\gamma = e^{i\phi}u + e^{-i\phi}v$ is real and nonnegative. For some $r < 1$ close to one we infer

$$\begin{aligned} & \left| \left(\begin{array}{c} r \\ ye^{i\phi}\sqrt{1-r^2} \end{array} \right)' A \left(\begin{array}{c} r \\ ye^{i\phi}\sqrt{1-r^2} \end{array} \right) \right| \\ &= \left| r^2a + r\sqrt{1-r^2}e^{i\phi}b'y + r\sqrt{1-r^2}e^{-i\phi}y'c - \frac{1-r^2}{n-1}a \right| \\ &= |a| \left| \frac{nr^2-1}{n-1} - r\sqrt{1-r^2}(e^{i\phi}u + e^{-i\phi}v) \right| = |a| \left| \frac{nr^2-1}{n-1} - \gamma r\sqrt{1-r^2} \right| < |a|. \end{aligned}$$

This is a contradiction since $\text{col}(r, ye^{i\phi}\sqrt{1-r^2})$ has norm one. $\square$

The following consequence [21] is recapitulated for the reader's convenience.

LEMMA A.5. *If $A \in \mathbb{C}^{n \times n}$ and $\text{Tr}(A) = 0$, then $\text{diag}(U'AU) = 0$ for some unitary U .*

Lemma A.4 is the first step in the following induction argument. Suppose after step j we have constructed a matrix U_j with j columns that satisfies $\text{diag}(U_j'AU_j) = 0$ and $U_j'U_j = I$. There exists an extension $(U_j \ V_j)$ to a unitary matrix. We infer $0 = \text{Tr}(A) = \text{Tr}((U_j \ V_j)'A(U_j \ V_j)) = \text{Tr}(U_j'AU_j) + \text{Tr}(V_j'AV_j) = \text{Tr}(V_j'AV_j)$. By Lemma A.4 there exists a unit vector z with $z'V_j'AV_jz = 0$, and $U_{j+1} := (U_j \ V_jz)$ has $j+1$ columns and satisfies $\text{diag}(U_{j+1}'AU_{j+1}) = 0$, $U_{j+1}'U_{j+1} = I$. $\square$

The proof is constructive if x in Lemma A.4 can actually be computed. This is simple if A is Hermitian: Determine a unitary U which renders $D = U'AU$ diagonal; if e is the all-ones vector, then $e'De = \text{Tr}(D) = 0$; choose $x := Ue/\|e\|$. $\square$

The following rather immediate consequence is instrumental for the proof of Theorem 5.3. Note that its proof is constructive if $A_3 = 0$.

LEMMA A.6. *If $A_j \in \mathbb{C}^{n \times n}$, $X \geq 0$, $X \neq 0$, $\langle A_j, X \rangle \geq 0$, $j = 1, 2, 3$, and $x \neq 0$, then $x'A_jx \geq 0$, $j = 1, 2, 3$.*

Decompose $X = U'U$ with U of full row rank $r > 0$ and note that $0 \leq \text{Tr}(A_jX) = \text{Tr}(U'A_jU)$. Define

$$M := \left[U'A_1U - \frac{1}{r} \text{Tr}(U'A_1U)I \right] + i \left[U'A_2U - \frac{1}{r} \text{Tr}(U'A_2U)I \right] \quad \text{with} \quad \text{Trace}(M) = 0.$$

By Lemma A.5 there exists a unitary matrix V with $\text{diag}(V'MV) = 0$. Since $0 \leq \text{Tr}(U'A_3U) = \text{Tr}(V'[U'A_3U]V)$, at least one column v of V satisfies $v'[U'A_3U]v \geq 0$. With $x = Uv$ the proof is concluded by observing

$$0 = \begin{pmatrix} \text{Re}(v'Mv) & \text{Im}(v'Mv) \end{pmatrix} = \begin{pmatrix} x'A_1x - \text{Tr}(A_1X)/r & x'A_2x - \text{Tr}(A_2X)/r \end{pmatrix},$$

and hence $x'A_1x \geq 0$ as well as $x'A_2x \geq 0$. $\square$

The following generalization of crucial technical decomposition lemmas as appearing in [35, 28, 41] is relevant for proving Theorem 5.4.

LEMMA A.7. *If $A, B \in \mathbb{C}^{n \times n}$, $q, r, \hat{q}, \hat{r} \in \mathbb{R}$, $s, \hat{s} \in \mathbb{C}$, $qr - |s|^2 < 0$, $\hat{q}\hat{r} - |\hat{s}|^2 < 0$, and $X \geq 0$, then*

$$(A.7) \quad \begin{pmatrix} A \\ B \end{pmatrix}' \begin{pmatrix} qX & sX \\ s'X & rX \end{pmatrix} \begin{pmatrix} A \\ B \end{pmatrix} = 0, \quad \begin{pmatrix} A \\ B \end{pmatrix}' \begin{pmatrix} \hat{q}X & \hat{s}X \\ \hat{s}'X & \hat{r}X \end{pmatrix} \begin{pmatrix} A \\ B \end{pmatrix} \geq 0.$$

Let $C = \begin{pmatrix} A \\ B \end{pmatrix}$. Then $x'Cx \geq \langle C, X \rangle$.

$$(A.8) \quad \begin{pmatrix} x'A \\ x'B \end{pmatrix}' \begin{pmatrix} q & s \\ s' & r \end{pmatrix} \begin{pmatrix} x'A \\ x'B \end{pmatrix} = 0, \quad \begin{pmatrix} x'A \\ x'B \end{pmatrix}' \begin{pmatrix} \hat{q} & \hat{s} \\ \hat{s}' & \hat{r} \end{pmatrix} \begin{pmatrix} x'A \\ x'B \end{pmatrix} \geq 0.$$

If we decompose $X = U'U$ with U of full row rank $n > 0$ we infer

$$(A.9) \quad \begin{pmatrix} UA \\ UB \end{pmatrix}' \begin{pmatrix} qI & sI \\ s'I & rI \end{pmatrix} \begin{pmatrix} UA \\ UB \end{pmatrix} = 0, \quad \begin{pmatrix} UA \\ UB \end{pmatrix}' \begin{pmatrix} \hat{q}I & \hat{s}I \\ \hat{s}'I & \hat{r}I \end{pmatrix} \begin{pmatrix} UA \\ UB \end{pmatrix} \geq 0.$$

Let us apply the result on the common transformation of nonsingular Hermitian matrices in [21, Theorem 4.5.19]: there exists a nonsingular $S \in \mathbb{C}^{2 \times 2}$ such that

either (Case 1)

$$S' \begin{pmatrix} q & s \\ s' & r \end{pmatrix} S = \pm \begin{pmatrix} 0 & 1 \\ 1 & 0 \end{pmatrix}, \quad S' \begin{pmatrix} \hat{q} & \hat{s} \\ \hat{s}' & \hat{r} \end{pmatrix} S = \pm \begin{pmatrix} 0 & \alpha \\ \alpha' & \beta \end{pmatrix} \quad \text{with } \alpha \neq 0$$

or (Case 2)

$$S' \begin{pmatrix} q & s \\ s' & r \end{pmatrix} S = \begin{pmatrix} -1 & 0 \\ 0 & 1 \end{pmatrix}, \quad S' \begin{pmatrix} \hat{q} & \hat{s} \\ \hat{s}' & \hat{r} \end{pmatrix} S = \begin{pmatrix} \alpha & 0 \\ 0 & \beta \end{pmatrix} \quad \text{with } \alpha, \beta \neq 0.$$

1. If the elements of S^{-1} are denoted by $t_{11}, t_{12}, t_{21}, t_{22}$ and if we define $F := t_{11}UA + t_{12}UB$, $G = t_{21}UA + t_{22}UB$, we infer for all $M = M'$ that

$$(A.10) \quad \begin{pmatrix} UA \\ UB \end{pmatrix}' \begin{pmatrix} qM & sM \\ s'M & rM \end{pmatrix} \begin{pmatrix} UA \\ UB \end{pmatrix} = \pm \begin{pmatrix} F \\ G \end{pmatrix}' \begin{pmatrix} 0 & M \\ M & 0 \end{pmatrix} \begin{pmatrix} F \\ G \end{pmatrix},$$

$$(A.11) \quad \begin{pmatrix} UA \\ UB \end{pmatrix}' \begin{pmatrix} \hat{q}M & \hat{s}M \\ \hat{s}'M & \hat{r}M \end{pmatrix} \begin{pmatrix} UA \\ UB \end{pmatrix} = \pm \begin{pmatrix} F \\ G \end{pmatrix}' \begin{pmatrix} 0 & \alpha M \\ \alpha' M & \beta M \end{pmatrix} \begin{pmatrix} F \\ G \end{pmatrix}.$$

Let us first concentrate on (A.10). For $M = I$ we infer $F'G + G'F = 0$. Choose a unitary V of dimension n such that

$$V'G = \begin{pmatrix} G_1 \\ 0 \end{pmatrix} \quad \text{with } G_1 \text{ of full row rank } m$$

and partition the rows of $V'F$ accordingly into the blocks F_1, F_2 . We observe $0 = F'G + G'F = F_1'G_1 + G_1'F_1$. Since $G_1x = 0$ implies $G_1'F_1x = 0$, and hence $F_1x = 0$, there exists a square matrix T of size m with $F_1 = TG_1$. From $0 = F_1'G_1 + G_1'F_1 = G_1'T'G_1 + G_1'TG_1$ we can conclude $T + T' = 0$ since G_1 has full row rank. Choose a set of orthonormal eigenvectors w_j with eigenvalue λ_j of T , where $\lambda_j' + \lambda_j = 0$. With the standard unit vectors e_j (of suitable length) define the orthonormal set of vectors

$$(A.12) \quad x_j = V \begin{pmatrix} w_j \\ 0 \end{pmatrix}, \quad j = 1, \dots, m, \quad x_j = V \begin{pmatrix} 0 \\ e_{j-m+1} \end{pmatrix}, \quad j = m+1, \dots, n.$$

For $j = 1, \dots, m$ we infer $G'x_j = G_1'w_j$ and $F'x_j = F_1'w_j = G_1'T'w_j = \lambda_j G_1'w_j$ and hence $F'x_jx_j'G + G'x_jx_j'F = (\lambda_j + \lambda_j')(G_1'w_jw_j'G_1) = 0$. For $j = m+1, \dots, n$ we have $x_j'G = 0$ by construction, and thus $F'x_jx_j'G + G'x_jx_j'F = 0$ as well. With (A.10) for $M = x_jx_j'$ we have

$$(A.13) \quad \begin{pmatrix} x_j'UA \\ x_j'UB \end{pmatrix}' \begin{pmatrix} q & s \\ s' & r \end{pmatrix} \begin{pmatrix} x_j'UA \\ x_j'UB \end{pmatrix} = 0, \quad j = 1, \dots, n.$$

Let us now turn to (A.11). For $M = I$ we infer $\alpha F'G + \alpha'G'F + \beta G'G \geq 0$, hence $\alpha F_1'G_1 + \alpha'G_1'F_1 + \beta G_1'G_1 \geq 0$, hence $G_1'(\alpha T' + \alpha'T + \beta I)G_1 \geq 0$, hence $\alpha T' + \alpha'T + \beta I \geq 0$, hence $\alpha\lambda_j' + \alpha'\lambda_j + \beta \geq 0$, and hence $\alpha F'x_jx_j'G + \alpha'G'x_jx_j'F + \beta G'x_jx_j'G = (\alpha\lambda_j + \alpha'\lambda_j' + \beta)(G_1'w_jw_j'G_1) \geq 0$ for all $j = 1, \dots, n$. We can thus conclude from (A.11) for $M = x_jx_j'$ that

$$(A.14) \quad \begin{pmatrix} x_j'UA \\ x_j'UB \end{pmatrix}' \begin{pmatrix} \hat{q} & \hat{s} \\ \hat{s}' & \hat{r} \end{pmatrix} \begin{pmatrix} x_j'UA \\ x_j'UB \end{pmatrix} \geq 0, \quad j = 1, \dots, n.$$

Finally, if $\max_{j=1,\dots,n} x_j' U C U' x_j = x_{j_0}' U C U' x_{j_0}$ we infer $\langle C, X \rangle = \text{Tr}(U C U') = \sum_{j=1}^n x_j' U C U' x_j \leq (\sqrt{n} U' x_{j_0})' C (\sqrt{n} U' x_{j_0})$, and hence $x = \sqrt{n} U' x_{j_0}$ does the job.

2. As in Case 1 we infer for all $M = M'$ that

$$(A.15) \quad \begin{pmatrix} U A \\ U B \end{pmatrix}' \begin{pmatrix} q M & s M \\ s' M & r M \end{pmatrix} \begin{pmatrix} U A \\ U B \end{pmatrix} = \begin{pmatrix} F \\ G \end{pmatrix}' \begin{pmatrix} -M & 0 \\ 0 & M \end{pmatrix} \begin{pmatrix} F \\ G \end{pmatrix},$$

$$(A.16) \quad \begin{pmatrix} U A \\ U B \end{pmatrix}' \begin{pmatrix} \hat{q} M & \hat{s} M \\ \hat{s}' M & \hat{r} M \end{pmatrix} \begin{pmatrix} U A \\ U B \end{pmatrix} = \begin{pmatrix} F \\ G \end{pmatrix}' \begin{pmatrix} \alpha M & 0 \\ 0 & \beta M \end{pmatrix} \begin{pmatrix} F \\ G \end{pmatrix}.$$

With (A.15) for $M = I$ we infer $G'G - F'F = 0$. Choose a unitary V as in Case 1 and observe $0 = G'G - F'F = G_1'G_1 - F_1'F_1$. Since $\ker(G_1) \subset \ker(F_1)$ there exists some T_1 with $F = T_1 G_1$, and $0 = G_1'(I - T_1' T_1)G_1$ implies $T_1' T_1 = I$. Extend to a unitary $(T_1 \ T_2)$ and define $T = (T_1 \ T_2)V'$ to conclude $F = TG$. With a set of orthonormal eigenvectors x_j of T with eigenvalues λ_j we infer $F'x_j x_j' F' = G' T x_j x_j' T' G = G' x_j x_j' G$ (since $|\lambda_j| = 1$), and thus, with (A.15) for $M = x_j x_j'$, we arrive at (A.13).

Let us finally show (A.14). For $M = I$ in (A.16) we infer $\alpha F'F + \beta G'G \geq 0$ or $(\beta - \alpha)G'G \geq 0$. If $G'G = 0$, then $F'F = 0$, and hence $G = 0$, $F = 0$. If $G'G \neq 0$, then $\beta \geq \alpha$, and thus $\alpha F'x_j x_j' F + \beta G'x_j x_j' G = (\beta - \alpha)G'x_j x_j' G \geq 0$. In both cases (A.14) follows from (A.16) for $M = x_j x_j'$, and the proof is completed as in Case 1. $\square$

Appendix B. Proof of Theorem 6.1. Let us introduce the abbreviations $F_1 = (I \ -F(\Delta_0))$, $F_2 = (F(\Delta_0)' \ I)$, $F = \text{col}(F_1, F_2)$. Note that $F_1 F_2' = 0$ and $F'F = \text{diag}(I + F(\Delta_0)F(\Delta_0)', I + F(\Delta_0)'F(\Delta_0))$, which implies $I \leq F'F \leq (1 + b^2)I$, and hence $\|F\| \leq 1 + b$ as well as $\|F^{-1}\| \leq 1$. Observe that the hypothesis on (Δ_0, ν_0) implies $\|(I - \Delta_0 A)F_1 M\| \leq \nu_0$, and hence $\|F_1 M\| \leq a\nu_0$, which guarantees $\|F_1 M F'\| \leq a(1 + b)\nu_0$. Define $\hat{M} = [F_2'(F_2 F_2')^{-1} F_2] M [F_2'(F_2 F_2')^{-1} F_2]$. We conclude

$$(B.1) \quad \begin{aligned} \|M - \hat{M}\| &= \|F^{-1} [F M F' - F F_2' (F_2 F_2')^{-1} F_2 M F_2' (F_2 F_2')^{-1} F_2 F'] (F')^{-1}\| \\ &\leq \left\| \begin{pmatrix} F_1 M F_1' & F_1 M F_2' \\ F_2 M F_1' & F_2 M F_2' \end{pmatrix} - \begin{pmatrix} 0 \\ F_2 \end{pmatrix} M \begin{pmatrix} 0 \\ F_2 \end{pmatrix}' \right\| \leq 2a(1 + b)\nu_0. \end{aligned}$$

Let us finally introduce $\hat{c}_j = -\langle W_j, \hat{M} \rangle$, $j = 1, \dots, n$, to define the vector $\hat{c}$. Now suppose that x is (ROB)-feasible. Since $F_2 W(x) F_2' \leq 0$ and $\hat{M} \geq 0$, $F_1 \hat{M} = 0$, $\langle W_j, \hat{M} \rangle + \hat{c}_j = 0$, $j = 1, \dots, n$, one shows as in the proof of Theorem 3.2 that $\langle \hat{c}, x \rangle \geq \langle W_0, \hat{M} \rangle$. If we recall $\langle W_0, M \rangle = \gamma_{\text{rel}}$, $c_j = -\langle W_j, M \rangle$, $j = 1, \dots, n$, and $\|x\|_1 \leq \xi$, we conclude

$$\begin{aligned} \langle c, x \rangle &= \langle \hat{c}, x \rangle + \sum_{j=1}^n \langle W_j, \hat{M} - M \rangle x_j \geq \gamma_{\text{rel}} + \langle W_0, \hat{M} - M \rangle + \sum_{j=1}^n \langle W_j, \hat{M} - M \rangle x_j \\ &\geq \gamma_{\text{rel}} - (1 + \xi) \max_{j=0}^n \|W_j\| \|\hat{M} - M\| \geq \nu_0(1 + \xi)r. \end{aligned}$$

Since x was an arbitrary (ROB)-feasible point we have proved (6.2). If x only satisfies $L(x, \Delta_0) < 0$, the same line of reasoning with $\|x\|_1 \leq \eta$ gives (6.3). $\square$

If B denotes the unit ball in $\mathbb{R}^m$, we observe that $\gamma_{\text{opt}}(t) = \min \{ \langle c, x \rangle : x \in \mathbb{R}^n, g(x, t) \leq 0 \}$, where

$$g(x, t) = \max_{(\Delta, w) \in \Delta \times B} w^T L(x, \Delta) w + t \| [F(\Delta) - F(\Delta_0)] w \|^2.$$

Since $(x, t) \rightarrow w^T L(x, \Delta)w + t\| [F(\Delta) - F(\Delta_0)]w \|^2$ is affine, and since the derivative with respect to t equals $\| [F(\Delta) - F(\Delta_0)]w \|^2$, we can invoke Danskin's theorem [5, p. 245] to infer that $g(\cdot, \cdot)$ is continuous and convex and that the right derivative of $g(x, \cdot)$ at $t = 0$ is given by

$$g'(x, 0) = \max_{(\Delta, w) \in M(x)} \| [F(\Delta) - F(\Delta_0)]w \|^2,$$

where $M(x) = \{(\Delta, w) \in \mathbf{\Delta} \times \mathbf{B} : w^T L(x, \Delta)w = g(x, 0)\}$. Now note $g(x_{\text{opt}}, 0) = 0$ (since c is nonzero and the objective $\langle c, x \rangle$ is linear in x) and $L(x_{\text{opt}}, \Delta) \leq 0$ for all $\Delta \in \mathbf{\Delta}$. Therefore $M(x_{\text{opt}}) = \{(\Delta, w) \in \mathbf{\Delta} \times \mathbf{B} : L(x_{\text{opt}}, \Delta)w = 0\}$. At this point we exploit the hypothesis to conclude that $g'(x_{\text{opt}}, 0) = 0$.

Since (ROB) was assumed feasible, there exists $x_0 \in \mathbb{R}^n$ with $g(x_0, 0) < 0$, and we can hence fix some $t_0 > 0$ and $\delta > 0$ such that $g(x_0, t) \leq -\alpha_0$ for all $t \in [0, t_0]$. For these parameters, the set $\{x \in \mathbb{R}^n : g(x, t) \leq 0\}$ is hence nonempty, and it is also compact since it is contained in X . Therefore $\gamma_{\text{opt}}(t)$ satisfies $\gamma_{\text{opt}}(0) \leq \gamma_{\text{opt}}(t) \leq \gamma_{\text{opt}}(t_0) < \infty$, the set of optimizers $X_{\text{opt}}(t)$ is nonempty and compact, and the set of Lagrange multipliers $\Lambda(t)$ is nonempty. Moreover, for all $t \in [0, t_0]$, $\lambda_t \in \Lambda(t)$, $x_t \in X(t)$ we infer with the usual saddle-point property of the Lagrangian that

$$\begin{aligned} -\alpha_0 \lambda_t &\geq \lambda_t g(x_0, t) = \lambda_t g(x_0, t) + \langle c, x_0 \rangle - \langle c, x_0 \rangle \\ &\geq \lambda_t g(x_t, t) + \langle c, x_t \rangle - \langle c, x_0 \rangle = \gamma_{\text{opt}}(t) - \langle c, x_0 \rangle \geq \gamma_{\text{opt}}(0) - \langle c, x_0 \rangle. \end{aligned}$$

With $\alpha = (\langle c, x_0 \rangle - \gamma_{\text{opt}}(0))/\alpha_0 > 0$ this implies $0 \leq \lambda_t \leq \alpha$ for all $\lambda_t \in \Lambda(t)$, $t \in [0, t_0]$.

Now choose an arbitrary $\epsilon > 0$. Since $g'(x_{\text{opt}}, 0) = 0$, there exists $\delta \in (0, t_0]$ with $g(x_{\text{opt}}, t) \leq g(x_{\text{opt}}, 0) + t\epsilon/\alpha$ for all $t \in (0, \delta]$. With $t \in (0, \delta]$, $x_t \in X(t)$, $\lambda_t \in \Lambda(t)$ we exploit again the saddle-point property to obtain

$$\begin{aligned} \gamma_{\text{opt}}(t) &= c^T x_t + \lambda_t g(x_t, t) \leq c^T x_{\text{opt}} + \lambda_t g(x_{\text{opt}}, t) \leq c^T x_{\text{opt}} + \lambda_t [g(x_{\text{opt}}, 0) + t\epsilon/\alpha] \\ &\leq c^T x_{\text{opt}} + \lambda_t g(x_{\text{opt}}, 0) + t\epsilon \leq c^T x_{\text{opt}} + \lambda_0 g(x_{\text{opt}}, 0) + t\epsilon \leq \gamma_{\text{opt}}(0) + t\epsilon, \end{aligned}$$

and hence $(\gamma_{\text{opt}}(t) - \gamma_{\text{opt}}(0))/t \leq \epsilon$. This proves $\gamma'_{\text{opt}}(0) \leq 0$, and hence $\gamma'_{\text{opt}}(0) = 0$. $\square$

Acknowledgments. Part of this research was performed during a one-month visit to the Department of Systems Science at Kyoto University in Japan. I would like to express my sincere gratitude to Prof. Toshiharu Sugie for his kind invitation, with many thanks to all the members of his group, and in particular to Dr. Kenji Fujimoto, for their truly excellent hospitality. Moreover, my thanks go to Shinji Hara for fruitful discussions and to Ted Iwasaki for pointing out the incompleteness of the proof of Lemma A.7 in the conference version of this paper. Finally, many thanks go to two anonymous reviewers for their constructive criticism.

REFERENCES

- [1] V. BALAKRISHNAN AND L. VANDENBERGHE, *Semidefinite programming duality and linear time-invariant systems*, IEEE Trans. Automat. Control, 48 (2003), pp. 30–41.
- [2] B. BARMISH, *Necessary and sufficient conditions for quadratic stability of an uncertain system*, J. Optim. Theory Appl., 46 (1985), pp. 399–408.
- [3] A. BEN-TAL AND A. NEMIROVSKI, *Lectures on Modern Convex Optimization: Analysis, Algorithms, and Engineering Applications*, MPS/SIAM Ser. Optim. 2, SIAM, Philadelphia, 2001.
- [4] A. BEN-TAL AND A. NEMIROVSKI, *On tractable approximations of uncertain linear matrix inequalities affected by interval uncertainty*, SIAM J. Optim., 12 (2002), pp. 811–833.

- [5] D. BERTSEKAS, *Convex Analysis and Optimization*, Athena Scientific, Belmont, MA, 2003.
- [6] C. BOUSSIOS AND E. FERON, *Estimating the conservatism of Popov's criterion for real parametric uncertainties*, Systems Control Lett., 31 (1997), pp. 173–183.
- [7] S. BOYD, V. BALAKRISHNAN, E. FERON, AND L. EL GHAOU, *Control system analysis and synthesis via linear matrix inequalities*, in Proceedings of the American Control Conference, San Francisco, CA, 1993, pp. 2147–2154.
- [8] G. CHESI, A. GARULLI, A. TESI, AND A. VICINO, *An LMI-based approach for characterizing the solution set of polynomial systems*, in Proceedings of the 39th IEEE Conference on Decision and Control, Sydney, Australia, 2000, pp. 1501–1506.
- [9] G. COXSON AND C. DEMARCO, *The computational complexity of approximating the minimal perturbation scaling to achieve instability in an interval matrix*, Math. Control Signals Systems, 7 (1994), pp. 279–292.
- [10] E. DE KLERK AND D. V. PASECHNIK, *Approximation of the stability number of a graph via copositive programming*, SIAM J. Optim., 12 (2002), pp. 875–892.
- [11] J. DE LOERA AND F. SANTOS, *An effective version of Pólya's theorem on positive definite forms*, J. Pure Appl. Algebra, 108 (1996), pp. 219–230.
- [12] M. DETTORI AND C. SCHERER, *Robust stability analysis for parameter dependent systems using full block S-procedure*, in Proceedings of the 37th IEEE Conference on Decision and Control, Tampa, FL, 1998, pp. 2798–2799.
- [13] L. EL GHAOU, F. OUSTRY, AND H. LEBRET, *Robust solutions to uncertain semidefinite programs*, SIAM J. Optim., 9 (1998), pp. 33–52.
- [14] A. FEINTUCH AND A. MARKUS, *The Toeplitz-Hausdorff theorem and robust stability theory*, Math. Intelligencer, 21 (1999), pp. 33–37.
- [15] M. FU, *The real μ is hardly approximable*, IEEE Trans. Automat. Control, 42 (1997), pp. 1286–1288.
- [16] M. FU AND S. DASGUPTA, *Computational complexity of real structured singular value in l_p setting*, IEEE Trans. Automat. Control, 45 (2000), pp. 2173–2176.
- [17] P. GAHINET AND P. APKARIAN, *A linear matrix inequality approach to H_∞ control*, Int. J. of Robust and Nonlinear Control, 4 (1994), pp. 421–448.
- [18] D. GOLDFARB AND K. SCHEINBERG, *Interior point trajectories in semidefinite programming*, SIAM J. Optim., 8 (1998), pp. 871–886.
- [19] D. HENRION AND J. B. LASSERRE, *Detecting global optimality and extracting solutions in gloptipoly*, in Positive Polynomials in Control, Lecture Notes in Control and Inform. Sci., D. Henrion and A. Garulli, eds., Springer-Verlag, Berlin, Heidelberg, New York, 2005.
- [20] D. HENRION AND G. MEINSMA, *Rank-one LMIs and Lyapunov's inequality*, IEEE Trans. Automat. Control, 46 (2001), pp. 1285–1288.
- [21] R. HORN AND C. JOHNSON, *Matrix Analysis*, Cambridge University Press, New York, 1985.
- [22] T. IWASAKI, *Robust stability analysis with quadratic separator: Parametric time-varying uncertainty case*, in Proceedings of the 36th IEEE Conference on Decision and Control, San Diego, CA, 1997, pp. 4880–4885.
- [23] T. IWASAKI AND S. HARA, *Well-posedness of feedback systems: Insights into exact robustness analysis*, in Proceedings of the 35th IEEE Conference on Decision and Control, Kobe, Japan, 1996, pp. 1863–1868.
- [24] T. IWASAKI AND S. HARA, *Well-posedness of feedback systems: Insights into exact robustness analysis and approximate computations*, IEEE Trans. Automat. Control, 43 (1998), pp. 619–630.
- [25] T. IWASAKI, S. HARA, AND H. YAMAUCHI, *Dynamical system design from a control perspective: Finite frequency positive-realness approach*, IEEE Trans. Automat. Control, 48 (2003), pp. 1337–1354.
- [26] T. IWASAKI AND G. SHIBATA, *LPV system analysis via quadratic separator for uncertain implicit systems*, IEEE Trans. Automat. Control, 46 (2001), pp. 1195–1208.
- [27] J. B. LASSERRE, *Global optimization with polynomials and the problem of moments*, SIAM J. Optim., 11 (2001), pp. 796–817.
- [28] G. MEINSMA, G. SHRIVASTAVA, AND M. FU, *A dual formulation of mixed μ and on the losslessness of (D, G) scaling*, IEEE Trans. Automat. Control, 42 (1997), pp. 1032–1036.
- [29] A. PACKARD AND J. DOYLE, *The complex structured singular value*, Automatica, 29 (1993), pp. 71–109.
- [30] P. PARRILO, *Structure Semidefinite Programs and Semialgebraic Geometry Methods in Robustness and Optimization*, Ph.D. thesis, California Institute of Technology, Pasadena, CA, 2000.
- [31] P. PARRILO AND B. STURMFELS, *Minimizing polynomial functions*, in Algorithmic and Quantitative Real Algebraic Geometry, DIMACS Ser. in Discrete Math. Theoret. Computer Sci. 60, AMS, Providence, RI, 2003, pp. 83–99.

- [32] D. PEAUCELLE, *Unified Formulation for Robust Analysis and Synthesis with Parameter Dependent Lyapunov Functions*, PH.D. thesis, Université Paul Sabatier, Toulouse cedex, France, 2000.
- [33] B. T. POLYAK AND R. TEMPO, *Probabilistic robust design with linear quadratic regulators*, Systems Control Lett., 43 (2001), pp. 343–353.
- [34] V. POWERS AND B. REZNICK, *A new bound for Pólya's theorem with applications to polynomials positive on polyhedra*, J. Pure Appl. Algebra, 164 (2001), pp. 221–229.
- [35] A. RANTZER, *On the Kalman-Yakubovich-Popov lemma*, Systems Control Lett., 28 (1996), pp. 7–10.
- [36] C. SCHERER, *Mixed H_2/H_∞ control for time-varying and linear parametrically-varying systems*, Internat. J. Robust Nonlinear Control, 6 (1996), pp. 929–952.
- [37] C. SCHERER, *A full block S-procedure with applications*, in Proceedings of the 36th IEEE Conference on Decision and Control, San Diego, CA, 1997, pp. 1510–1515.
- [38] C. SCHERER, *LPV control and full block multipliers*, Automatica, 37 (2001), pp. 361–375.
- [39] C. SCHERER, *When are multiplier relaxations exact?*, in Proceedings of the 4th IFAC Symposium on Robust Control Design, Milano, Italy, 2003.
- [40] C. SCHERER AND C. HOL, *Asymptotically exact relaxations for robust LMI problems based on matrix-valued sum-of-squares*, in Proceedings of the 16th Mathematical Theory of Networks and Systems Conference 2004, Leuven, Belgium, 2004.
- [41] J. STURM AND S. ZHANG, *On cones of nonnegative quadratic functions*, Math. Oper. Res., 28 (2003), pp. 246–267.
- [42] J. TERLOUW, P. LAMBRECHTS, S. BENNANI, AND M. STEINBUCH, *Parametric uncertainty modeling using LFT's*, in Proceedings of the AIAA GNC Conference, Hilton Head, SC, 1992.
- [43] O. TOKER AND H. OZBAY, *On the NP-hardness of the purely complex μ computation, analysis/synthesis, and some related problems in multidimensional systems*, in Proceedings of the American Control Conference, Seattle, WA, 1995, pp. 447–451.
- [44] A. TROFINO AND C. DE SOUZA, *Bi-quadratic stability of uncertain linear systems*, in Proceedings of the 38th IEEE Conference on Decision and Control, Phoenix, AZ, 1999, pp. 5016–5021.
- [45] H. D. TUAN AND P. APKARIAN, *Monotonic relaxations for robust control: New characterizations*, IEEE Trans. Automat. Control, 47 (2002), pp. 378–384.
- [46] K. ZHOU, J. DOYLE, AND K. GLOVER, *Robust and Optimal Control*, Prentice-Hall, Upper Saddle River, NJ, 1996.

COMPUTING MOMENTS OF FIRST PASSAGE TIMES TO A SUBSET OF STATES IN MARKOV CHAINS*

TUĞRUL DAYAR[†] AND NAIL AKAR[‡]

Abstract. This paper presents a relatively efficient and accurate method to compute the moments of first passage times to a subset of states in finite ergodic Markov chains. With the proposed method, the moment computation problem is reduced to the solution of a linear system of equations with the right-hand side governed by a novel recurrence for computing the higher-order moments. We propose using a form of the Grassmann–Taksar–Heyman (GTH) algorithm to solve these linear equations. Due to the form of the linear systems involved, the proposed method does not suffer from the drawbacks associated with GTH in a row-wise sparse implementation.

Key words. Markov chains, unsafe states, first passage times, mean, variance, moments, Grassmann–Taksar–Heyman algorithm

AMS subject classifications. 60J10, 60J27, 65F05, 65F10

DOI. 10.1137/S0895479804442462

1. Introduction. Let the state space of a Markov chain (MC) be partitioned into the subsets of *safe* and *unsafe* states. The question “How much time on the average will it take the system to reach an unsafe state if it starts in a safe state and what is its variance?” is to be answered when reliability is of concern in the system that is under investigation. Such reliability measures need to be known, for instance, in repairable computer systems to forecast the time to an unavailable system [22, pp. 328–330] or in railway systems [16, p. 12] where the passing of a train to a prohibited track on the red light is considered to be unsafe.

Determining the mean and variance of first passage times (or hitting times) in finite ergodic MCs [10, 11, 12] has centered around computing the stationary vector of the MC at hand and the fundamental matrix [14] (or the group generalized inverse [17]). In [23, p. 10], an iterative technique that does not require the computation of the stationary vector, the fundamental matrix, or the group generalized inverse had been proposed. Recently, another technique appeared in [7] to compute these measures through Laplace transforms. Unfortunately, all of these techniques require dealing with systems in the order of the state space size. However, when the mean and variance of first passage times only to a subset of states in the MC are sought, it is unnecessary to consider systems as large as the state space size. In fact, it suffices to solve a linear system with as many equations as the number of safe states for two different right-hand sides.

In this paper, we study the more general problem of computing *all* moments of first passage times to a subset of states in finite ergodic MCs. The proposed method reduces the computation of the first k moments to the solution of k linear systems with a common coefficient matrix but different right-hand sides (one for each moment), where the right-hand sides are governed by a novel recurrence involving the universally

*Received by the editors March 23, 2004; accepted for publication (in revised form) by D. Bini February 12, 2005; published electronically October 31, 2005.

<http://www.siam.org/journals/simax/27-2/44246.html>

[†]Department of Computer Engineering, Bilkent University, TR-06800 Bilkent, Ankara, Turkey (tugrul@cs.bilkent.edu.tr).

[‡]Department of Electrical and Electronics Engineering, Bilkent University, TR-06800 Bilkent, Ankara, Turkey (akar@ee.bilkent.edu.tr).

well-known Pascal's triangle. We also use a form of the Grassmann–Taksar–Heyman (GTH) algorithm [6] for solving such linear systems. Due to the form of the linear systems involved, the proposed method does not suffer from the drawbacks associated with GTH in a row-wise sparse implementation.

The remainder of the paper is organized as follows. Section 2 provides an overview of the existing methods for the computation of the mean and variance of first passage times to a subset of states in MCs. Section 3 introduces the more efficient method which is based on the novel recurrence to compute all moments of first passage times to the subset of states. Section 4 shows how this method can be implemented relatively accurately using a form of the GTH algorithm and discusses a row-wise sparse implementation. Section 5 includes numerical experiments with MCs from the literature, and section 6 concludes the paper. The recurrence for computing the moments of first passage times in discrete-time MCs (DTMCs) is proved in the appendix.

In the following, P denotes the transition probability matrix of a finite ergodic MC of order n with steady state probability (row) vector π . As the steady state vector, π is also the stationary vector and satisfies $\pi P = \pi$ with the normalization condition $\sum_{i=1}^n \pi_i = 1$. The proposed method in this paper concerns DTMCs but can be easily extended to continuous-time MCs (CTMCs) as we do in one of the test problems.

2. Mean and variance of first passage times to a subset of states. Without loss of generality, let us assume that P has the following block partitioning:

$$(1) \quad P = \begin{pmatrix} P_{S,S} & P_{S,U} \\ P_{U,S} & P_{U,U} \end{pmatrix},$$

where S and U are, respectively, the subsets of *safe* and *unsafe* states; let the number of states in these subsets be, respectively, n_S and n_U . Otherwise, P can always be symmetrically permuted to the block form in (1). Here $P_{S,S}$ is the square submatrix of order n_S obtained from P by deleting the rows and columns that correspond to the states in U . Let the stationary vector be partitioned conformally as in $\pi = (\pi_S \ \pi_U)$, where π_S and π_U are the subvectors of π associated with the states in S and U , respectively.

The mean and variance of first passage times to subset U from states in subset S can be determined by first computing π using the GTH algorithm [6] and forming the aggregated matrix

$$K = \begin{pmatrix} P_{S,S} & P_{S,U}e \\ \frac{\pi_U}{\|\pi_U\|_1} P_{U,S} & \frac{\pi_U}{\|\pi_U\|_1} P_{U,U}e \end{pmatrix},$$

where e is a column vector of ones with appropriate length. Since $\pi_K = (\pi_S \ \|\pi_U\|_1)$ is the stationary vector of K , its fundamental matrix [14, p. 75] can be computed from

$$(2) \quad Z = (I - K + e\pi_K)^{-1},$$

where I is the identity matrix of appropriate order. Now let us assume that $\text{diag}(A)$ returns a column vector formed of the diagonal elements of matrix A , and $\text{diag}(a)$ returns a diagonal matrix having the elements of vector a along its diagonal. Then the fundamental matrix enables us to compute the first and second moment matrices,

namely $F^{(1)}$ and $F^{(2)}$, respectively, of first passage times among states in K from [14, p. 79]

$$F^{(1)} = (I - K + e \operatorname{diag}(Z)^T)D$$

and [14, p. 83]

$$F^{(2)} = F^{(1)}(2 \operatorname{diag}(\operatorname{diag}(Z))D - I) + 2(ZF^{(1)} - e \operatorname{diag}(ZF^{(1)})^T),$$

where D is the diagonal matrix having reciprocals of the elements of π_K along its diagonal. The (i, j) th element of $F^{(1)}$, denoted by $f_{i,j}^{(1)}$, is the mean first passage time from state i to state j in K . Similarly, the (i, j) th element of $F^{(2)}$, denoted by $f_{i,j}^{(2)}$, is the second moment of the first passage time from state i to state j in K .

In terms of the entries of $F^{(1)}$ and $F^{(2)}$, the mean and variance of first passage times from state $i \in \mathcal{S}$ to $\mathcal{U}$ are given respectively by

$$(3) \quad m_i^{(1)} = f_{i,n_{\mathcal{S}+1}}^{(1)}$$

and

$$(4) \quad v_i = f_{i,n_{\mathcal{S}+1}}^{(2)} - (f_{i,n_{\mathcal{S}+1}}^{(1)})^2,$$

where $f_{i,n_{\mathcal{S}+1}}^{(2)}$ is the second moment of first passage time from state $i \in \mathcal{S}$ to $\mathcal{U}$ and will be denoted by $m_i^{(2)}$. We remark that the time complexity of obtaining the two measures in (3) and (4) with the described approach is $O(n^3)$ due to the necessity to compute π .

It is well known that the computation of the fundamental matrix through matrix inversion as in (2) is prone to numerical inaccuracies. This affects the accuracy of the first and second moment matrices of first passage times that are computed using the fundamental matrix. Through a series of papers [10, 11, 12], Heyman and O'Leary have identified this situation and have provided algorithms that aim at remedying it. Their idea is based on using the GTH algorithm [6] to factorize a singular linear system involving the MC at hand and to compute the stationary vector accurately.

Since we are interested in computing the moments of first passage times to a subset of states, in the next section we present a method that does not require the stationary vector of the MC to be computed and has a time complexity that depends on $n_{\mathcal{S}}$, as opposed to n with the existing techniques. After this presentation, we show how a form of the GTH algorithm can be used to provide relatively accurate results in solving the nonsingular linear systems that arise in this context.

3. A more efficient method. Let X be the random variable associated with the first passage times to subset $\mathcal{U}$ conditioned on an initial probability vector τ in the subset of safe states, $\mathcal{S}$. Given $T = P_{\mathcal{S},\mathcal{S}}$, we have [15, p. 50]

$$(5) \quad E[X(X-1) \cdots (X-k+1)] = \tau \gamma^{(k)},$$

where

$$(6) \quad \gamma^{(k)} = k!(I - T)^{-k} T^{k-1} e$$

is defined as the k th factorial moment vector of first passage times from states in $\mathcal{S}$ to $\mathcal{U}$ for $k > 0$. This result follows from probabilistic arguments and is based on

aggregating the states in $\mathcal{U}$ to a single state and constructing an absorbing MC in which all states in $\mathcal{S}$ are transient and the aggregated state is the single absorbing state (see [15, pp. 47–50]).

When $k = 1$, (5) becomes

$$(7) \quad E[X] = \tau(I - T)^{-1}e.$$

Then the mean first passage time from state $i \in \mathcal{S}$ to $\mathcal{U}$, denoted by $m_i^{(1)}$, follows from (7) by substituting e_i^T (i.e., i th row of I) for τ . More generally, the first moment vector $m^{(1)}$, whose i th element is $m_i^{(1)}$, can be obtained from

$$(8) \quad (I - T)m^{(1)} = e.$$

We remark that $(I - T)$ is a nonsingular M-matrix¹ and $(I - T)^{-1} \geq 0$ [2]. Hence, $m^{(1)} = (I - T)^{-1}e$, simply the row sums of the nonnegative matrix $(I - T)^{-1}$. This nonnegative matrix is referred to as the fundamental matrix of an absorbing MC in which the states in $\mathcal{S}$ are transient and those in $\mathcal{U}$ are ergodic [14, p. 46]. In fact, (8) is available in [14, p. 51] and has been derived following a different approach.

When $k = 2$, (5) and (6) yield

$$(9) \quad E[X^2] = E[X] + 2\tau(I - T)^{-2}Te.$$

Defining $m^{(2)}$ as the second moment vector of first passage times from states in $\mathcal{S}$ to $\mathcal{U}$, we obtain

$$(10) \quad (I - T)m^{(2)} = 2m^{(1)} - e$$

from (9) after some algebra and again considering all rows of I for τ . Although not exactly in the form of (10), an equivalent representation for $m^{(2)}$ is available in [14, p. 51].

Once $m^{(2)}$ is available, the variance of the first passage time from state i in $\mathcal{S}$ to $\mathcal{U}$ can be computed from

$$(11) \quad v_i = m_i^{(2)} - (m_i^{(1)})^2 \quad \text{for } i \in \mathcal{S}.$$

Equations (8) and (10) provide an alternative way (to that in section 2) of computing the first and second moment vectors of first passage times to a subset of states in finite ergodic MCs. Observe that neither a complete factorization based on P nor the stationary vector of P is required to compute these measures. In conclusion, one needs to factorize $(I - P_{\mathcal{S},\mathcal{S}})$ once, compute $m^{(1)}$, and then compute $m^{(2)}$ using $m^{(1)}$, e , and the factors of $(I - P_{\mathcal{S},\mathcal{S}})$.

In fact, it is possible to compute (higher) moments of first passage times from states in $\mathcal{S}$ to $\mathcal{U}$ using the recurrence

$$(12) \quad Sm^{(i+1)} = \sum_{j=0}^i (-1)^{i-j} \binom{i+1}{j} m^{(j)} \quad \text{for } i \geq 0 \quad \text{with } m^{(0)} = e,$$

where $S = I - P_{\mathcal{S},\mathcal{S}}$. Here $\binom{i+1}{j}$ denotes the binomial coefficient in Pascal's triangle that is read “ $(i+1)$ choose j ” [5, p. 153]. To the best of our knowledge, the recurrence in (12) has not been given before and is proved in the appendix.

¹Any matrix A of the form $A = sI - B$ with $s > 0$ and $B \geq 0$ for which $s \geq \rho(B)$, the spectral radius of B .

In the next section we show how the more efficient method which essentially solves linear systems of the form $Sy = c$ in (12), where $S = I - P_{S,S}$, can be implemented relatively accurately using a form of the GTH algorithm and discuss a row-wise sparse implementation.

4. Implementation issues. In order to compute (higher) moment vectors of first passage times, we use Algorithm 1 which follows from (12), which has the matrix $S = I - P_{S,S}$ on the left-hand side and a linear combination of e and the previous moment vectors on the right-hand side. We LU factorize S once at the outset using GTH as discussed in the next subsection. Here, L and U are, respectively, the unit-lower triangular factor and the upper-triangular factor of S . The very reason we use GTH is to improve the accuracy in the computed factors when S is close to being singular. Otherwise, S could very well be factorized using Gaussian elimination (GE) but in no way should be inverted and passed to the right-hand side in (12). Finally, the integer coefficients of the linear combination on the right-hand side of (12) may be computed easily by observing the alternating signs and using the well-known identity [5, p. 158]

$$\binom{i+1}{j} = \binom{i}{j-1} + \binom{i}{j}$$

that binomial coefficients satisfy.

ALGORITHM 1.

Computing the k th moment vector of first passage times to a subset of states in DTMCs.

LU factorize S using GTH;

$a_0 = 1; a_1 = -1; m^{(0)} = e;$

Solve for $m^{(1)}$ in $LUm^{(1)} = m^{(0)};$

For $i = 1$ up to $k - 1$ {

$a_{i+1} = -1;$

For $j = i$ down to 1

$a_j = a_{j-1} - a_j;$

$a_0 = -a_0;$

Solve for $m^{(i+1)}$ in $LUm^{(i+1)} = \sum_{j=0}^i a_j m^{(j)};$

}

4.1. Factorization using GTH. The GTH algorithm due to Grassmann, Tak-sar, and Heyman [6] emerges from probabilistic arguments and shows how one can compute the stationary vector of an MC using only nonnegative numbers and avoiding subtraction operations. This algorithm achieves significantly greater accuracy than other algorithms in the literature [8, 20]. The entrywise relative error in the stationary vector computed using the GTH algorithm is shown to be small, independent of the structure of the matrix and the magnitudes of its elements [19].

A form of the GTH algorithm is used in the disaggregation phase of the iterative aggregation-disaggregation (IAD) algorithm geared towards nearly completely decomposable (NCD) MCs [4] (we will return to NCD MCs in the section on numerical results). Within that phase, linear systems of the form $y^T S = c^T$ (or the transposed form $S^T y = c$) need to be solved for y . To each such system one more unknown and one more equation are added so as to have a homogeneous linear system with a coefficient matrix that is a singular M-matrix. Here we follow a slightly different approach

and construct a nonhomogeneous but consistent linear system with a singular coefficient matrix.

Consider $Ax = b$, where

$$(13) \quad A = \begin{pmatrix} I - P_{S,S} & -P_{S,U}e \\ w^T & -w^T e \end{pmatrix}, \quad x = \begin{pmatrix} y \\ 0 \end{pmatrix}, \quad \text{and } b = \begin{pmatrix} c \\ v \end{pmatrix}.$$

Observe that A has zero row sums since $(I - P_{S,S})e = e - P_{S,S}e = P_{S,U}e$, and A is a singular M-matrix with rank n_S . The values of the vector w and the scalar v are irrelevant since in the LU factorization of the singular matrix A , the last row of the upper-triangular factor turns out to be zero.

The properties of a singular M-matrix coupled with the idea of using nonnegative arithmetic in the GTH algorithm suggests the following modification to GE when upper-triangularizing A . At each step of GE, rather than computing the pivot element in the usual way, one can correct the pivot by replacing it with the negated sum of the off-diagonal elements in the unreduced part of the same row as the pivot. This follows from the fact that at each step of GE, the unreduced square submatrix that is to be updated turns out to be a singular M-matrix having zero row sums. We remark that other than providing better roundoff properties, this modification has the additional advantage of being able to be carried out completely in row-wise sparse format with delayed row updates [23]. In particular, the computation of pivots is in no way orthogonal to the access scheme of row-wise sparse format as opposed to the situation with the systems solved in IAD (see section 4 in [4]) or the original formulation of the GTH algorithm [6]. It is this modified GE which we refer to as the GTH implementation in this paper.

The GTH reduction needs to be performed for only $(n_S - 1)$ steps. This amounts to premultiplying A by a unit lower-triangular multiplier matrix as in

$$\begin{pmatrix} L^{-1} & 0 \\ 0^T & 1 \end{pmatrix} \begin{pmatrix} I - P_{S,S} & -P_{S,U}e \\ w^T & -w^T e \end{pmatrix} = \begin{pmatrix} U & -L^{-1}P_{S,U}e \\ w^T & -w^T e \end{pmatrix}$$

so that $(I - P_{S,S}) = LU$, where L is unit lower-triangular and U is upper-triangular. The product $-L^{-1}P_{S,U}e$ needs to be computed along the way to enable the correction of the pivots. Then y can be obtained from $LUy = c$ through forward and backward substitutions (cf. (13)). Note that both L and U need to be stored since we will solve for (at least) two right-hand sides.

4.2. Analysis. In this subsection we discuss issues pertaining to the row-wise sparse factorization of $S = I - P_{S,S}$ using GTH and the sparse implementation of Algorithm 1. Let nz_S , nz_L , and nz_U denote, respectively, the number of nonzeros in S , the strictly lower-triangular part of L (since 1's along its diagonal need not be stored), and U . Also let k denote the order of the highest moment vector to be computed.

The nonzeros in nz_L and nz_U depend on the sparsity pattern of S , and there are myriad ways discussed in the literature to reduce the fill-in that results from its sparse factorization. However, a discussion on fill reducing orderings and their implications is beyond the scope of this paper. Therefore, assuming that S is not reordered and a row-wise sparse format is used, one needs $(nz_L + nz_U)$ real and $(nz_L + nz_U + n_S + 1)$ integer storage space to store and index the nonzeros in the L and U factors of S [23, pp. 80–81]. Since the sparse factorization can be performed through delayed row updates (see section 4.1), it suffices to have an additional real row vector of length n_S

to expand the row to be updated, do the reduction, compact the reduced row, and store it. When GTH is used to carry out the factorization, a real vector of length n_S which sums the rows of S to zero (i.e., last column of A in (13)) should also be allocated. This real vector of length n_S is the only extra storage space compared to using GE.

Algorithm 1 requires k real vectors of length n_S to store the moment vectors and one real vector of length n_S to store the right-hand side in (12). This right-hand side is a linear combination with well-defined integer coefficients of the previous moment vectors and does not pose problems from a computational point of view other than overflow as long as the integer coefficients are stored in a real vector of length $(k + 1)$ (see the coefficients a_i). Hence, the total storage space required by Algorithm 1 in addition to the sparse factors of S is $(k + 1)$ real vectors of length n_S and one real vector of length $(k + 1)$. Note that since the absolute values of the coefficients in (12) follow from Pascal's triangle and that this triangle is symmetric, it is possible to do by storing only half of the coefficients.

The number of floating-point arithmetic operations (flops) to perform sparse GTH on S is equal to the number of flops to perform sparse GE on S , minus $2(n_S - 1)$ flops for those that do not get executed in GTH due to the new way of computing the pivots (note that the pivot in the first row is not updated) plus nz_U flops due to computing each pivot by summing the nonzeros in the strictly upper-triangular part of U and the element of the updated extra column in the same row as the pivot. Hence, the only disadvantage of GTH with respect to GE in this setting is the extra $(nz_U - 2n_S + 2)$ flops. Assuming that nz_U is linear in n_S for the sparse matrix S , this extra cost in GTH is negligible since the number of flops to perform sparse GE on S will be quadratic in n_S . In passing to the analysis of Algorithm 1, we remark that each pair of forward and backward substitutions (with the matrices L and U , respectively) for a different right-hand side requires $2(nz_L + nz_U) - n_S$ flops.

In the i th pass, the outer for-loop in Algorithm 1 requires the computation of $(i + 1)$ coefficients (which takes i floating-point subtractions on integers), the computation of a right-hand side vector (which takes $2(i + 1)n_S$ flops), and a pair of forward and backward substitutions (which take $2(nz_L + nz_U) - n_S$ flops). For a total of $(k - 1)$ passes, these amount to $(k - 1)[2(nz_L + nz_U) + n_S(k + 1) + k/2]$ flops.

In the following section we provide results of numerical experiments with two NCD MCs and a reliability problem from the literature.

5. Numerical results. In this section, we present results with three problems to illustrate the proposed method. We have implemented the proposed method in C using IEEE floating-point arithmetic with and without GTH [21]. We refer to the latter approach as the proposed method with GE. Using these methods, we have computed the moment vectors of first passage times to a subset of states in each MC. In all problems, the results of the proposed method with GTH and GE in double precision floating-point arithmetic (i.e., about sixteen decimal digits of precision) agree to more than seven decimal digits. Hence, we have taken the first seven decimal digits of these results as exact results and have compared the results of the proposed method with GTH and GE in single precision floating-point arithmetic (i.e., about seven decimal digits of precision) against these exact results. Furthermore, the condition number of each coefficient matrix is reported so as to indicate the inherent difficulty associated with computing the moment vectors accurately. Note that 10^{38} is about the size of the largest positive number in single precision floating-point arithmetic, and any attempt to compute moment vectors having elements larger than this value will be

futile. Assuming that k is the highest-order moment vector that can be computed in single precision floating-point arithmetic for a given problem, in order to save space we report only the first, second, and k th moment vectors. Since k in each of the problems turns out to be less than or equal to twenty-two, the computation of binomial coefficients in single precision arithmetic does not pose problems. In other words, the largest entry in the twenty-second row of Pascal's triangle, $22 \text{ choose } 11$, is equal to 705,432 and can be represented in seven decimal digits of precision.

The first two problems are also considered in [11] and are examples of NCD MCs [3, 18], which are irreducible stochastic matrices that can be symmetrically permuted to the block form

$$(14) \quad P_{n \times n} = \begin{pmatrix} P_{11} & P_{12} & \cdots & P_{1N} \\ P_{21} & P_{22} & \cdots & P_{2N} \\ \vdots & \vdots & \ddots & \vdots \\ P_{N1} & P_{N2} & \cdots & P_{NN} \end{pmatrix} \begin{pmatrix} n_1 \\ n_2 \\ \vdots \\ n_N \end{pmatrix}$$

where nonzero elements in the off-diagonal blocks are small compared with those in the diagonal blocks [23, p. 286]. Let $P = \text{diag}(P_{11}, P_{22}, \dots, P_{NN}) + E$. The diagonal blocks P_{ii} are square, of order n_i , with $n = \sum_{i=1}^N n_i$. The quantity $\|E\|_\infty$ is referred to as the degree of coupling and is taken to be a measure of the decomposability of P . Such matrices are known to be numerically challenging. The third problem is the reliability model that appears in [22].

5.1. Test problem 1. The first problem is the 8×8 matrix due to Courtois [3] with all row sums equal to 1 and given by

$$P = \begin{pmatrix} 0.85 & 0 & 0.149 & 0.0009 & 0 & 0.00005 & 0 & 0.00005 \\ 0.1 & 0.65 & 0.249 & 0 & 0.0009 & 0.00005 & 0 & 0.00005 \\ 0.1 & 0.8 & 0.0996 & 0.0003 & 0 & 0 & 0.0001 & 0 \\ 0 & 0.0004 & 0 & 0.7 & 0.2995 & 0 & 0.0001 & 0 \\ 0.0005 & 0 & 0.0004 & 0.399 & 0.6 & 0.0001 & 0 & 0 \\ 0 & 0.00005 & 0 & 0 & 0.00005 & 0.6 & 0.2499 & 0.15 \\ 0.00003 & 0 & 0.00003 & 0.00004 & 0 & 0.1 & 0.8 & 0.0999 \\ 0 & 0.00005 & 0 & 0 & 0.00005 & 0.1999 & 0.25 & 0.55 \end{pmatrix}.$$

This matrix has the three subsets of NCD states, $\{1, 2, 3\}$, $\{4, 5\}$, $\{6, 7, 8\}$, for a degree of coupling of 0.001.

Let us take $\mathcal{S} = \{1, 2, 3\}$, which corresponds to an NCD subset of states. For this choice of safe states, the 2-norm condition number of the coefficient matrix is $\text{cond}_2(I - P_{\mathcal{S}, \mathcal{S}}) = 1.5 \times 10^3$ from MATLAB. For this problem the first ten moment vectors can be computed in single precision floating-point arithmetic. The computed results are

$$\begin{pmatrix} m_1^{(1)} \\ m_2^{(1)} \\ m_3^{(1)} \end{pmatrix}_{\text{exact}} = \begin{pmatrix} 1.122227 \\ 1.122461 \\ 1.123047 \end{pmatrix} \times 10^3, \quad \begin{pmatrix} m_1^{(1)} \\ m_2^{(1)} \\ m_3^{(1)} \end{pmatrix}_{GE} = \begin{pmatrix} 1.122239 \\ 1.122473 \\ 1.123059 \end{pmatrix} \times 10^3,$$

$$\begin{pmatrix} m_1^{(1)} \\ m_2^{(1)} \\ m_3^{(1)} \end{pmatrix}_{GTH} = \begin{pmatrix} 1.122227 \\ 1.122461 \\ 1.123047 \end{pmatrix} \times 10^3;$$

$$\begin{aligned}
\begin{pmatrix} m_1^{(2)} \\ m_2^{(2)} \\ m_3^{(2)} \end{pmatrix}_{exact} &= \begin{pmatrix} 2.518215 \\ 2.518743 \\ 2.520059 \end{pmatrix} \times 10^6, & \begin{pmatrix} m_1^{(2)} \\ m_2^{(2)} \\ m_3^{(2)} \end{pmatrix}_{GE} &= \begin{pmatrix} 2.518271 \\ 2.518799 \\ 2.520115 \end{pmatrix} \times 10^6, \\
\begin{pmatrix} m_1^{(2)} \\ m_2^{(2)} \\ m_3^{(2)} \end{pmatrix}_{GTH} &= \begin{pmatrix} 2.518216 \\ 2.518744 \\ 2.520060 \end{pmatrix} \times 10^6; \\
\begin{pmatrix} m_1^{(10)} \\ m_2^{(10)} \\ m_3^{(10)} \end{pmatrix}_{exact} &= \begin{pmatrix} 1.147324 \\ 1.147565 \\ 1.148164 \end{pmatrix} \times 10^{37}, & \begin{pmatrix} m_1^{(10)} \\ m_2^{(10)} \\ m_3^{(10)} \end{pmatrix}_{GE} &= \begin{pmatrix} 1.147451 \\ 1.147692 \\ 1.148292 \end{pmatrix} \times 10^{37}, \\
\begin{pmatrix} m_1^{(10)} \\ m_2^{(10)} \\ m_3^{(10)} \end{pmatrix}_{GTH} &= \begin{pmatrix} 1.147325 \\ 1.147566 \\ 1.148166 \end{pmatrix} \times 10^{37}.
\end{aligned}$$

We remark that the first moment vector of first passage times from the chosen NCD subset to the remaining states has elements that are close to each other and are of the order 10^3 . See [1, 9] for detailed information regarding this phenomenon. The relative errors in $(m_{GE}^{(1)}, m_{GE}^{(2)}, m_{GE}^{(10)})$ and $(m_{GTH}^{(1)}, m_{GTH}^{(2)}, m_{GTH}^{(10)})$ for $\mathcal{S} = \{1, 2, 3\}$ are, respectively, $(1.1 \times 10^{-5}, 2.2 \times 10^{-5}, 1.1 \times 10^{-4})$ and $(0, 4.0 \times 10^{-7}, 1.7 \times 10^{-6})$ in the infinity norm.

Now, let us consider $\mathcal{S} = \{1, 4, 6\}$ in which each state comes from a different NCD subset. For this choice of safe states, we have $\text{cond}_2(I - P_{\mathcal{S}, \mathcal{S}}) = 2.7 \times 10^0$ from MATLAB. For this problem the first twenty-two moment vectors can be computed in single precision floating-point arithmetic. The computed results are

$$\begin{aligned}
\begin{pmatrix} m_1^{(1)} \\ m_4^{(1)} \\ m_6^{(1)} \end{pmatrix}_{exact} &= \begin{pmatrix} 6.687500 \\ 3.333333 \\ 2.500000 \end{pmatrix} \times 10^0, & \begin{pmatrix} m_1^{(1)} \\ m_4^{(1)} \\ m_6^{(1)} \end{pmatrix}_{GE} &= \begin{pmatrix} 6.687501 \\ 3.333333 \\ 2.500000 \end{pmatrix} \times 10^0, \\
\begin{pmatrix} m_1^{(1)} \\ m_4^{(1)} \\ m_6^{(1)} \end{pmatrix}_{GTH} &= \begin{pmatrix} 6.687501 \\ 3.333333 \\ 2.500000 \end{pmatrix} \times 10^0; \\
\begin{pmatrix} m_1^{(2)} \\ m_4^{(2)} \\ m_6^{(2)} \end{pmatrix}_{exact} &= \begin{pmatrix} 8.261667 \\ 1.888889 \\ 1.000000 \end{pmatrix} \times 10^1, & \begin{pmatrix} m_1^{(2)} \\ m_4^{(2)} \\ m_6^{(2)} \end{pmatrix}_{GE} &= \begin{pmatrix} 8.261670 \\ 1.888889 \\ 1.000000 \end{pmatrix} \times 10^1, \\
\begin{pmatrix} m_1^{(2)} \\ m_4^{(2)} \\ m_6^{(2)} \end{pmatrix}_{GTH} &= \begin{pmatrix} 8.261668 \\ 1.888889 \\ 1.000000 \end{pmatrix} \times 10^1; \\
\begin{pmatrix} m_1^{(22)} \\ m_4^{(22)} \\ m_6^{(22)} \end{pmatrix}_{exact} &= \begin{pmatrix} 2.814053 \times 10^{38} \\ 9.561367 \times 10^{30} \\ 3.840642 \times 10^{27} \end{pmatrix}, & \begin{pmatrix} m_1^{(22)} \\ m_4^{(22)} \\ m_6^{(22)} \end{pmatrix}_{GE} &= \begin{pmatrix} 2.814063 \times 10^{38} \\ 9.561357 \times 10^{30} \\ 3.840647 \times 10^{27} \end{pmatrix}, \\
\begin{pmatrix} m_1^{(22)} \\ m_4^{(22)} \\ m_6^{(22)} \end{pmatrix}_{GTH} &= \begin{pmatrix} 2.814057 \times 10^{38} \\ 9.561385 \times 10^{30} \\ 3.840641 \times 10^{27} \end{pmatrix}.
\end{aligned}$$

In this case, the relative errors in $(m_{GE}^{(1)}, m_{GE}^{(2)}, m_{GE}^{(22)})$ and $(m_{GTH}^{(1)}, m_{GTH}^{(2)}, m_{GTH}^{(22)})$ are, respectively,

$(1.5 \times 10^{-7}, 3.6 \times 10^{-7}, 3.6 \times 10^{-6})$ and $(1.5 \times 10^{-7}, 1.2 \times 10^{-7}, 1.4 \times 10^{-6})$ in the infinity norm. It is not surprising to encounter higher relative errors when computing the larger means and variances of first passage times corresponding to the NCD subset $\mathcal{S} = \{1, 2, 3\}$.

5.2. Test problem 2. The second problem we consider is inspired by the class of 10×10 matrices used in [20]. Let

$$S = \begin{pmatrix} 0.1 & 0.3 & 0.1 & 0.2 & 0.3 & \beta & 0 & 0 & 0 & 0 \\ 0.2 & 0.1 & 0.1 & 0.2 & 0.4 & 0 & 0 & 0 & 0 & 0 \\ 0.1 & 0.2 & 0.2 & 0.4 & 0.1 & 0 & 0 & 0 & 0 & 0 \\ 0.4 & 0.2 & 0.1 & 0.2 & 0.1 & 0 & 0 & 0 & 0 & 0 \\ 0.6 & 0.3 & 0 & 0 & 0.1 & 0 & 0 & 0 & 0 & 0 \\ \beta & 0 & 0 & 0 & 0 & 0.1 & 0.2 & 0.2 & 0.4 & 0.1 \\ 0 & 0 & 0 & 0 & 0 & 0.2 & 0.2 & 0.1 & 0.3 & 0.2 \\ 0 & 0 & 0 & 0 & 0 & 0.1 & 0.3 & 0.2 & 0.2 & 0.2 \\ 0 & 0 & 0 & 0 & 0 & 0.2 & 0.2 & 0.1 & 0.3 & 0.2 \\ 0 & 0 & 0 & 0 & 0 & 0.1 & 0.7 & 0 & 0 & 0.2 \end{pmatrix}.$$

If $D = \text{diag}([1/(1+\beta), 1, 1, 1, 1, 1/(1+\beta), 1, 1, 1, 1]^T)$, then $P = DS$ is an NCD MC of order 10 with degree of coupling $\beta/(1+\beta)$. Note that P has two subsets of NCD states, $\{1, 2, 3, 4, 5\}$ and $\{6, 7, 8, 9, 10\}$, that communicate with each other over a single state in each direction.

We let $\mathcal{S} = \{1, 2, 3, 4, 5\}$. For this choice of safe states and for $\beta = 10^{-7}$, we have $\text{cond}_2(I - P_{\mathcal{S}, \mathcal{S}}) = 5.3 \times 10^7$ from MATLAB. For this problem only the first four moment vectors can be computed in single precision floating-point arithmetic. The computed results are

$$\begin{aligned} \begin{pmatrix} m_1^{(1)} \\ m_2^{(1)} \\ m_3^{(1)} \\ m_4^{(1)} \\ m_5^{(1)} \end{pmatrix}_{\text{exact}} &= \begin{pmatrix} 3.478633 \\ 3.478633 \\ 3.478633 \\ 3.478633 \\ 3.478633 \end{pmatrix} \times 10^7, & \begin{pmatrix} m_1^{(1)} \\ m_2^{(1)} \\ m_3^{(1)} \\ m_4^{(1)} \\ m_5^{(1)} \end{pmatrix}_{GE} &= \begin{pmatrix} 1.484419 \\ 1.484419 \\ 1.484419 \\ 1.484419 \\ 1.484419 \end{pmatrix} \times 10^8, \\ & \begin{pmatrix} m_1^{(1)} \\ m_2^{(1)} \\ m_3^{(1)} \\ m_4^{(1)} \\ m_5^{(1)} \end{pmatrix}_{GTH} &= \begin{pmatrix} 3.478632 \\ 3.478633 \\ 3.478633 \\ 3.478633 \\ 3.478633 \end{pmatrix} \times 10^7; \\ \begin{pmatrix} m_1^{(2)} \\ m_2^{(2)} \\ m_3^{(2)} \\ m_4^{(2)} \\ m_5^{(2)} \end{pmatrix}_{\text{exact}} &= \begin{pmatrix} 2.420177 \\ 2.420177 \\ 2.420177 \\ 2.420177 \\ 2.420177 \end{pmatrix} \times 10^{15}, & \begin{pmatrix} m_1^{(2)} \\ m_2^{(2)} \\ m_3^{(2)} \\ m_4^{(2)} \\ m_5^{(2)} \end{pmatrix}_{GE} &= \begin{pmatrix} 4.406998 \\ 4.406999 \\ 4.406998 \\ 4.406998 \\ 4.406999 \end{pmatrix} \times 10^{16}, \\ & \begin{pmatrix} m_1^{(2)} \\ m_2^{(2)} \\ m_3^{(2)} \\ m_4^{(2)} \\ m_5^{(2)} \end{pmatrix}_{GTH} &= \begin{pmatrix} 2.420177 \\ 2.420177 \\ 2.420177 \\ 2.420177 \\ 2.420177 \end{pmatrix} \times 10^{15}; \end{aligned}$$

$$\begin{aligned}
\begin{pmatrix} m_1^{(4)} \\ m_2^{(4)} \\ m_3^{(4)} \\ m_4^{(4)} \\ m_5^{(4)} \end{pmatrix}_{exact} &= \begin{pmatrix} 3.514354 \\ 3.514355 \\ 3.514355 \\ 3.514354 \\ 3.514354 \end{pmatrix} \times 10^{31}, & \begin{pmatrix} m_1^{(4)} \\ m_2^{(4)} \\ m_3^{(4)} \\ m_4^{(4)} \\ m_5^{(4)} \end{pmatrix}_{GE} &= \begin{pmatrix} 1.165298 \\ 1.165298 \\ 1.165298 \\ 1.165298 \\ 1.165298 \end{pmatrix} \times 10^{34}, \\
\begin{pmatrix} m_1^{(4)} \\ m_2^{(4)} \\ m_3^{(4)} \\ m_4^{(4)} \\ m_5^{(4)} \end{pmatrix}_{GTH} &= \begin{pmatrix} 3.514354 \\ 3.514354 \\ 3.514354 \\ 3.514354 \\ 3.514354 \end{pmatrix} \times 10^{31}.
\end{aligned}$$

We remark that the first moment vector of first passage times from the chosen NCD subset to the remaining states has elements that are identical to each other and are of order 10^7 . A similar argument is valid for the second moment vector of first passage times, though with the squared exponent. This situation can be attributed to the very strongly coupled interaction among the states in the NCD subset and to the very weak single transition to the remaining states. The relative errors in $(m_{GE}^{(1)}, m_{GE}^{(2)}, m_{GE}^{(4)})$ and $(m_{GTH}^{(1)}, m_{GTH}^{(2)}, m_{GTH}^{(4)})$ are, respectively, $(3.3 \times 10^0, 1.7 \times 10^1, 3.3 \times 10^2)$ and $(2.9 \times 10^{-7}, 0, 2.8 \times 10^{-7})$ in the infinity norm. Hence, the proposed method with GE does not have even a single digit of accuracy in any of the moment vectors of first passage times.

5.3. Test problem 3. The third problem appears in [22] and is about a repairable computer system. The system has two processors and three memory units connected to a bus. All of these components can fail independently of each other and are repaired by a single repairman. The system is up if at least one processor, two memory units, and the bus are operational. It is assumed that no failures occur when the system is down. We consider the first example model described in [22, pp. 328–329] having 21 states, 6 of which are those corresponding to operational states. For our purposes, these 6 states are classified as safe states and the rest as unsafe states.

In this case, the underlying problem yields a CTMC whose infinitesimal generator matrix is denoted by Q . This matrix has nonnegative off-diagonal elements and diagonal elements that are negated row sums of its off-diagonal elements; hence, $Qe = 0$. Now, let Y be the random variable associated with the first passage times to subset $\mathcal{U}$ conditioned on an initial probability vector τ in the subset of safe states, $\mathcal{S}$. Assuming the same kind of block partitioning as in (1) and $R = Q_{\mathcal{S}, \mathcal{S}}$, the moments of Y (cf. (5)) are given by

$$(15) \quad E[Y^k] = k! \tau (-R^{-1})^k e$$

for $k \geq 1$ [15, p. 45].

When $k = 1$, (15) becomes

$$E[Y] = -\tau R^{-1} e.$$

Then the first moment vector of first passage times from states in $\mathcal{S}$ to $\mathcal{U}$ can be obtained from

$$(16) \quad -Rm^{(1)} = e.$$

We remark that (16) becomes identical to (8) when one replaces $-R$ with $(I - T)$. The coefficient matrix, $-R$, is again a nonsingular M-matrix and $-R^{-1} \geq 0$. Hence, $m^{(1)} = -R^{-1}e$, simply the row sums of the nonnegative matrix $-R^{-1}$.

When $k = 2$, (15) becomes

$$E[Y^2] = 2\tau R^{-2}e.$$

Then the second moment vector of first passage times from states in $\mathcal{S}$ to $\mathcal{U}$ can be obtained from

$$(17) \quad -Rm^{(2)} = 2m^{(1)}.$$

Note that (17) has a different right-hand side compared to that of (10).

As for DTMCs, it is possible to compute higher moments of first passage times from states in $\mathcal{S}$ to $\mathcal{U}$ using a recurrence. However, the recurrence is much simpler in this case and follows from (15) as

$$(18) \quad -Rm^{(i+1)} = (i+1)m^{(i)} \quad \text{for } i \geq 0 \quad \text{with } m^{(0)} = e.$$

In conclusion, one needs to factorize $-Q_{\mathcal{S},\mathcal{S}}$ once, compute $m^{(1)}$, and then compute $m^{(2)}$ using $m^{(1)}$ and the factors of $-Q_{\mathcal{S},\mathcal{S}}$. The factorization can be performed using GTH as in section 4, but this time on the coefficient matrix

$$A = \begin{pmatrix} -Q_{\mathcal{S},\mathcal{S}} & -Q_{\mathcal{S},\mathcal{U}}e \\ w^T & -w^Te \end{pmatrix}$$

in (13).

An algorithm similar to Algorithm 1 can be given for CTMCs. The matrix S should be replaced with R , and R should be LU factorized using GTH. There should be a single for-loop consisting of a single statement in which the linear system with the right-hand side, $(i+1)$ times the i th moment vector, in (18) is solved for the $(i+1)$ st moment vector through forward and backward substitutions using the L and U factors of R . The storage requirement of the algorithm for CTMCs is smaller than its counterpart for DTMCs since there is no need for a real vector of length $(k+1)$ to store coefficients and there is no need for k real vectors of length $n_{\mathcal{S}}$ to store moment vectors. Other than storage space for the L and U factors, it suffices to have two real vectors of length $n_{\mathcal{S}}$, one for the moment vector to be computed and one for the right-hand side. The time requirement is also smaller since the computation of the right-hand side takes altogether only $(n_{\mathcal{S}} + 1)$ flops. For a total of $(k-1)$ passes over the for-loop, the computation of the right-hand side and the forward and backward substitutions amount to $(k-1)[2(n_{\mathcal{S}} + 1) + k/2]$ flops.

In the reliability problem, $\mathcal{S} = \{1, 2, 3, 4, 6, 7\}$ and we have $\text{cond}_2(-Q_{\mathcal{S},\mathcal{S}}) = 2.7 \times 10^3$ from MATLAB. For this problem the first ten moment vectors can be computed in single precision floating-point arithmetic. The computed results are

$$\begin{pmatrix} m_1^{(1)} \\ m_2^{(1)} \\ m_3^{(1)} \\ m_4^{(1)} \\ m_6^{(1)} \\ m_7^{(1)} \end{pmatrix}_{\text{exact}} = \begin{pmatrix} 9.031009 \\ 8.940884 \\ 8.448356 \\ 8.751327 \\ 8.664630 \\ 7.766966 \end{pmatrix} \times 10^2, \quad \begin{pmatrix} m_1^{(1)} \\ m_2^{(1)} \\ m_3^{(1)} \\ m_4^{(1)} \\ m_6^{(1)} \\ m_7^{(1)} \end{pmatrix}_{GE} = \begin{pmatrix} 9.031031 \\ 8.940908 \\ 8.448377 \\ 8.751351 \\ 8.664653 \\ 7.766987 \end{pmatrix} \times 10^2,$$

$$\begin{pmatrix} m_1^{(1)} \\ m_2^{(1)} \\ m_3^{(1)} \\ m_4^{(1)} \\ m_6^{(1)} \\ m_7^{(1)} \end{pmatrix}_{GTH} = \begin{pmatrix} 9.031010 \\ 8.940884 \\ 8.448356 \\ 8.751328 \\ 8.664630 \\ 7.766965 \end{pmatrix} \times 10^2;$$

$$\begin{aligned}
\begin{pmatrix} m_1^{(2)} \\ m_2^{(2)} \\ m_3^{(2)} \\ m_4^{(2)} \\ m_6^{(2)} \\ m_7^{(2)} \end{pmatrix}_{exact} &= \begin{pmatrix} 1.627131 \\ 1.610873 \\ 1.521222 \\ 1.576529 \\ 1.560893 \\ 1.397454 \end{pmatrix} \times 10^6, & \begin{pmatrix} m_1^{(2)} \\ m_2^{(2)} \\ m_3^{(2)} \\ m_4^{(2)} \\ m_6^{(2)} \\ m_7^{(2)} \end{pmatrix}_{GE} &= \begin{pmatrix} 1.627139 \\ 1.610882 \\ 1.521230 \\ 1.576538 \\ 1.560901 \\ 1.397461 \end{pmatrix} \times 10^6, \\
\begin{pmatrix} m_1^{(2)} \\ m_2^{(2)} \\ m_3^{(2)} \\ m_4^{(2)} \\ m_6^{(2)} \\ m_7^{(2)} \end{pmatrix}_{GTH} &= \begin{pmatrix} 1.627131 \\ 1.610873 \\ 1.521222 \\ 1.576530 \\ 1.560893 \\ 1.397454 \end{pmatrix} \times 10^6; \\
\begin{pmatrix} m_1^{(10)} \\ m_2^{(10)} \\ m_3^{(10)} \\ m_4^{(10)} \\ m_6^{(10)} \\ m_7^{(10)} \end{pmatrix}_{exact} &= \begin{pmatrix} 1.280432 \\ 1.267638 \\ 1.197083 \\ 1.240611 \\ 1.228306 \\ 1.099680 \end{pmatrix} \times 10^{36}, & \begin{pmatrix} m_1^{(10)} \\ m_2^{(10)} \\ m_3^{(10)} \\ m_4^{(10)} \\ m_6^{(10)} \\ m_7^{(10)} \end{pmatrix}_{GE} &= \begin{pmatrix} 1.280466 \\ 1.267672 \\ 1.197115 \\ 1.240645 \\ 1.228339 \\ 1.099710 \end{pmatrix} \times 10^{36}, \\
\begin{pmatrix} m_1^{(10)} \\ m_2^{(10)} \\ m_3^{(10)} \\ m_4^{(10)} \\ m_6^{(10)} \\ m_7^{(10)} \end{pmatrix}_{GTH} &= \begin{pmatrix} 1.280433 \\ 1.267639 \\ 1.197084 \\ 1.240612 \\ 1.228307 \\ 1.099681 \end{pmatrix} \times 10^{36}.
\end{aligned}$$

The relative errors in $(m_{GE}^{(1)}, m_{GE}^{(2)}, m_{GE}^{(10)})$ and $(m_{GTH}^{(1)}, m_{GTH}^{(2)}, m_{GTH}^{(10)})$ are, respectively, $(2.7 \times 10^{-6}, 5.5 \times 10^{-6}, 2.7 \times 10^{-5})$ and $(1.1 \times 10^{-7}, 6.1 \times 10^{-7}, 7.8 \times 10^{-7})$ in the infinity norm. In this problem, the unit of time is hour, which indicates that on the average failure does not happen before a month (see $m^{(1)}$), and thus the system is considered reliable.

6. Conclusion. We have presented a relatively efficient and accurate method to compute the moments of first passage times to a subset of states in finite ergodic MCs. The method amounts to solving linear systems with a common coefficient matrix but different right-hand sides with the latter governed by a novel recurrence introduced in this paper. This proposed recurrence involves the Pascal's triangle and is shown to lead to a stable implementation. The efficiency and accuracy of the proposed method rest respectively on the smaller linear systems solved and the GTH implementation used. Due to the form of the linear systems involved, the method can be implemented in row-wise sparse format with delayed updates. The proposed method is expected to be useful in large MCs that have a relatively small number of safe states and pose computational problems.

Appendix. We prove (12) after we present some basic results about Stirling numbers, which are closely associated with binomial coefficients. We follow the discussion and notation in [5]. Further information concerning Stirling numbers may be obtained from [13].

LEMMA 1. *The k th factorial moment vector, $\gamma^{(k)}$, is a linear combination of the first k moment vectors $m^{(i)}$, $i \in \{1, 2, \dots, k\}$, and is in the form*

$$(19) \quad \gamma^{(k)} = \sum_{i=0}^k (-1)^{k-i} \begin{bmatrix} k \\ i \end{bmatrix} m^{(i)} \quad \text{for } k \geq 0 \quad \text{with } m^{(0)} = e,$$

where $\begin{bmatrix} k \\ i \end{bmatrix}$ denotes the Stirling number of the first kind that is read “ k cycle i .”

Proof. This result follows from the left-hand side of (5), our definition of $m^{(i)}$ as the i th moment vector, the definition of Stirling numbers of the first kind [5, p. 245], and the definition involving Stirling numbers of the first kind with alternating signs [5, p. 249]. $\square$

Remark 1. Stirling numbers of the first kind satisfy the recurrence [5, p. 250]

$$\begin{bmatrix} k \\ i \end{bmatrix} = (k-1) \begin{bmatrix} k-1 \\ i \end{bmatrix} + \begin{bmatrix} k-1 \\ i-1 \end{bmatrix} \quad \text{for } k > 0$$

with

$$\begin{bmatrix} k \\ i \end{bmatrix} = 0 \quad \text{for } k < i, \quad \begin{bmatrix} k \\ 0 \end{bmatrix} = 0 \quad \text{for } k > 0, \quad \text{and} \quad \begin{bmatrix} k \\ k \end{bmatrix} = 1.$$

The next result is intimately related to Lemma 1 and is used in the proof of (12).

LEMMA 2. *The k th moment vector, $m^{(k)}$, is a linear combination of the first k factorial moment vectors $\gamma^{(i)}$, $i \in \{1, 2, \dots, k\}$, and is in the form*

$$(20) \quad m^{(k)} = \sum_{i=0}^k \left\{ \begin{matrix} k \\ i \end{matrix} \right\} \gamma^{(i)} \quad \text{for } k \geq 0 \quad \text{with } \gamma^{(0)} = e,$$

where $\left\{ \begin{matrix} k \\ i \end{matrix} \right\}$ denotes the Stirling number of the second kind that is read “ k subset i .”

Proof. This result follows from Lemma 1, the definition of Stirling numbers of the second kind [5, p. 244], and (6.10) in [5, p. 248]. $\square$

Remark 2. Stirling numbers of the second kind satisfy the recurrence [5, p. 250]

$$\left\{ \begin{matrix} k \\ i \end{matrix} \right\} = i \left\{ \begin{matrix} k-1 \\ i \end{matrix} \right\} + \left\{ \begin{matrix} k-1 \\ i-1 \end{matrix} \right\} \quad \text{for } k > 0$$

with

$$\left\{ \begin{matrix} k \\ i \end{matrix} \right\} = 0 \quad \text{for } k < i, \quad \left\{ \begin{matrix} k \\ 0 \end{matrix} \right\} = 0 \quad \text{for } k > 0, \quad \text{and} \quad \left\{ \begin{matrix} k \\ k \end{matrix} \right\} = 1.$$

The following is a fundamental result which is also used in the proof of (12).

LEMMA 3. *The factorial moment vectors can be computed from the recurrence*

$$(21) \quad S\gamma^{(k+1)} = \sum_{i=0}^k (-1)^{k-i} \frac{(k+1)!}{i!} \gamma^{(i)} \quad \text{for } k \geq 0 \quad \text{with } \gamma^{(0)} = e.$$

Proof. We recall that S is a nonsingular M-matrix and $S^{-1} \geq 0$. Furthermore, $S^{-1}T = TS^{-1}$, where $T = I - S$. The basis $S\gamma^{(1)} = e$ is easy to derive from (6). Using the fact that S^{-1} and T commute, after some algebra on (6), we obtain

$$(22) \quad \gamma^{(k+1)} = (k+1)S^{-1}T\gamma^{(k)} \quad \text{for } k > 0.$$

The recurrence between two consecutive factorial moment vectors in (22) yields

$$(23) \quad \gamma^{(k+1)} = (k+1)!(S^{-1}T)^k \gamma^{(1)} \quad \text{for } k \geq 0.$$

Premultiplying both sides of (6) by T and using the fact that S^{-1} and T commute, we obtain

$$(24) \quad T\gamma^{(k)} = k!(S^{-1}T)^{k-1}S^{-1}Te.$$

But

$$(25) \quad S^{-1}Te = \gamma^{(1)} - e$$

since $T = I - S$ and $S\gamma^{(1)} = e$. Hence, (24) becomes

$$\frac{T\gamma^{(k)}}{k!} = (S^{-1}T)^{k-1}\gamma^{(1)} - (S^{-1}T)^{k-1}e.$$

Using (23), this can be written as

$$(26) \quad \frac{T\gamma^{(k)}}{k!} = \frac{\gamma^{(k)}}{k!} - (S^{-1}T)^{k-1}e.$$

Factoring $(S^{-1}T)^{k-1}$ as $(S^{-1}T)^{k-2}(S^{-1}T)$ in (26), using (25), and continuing in this manner, we obtain

$$(27) \quad T\gamma^{(k)} = \sum_{i=0}^k (-1)^{k-i} \frac{k!}{i!} \gamma^{(i)} \quad \text{for } k > 0 \quad \text{with } \gamma^{(0)} = e.$$

Now, rewriting (22) as

$$S\gamma^{(k+1)} = (k+1)T\gamma^{(k)}$$

and using the result in (27) on its right-hand side, we have (21). $\square$

COROLLARY 1. *The $(k+1)$ st moment vector can be computed from the linear system obtained by premultiplying both sides of (20) by S and then using the recurrence in (21).*

Now we state and prove the theorem.

THEOREM 1. *Let P be the transition probability matrix of an ergodic DTMC, $\mathcal{S}$ be a subset of states in P , and $m^{(k+1)}$ denote the $(k+1)$ st moment vector of first passage times from states in $\mathcal{S}$ to states outside $\mathcal{S}$ for $k \geq 0$. Then $m^{(k+1)}$ can be computed from the recurrence*

$$(28) \quad Sm^{(k+1)} = \sum_{i=0}^k (-1)^{k-i} \binom{k+1}{i} m^{(i)} \quad \text{for } k \geq 0 \quad \text{with } m^{(0)} = e,$$

where $P_{\mathcal{S},\mathcal{S}}$ is the submatrix of P corresponding to the states in $\mathcal{S}$ and $S = I - P_{\mathcal{S},\mathcal{S}}$.

Proof. Our objective is to express S times the $(k+1)$ st moment vector in terms of the first k moment vectors as in (28) (i.e., (12) in the text). To that end, following Corollary 1, we write

$$(29) \quad Sm^{(k+1)} = \sum_{i=0}^k \left\{ \begin{matrix} k+1 \\ i+1 \end{matrix} \right\} S\gamma^{(i+1)} \quad \text{for } k \geq 0.$$

After substituting the result in (19) in the right-hand side of (21) and then substituting the obtained result in the right-hand side of (29), we get

$$Sm^{(k+1)} = \sum_{i=0}^k \sum_{j=0}^i \sum_{l=0}^j (-1)^{i-l} \frac{(i+1)!}{j!} \left\{ \begin{matrix} k+1 \\ i+1 \end{matrix} \right\} \left[\begin{matrix} j \\ l \end{matrix} \right] m^{(l)} \quad \text{for } k \geq 0 \quad \text{with } m^{(0)} = e. \quad (30)$$

We prove the theorem by showing that (30) reduces to (28); in other words,

$$\sum_{i=0}^k \sum_{j=0}^i \sum_{l=0}^j (-1)^{i-l} \frac{(i+1)!}{j!} \left\{ \begin{matrix} k+1 \\ i+1 \end{matrix} \right\} \left[\begin{matrix} j \\ l \end{matrix} \right] m^{(l)} = \sum_{i=0}^k (-1)^{k-i} \binom{k+1}{i} m^{(i)} \quad \text{for } k \geq 0. \quad (31)$$

Let $Y(k)$ refer to the left-hand side of (31). Note that therein it is possible to replace the upper limit of the third summation with k since $j < k$ and $\left[\begin{smallmatrix} j \\ l \end{smallmatrix} \right] = 0$ for $l > j$. Hence, we can write

$$\begin{aligned} Y(k) &= \sum_{i=0}^k \sum_{j=0}^i \sum_{l=0}^k (-1)^{i-l} \frac{(i+1)!}{j!} \left\{ \begin{matrix} k+1 \\ i+1 \end{matrix} \right\} \left[\begin{matrix} j \\ l \end{matrix} \right] m^{(l)} \\ &= \sum_{l=0}^k (-1)^{k-l} m^{(l)} \sum_{i=0}^k (-1)^{i-k} \left\{ \begin{matrix} k+1 \\ i+1 \end{matrix} \right\} (i+1)! \sum_{j=0}^i \left[\begin{matrix} j \\ l \end{matrix} \right] / j!. \end{aligned}$$

But

$$i! \sum_{j=0}^i \left[\begin{matrix} j \\ l \end{matrix} \right] / j! = \left[\begin{matrix} i+1 \\ l+1 \end{matrix} \right]$$

from identity (6.21) on page 251 in [5]. Therefore, we have

$$Y(k) = \sum_{l=0}^k (-1)^{k-l} m^{(l)} \sum_{i=0}^k (-1)^{i-k} \left\{ \begin{matrix} k+1 \\ i+1 \end{matrix} \right\} (i+1) \left[\begin{matrix} i+1 \\ l+1 \end{matrix} \right].$$

After replacing the lower limit of the second summation with l since $\left[\begin{smallmatrix} i+1 \\ l+1 \end{smallmatrix} \right] = 0$ for $i < l$ and then exchanging the variables i and l , we obtain

$$Y(k) = \sum_{i=0}^k (-1)^{k-i} m^{(i)} \sum_{l=i}^k (-1)^{l-k} (l+1) \left\{ \begin{matrix} k+1 \\ l+1 \end{matrix} \right\} \left[\begin{matrix} l+1 \\ i+1 \end{matrix} \right].$$

Hence, the only thing that remains is to show that

$$\sum_{l=i}^k (-1)^{l-k} (l+1) \left\{ \begin{matrix} k+1 \\ l+1 \end{matrix} \right\} \left[\begin{matrix} l+1 \\ i+1 \end{matrix} \right] = \binom{k+1}{i}.$$

But this follows from identity (12) on page 188 in [13]. $\square$

Acknowledgment. We thank the anonymous referees for their constructive reports, which led to an improved manuscript.

REFERENCES

- [1] K. E. AVRACHENKOV AND M. HAVIV, *The first Laurent series coefficients for singularly perturbed stochastic matrices*, Linear Algebra Appl., 386 (2004), pp. 243–259.
- [2] A. BERMAN AND R. J. PLEMMONS, *Nonnegative Matrices in the Mathematical Sciences*, SIAM, Philadelphia, 1994.
- [3] P.-J. COURTOIS, *Decomposability: Queueing and Computer System Applications*, Academic Press, New York, 1977.
- [4] T. DAYAR AND W. J. STEWART, *On the effects of using the Grassmann–Taksar–Heyman method in iterative aggregation–disaggregation*, SIAM J. Sci. Comput., 17 (1996), pp. 287–303.
- [5] R. L. GRAHAM, D. E. KNUTH, AND O. PATASHNIK, *Concrete Mathematics: A Foundation for Computer Science*, Addison-Wesley, Reading, MA, 1989.
- [6] W. K. GRASSMANN, M. I. TAKSAR, AND D. P. HEYMAN, *Regenerative analysis and steady state distributions for Markov chains*, Oper. Res., 33 (1985), pp. 1107–1116.
- [7] P. G. HARRISON AND W. J. KNOTTENBELT, *Passage time distribution in large Markov chains*, Performance Evaluation Review, 30 (2002), pp. 77–85.
- [8] W. J. HARROD AND R. J. PLEMMONS, *Comparison of some direct methods for computing the stationary distributions of Markov chains*, SIAM J. Sci. Statist. Comput., 5 (1984), pp. 453–469.
- [9] R. HASSIN AND M. HAVIV, *Mean passage times and nearly uncoupled Markov chains*, SIAM J. Discrete Math., 5 (1992), pp. 386–397.
- [10] D. P. HEYMAN, *Accurate computation of the fundamental matrix of a Markov chain*, SIAM J. Matrix Anal. Appl., 16 (1995), pp. 954–963.
- [11] D. P. HEYMAN AND D. P. O’LEARY, *What is fundamental for Markov chains: First passage times, fundamental matrices, and group generalized inverses*, in Computations with Markov Chains, W. J. Stewart, ed., Kluwer, Boston, MA, 1995, pp. 151–161.
- [12] D. P. HEYMAN AND D. P. O’LEARY, *Overcoming instability in computing the fundamental matrix for a Markov chain*, SIAM J. Matrix Anal. Appl., 19 (1998), pp. 534–540.
- [13] C. JORDAN, *Calculus of Finite Differences*, 3rd ed., Chelsea, New York, 1979.
- [14] J. G. KEMENY AND J. L. SNELL, *Finite Markov Chains*, Springer-Verlag, New York, 1983.
- [15] G. LATOUCHE AND V. RAMASWAMI, *Introduction to Matrix Analytic Methods in Stochastic Modeling*, SIAM, Philadelphia, 1999.
- [16] I. MAREK, *Iterative aggregation/disaggregation methods for computing some characteristics of Markov chains. II. Fast convergence*, Appl. Numer. Math., 45 (2003), pp. 11–28.
- [17] C. D. MEYER, JR., *The role of the group generalized inverse in the theory of finite Markov chains*, SIAM Rev., 17 (1975), pp. 443–464.
- [18] C. D. MEYER, *Stochastic complementation, uncoupling Markov chains, and the theory of nearly reducible systems*, SIAM Rev., 31 (1989), pp. 240–272.
- [19] C. A. O’CINNEIDE, *Entrywise perturbation theory and error analysis for Markov chains*, Numer. Math., 65 (1993), pp. 109–120.
- [20] C. C. PAIGE, P. H. STYAN, AND P. G. WACHTER, *Computation of the stationary distribution of a Markov chain*, J. Stat. Comput. Simul., 4 (1975), pp. 1156–1179.
- [21] *Software for computing moments of first passage times to a subset of states in Markov chains*, <http://www.cs.bilkent.edu.tr/~tugrul/software.html>.
- [22] E. DE SOUZA E SILVA AND H. R. GAIL, *Calculating cumulative operational time distributions of repairable computer systems*, IEEE Trans. Comput., C-35 (1986), pp. 322–332.
- [23] W. J. STEWART, *Introduction to the Numerical Solution of Markov Chains*, Princeton University Press, Princeton, NJ, 1994.

DRAZIN INVERSE AND PERIODIC COLLECTION OF MATRICES*

BEGOÑA CANTÓ[†], CARMEN COLL[†], AND ELENA SÁNCHEZ[†]

Abstract. This paper is concerned with weakly N -cyclic matrices and their Drazin inverses. In this study an N -periodic collection of monodromy matrices is used. It is shown that the Drazin inverses of these matrices at consecutive times are related. An explicit expression for the Drazin inverse of a weakly N -cyclic matrix is given. Finally, the Drazin inverse of a weakly N -cyclic matrix is used to obtain the explicit expression for the solution of an N -periodic descriptor system of difference equations in terms of N -periodic collections of matrices which are the coefficients of the N -periodic system.

Key words. Drazin inverse, monodromy matrix, weakly N -cyclic matrix, difference equations

AMS subject classifications. 15A09, 39A05

DOI. 10.1137/S0895479803435968

1. Introduction and background. In recent years, the study of generalized inverses has been of leading interest for many researchers. This is because such inverses are useful tools for several applications. Until now these inverses have been applied to problems involved in the theory of linear systems of algebraic equations (see [1]), in the theory of Markov chains (see [13]), and in several economic problems (see [16]) or electrical problems (see [4]), which are modelled by descriptor systems of differential equations or of difference equations.

The solution of a discrete-time descriptor system may be written in terms of Drazin inverses of the coefficient matrices of the system (see [11]). This fact implies that the theory of Drazin inverses is becoming an active area of research for addressing certain control problems. Topics related to Drazin inverses have been studied by several authors. For example, some additive perturbation results for Drazin inverses have been studied in [10], the conditions to characterize the convergence of the Neumann-type series for the Drazin inverse have been given in [6], necessary and sufficient conditions for the reverse order law for the Drazin inverse of a product of matrices have been obtained in [20], and a characterization and representation of the Drazin inverse has been presented in [22]. The perturbation theory for the Drazin inverse and the computational aspects of the Drazin inverse can be found in [21]. Nonnegative inverses are studied in [2] and [14].

The study of N -periodic descriptor systems leads to the use of N -periodic collections of matrices. The achievement of an analytic expression of its solution is important because it facilitates the study of many properties of the periodic descriptor system. This fact allows the extension of the results in a theoretical way from the time-invariant case to the time-periodic case. For the N -periodic system, an explicit expression for the solution is difficult to obtain because it involves the Drazin inverse of a weakly N -cyclic matrix constructed from the N -periodic collection of matrices. The knowledge of properties of the Drazin inverse of products of matrices is useful in developing this study.

*Received by the editors October 8, 2003; accepted for publication (in revised form) by V. Mehrmann December 15, 2004; published electronically November 14, 2005. This research was partially supported by Spanish grant BFM2001-2783.

<http://www.siam.org/journals/simax/27-2/43596.html>

[†]Departament de Matemàtica Aplicada, Institut de Matemàtica Multidisciplinar, Universitat Politècnica de València, 46071, Valencia, Spain (esanchezj@mat.upv.es).

This paper is organized as follows. In section 2, an N -periodic collection of matrices is considered and results related to the Drazin inverses of the monodromy matrices (2.2) are given. Next, in section 3, the Drazin inverse of a weakly N -cyclic matrix is obtained. Finally, in section 4, the Drazin inverse of this matrix is used to obtain an explicit expression for the solution of an N -periodic descriptor system of difference equations.

2. Preliminaries. For achieving these aims the following notation and definition are needed.

The Drazin inverse of an $n \times n$ matrix M is the matrix M^D satisfying (i) $M^D M M^D = M^D$, (ii) $M M^D = M^D M$, and (iii) $M^{k+1} M^D = M^k$, where $k = \text{ind}(M)$ is the index of M ; that is, it is the smallest nonnegative integer k such that $\text{rank}(M^k) = \text{rank}(M^{k+1})$.

It is well known (see [5]) that if $M = XY$, then

$$(2.1) \quad M^D = X[(YX)^2]^D Y.$$

All the matrices $M(k)$ introduced in the following definition are supposed to be square. To extend some results of this paper to the time-varying dimension case, we need matrices satisfying special properties, for instance, collection of matrices cyclically multiplicable; see [8].

Given a collection of matrices $\{M(k) \in \mathbb{R}^{n \times n}, k \in \mathbb{Z}\}$ depending on the parameter $k \in \mathbb{Z}$, briefly denoted by $M(\cdot)$, we define the matrix

$$\Phi_M(k, k_0) = \begin{cases} M(k-1)M(k-2) \cdots M(k_0), & k > k_0, \\ I_n, & k = k_0. \end{cases}$$

A collection of matrices $\{M(k) \in \mathbb{R}^{n \times n}, k \in \mathbb{Z}\}$ is periodic of period N if $M(k+N) = M(k)$, $k \in \mathbb{Z}$. In this case, $\Phi_M(k+N, k_0+N) = \Phi_M(k, k_0)$, and

$$(2.2) \quad M_k = \Phi_M(k+N, k)$$

is called the *monodromy matrix* at time k of the N -periodic collection of matrices $\{M(k) \in \mathbb{R}^{n \times n}, k \in \mathbb{Z}\}$.

An N -periodic collection $\{M(k) \in \mathbb{R}^{n \times n}, k \in \mathbb{Z}\}$ is *contragrediently equivalent* to an N -periodic collection $\{N(k) \in \mathbb{R}^{n \times n}, k \in \mathbb{Z}\}$ if there exists an N -periodic collection of nonsingular matrices $\{T(k) \in \mathbb{R}^{n \times n}, k \in \mathbb{Z}\}$ such that $N(k) = T(k)M(k)T(k-1)^{-1}$, $k \in \mathbb{Z}$; see [8, 17].

The expressions $\text{diag}[M(i)]_{i=0}^{N-1}$ and $\text{col}[v(i)]_{i=0}^{N-1}$ denote a diagonal matrix and a column vector given by

$$\begin{bmatrix} M(0) & \cdots & 0 \\ \vdots & \ddots & \vdots \\ 0 & \cdots & M(N-1) \end{bmatrix} \quad \text{and} \quad \begin{bmatrix} v(0) \\ v(1) \\ \vdots \\ v(N-1) \end{bmatrix},$$

respectively.

An $n \times n$ matrix M is called weakly N -cyclic, $N > 1$, if there exists an $n \times n$ permutation matrix P such that

$$PMP^T = \begin{bmatrix} 0 & M(0) \\ \text{diag}[M(i)]_{i=1}^{N-1} & 0 \end{bmatrix}.$$

We let S_j denote the weakly N -cyclic matrix given by

$$(2.3) \quad S_j = \begin{bmatrix} 0 & I_{j(N-1)} \\ I_j & 0 \end{bmatrix}.$$

3. Drazin inverse of a weakly N -cyclic matrix. Consider an N -periodic collection of matrices $\{A(k) \in \mathbb{R}^{n \times n}, k \in \mathbb{Z}\}$ and its collection of monodromy matrices $\{A_s, s \in \mathbb{Z}\}$. From the results given in [9] we have the following relation among the indices of the monodromy matrices: if $q_s = \text{ind}(A_s)$ and $q_t = \text{ind}(A_t)$, then $|q_t - q_s| \leq 1 \forall s, t \in \mathbb{Z}$.

On the other hand, associated with an N -periodic collection of matrices $\{A(k) \in \mathbb{R}^{n \times n}, k \in \mathbb{Z}\}$ we define a weakly N -cyclic matrix given by

$$(3.1) \quad A_e = \begin{bmatrix} 0 & A(0) \\ \text{diag}[A(i)]_{i=1}^{N-1} & 0 \end{bmatrix}.$$

As A_e is a weakly N -cyclic matrix, we have $A_e^N = \text{diag}[A_i]_{i=0}^{N-1}$. If $\text{ind}(A_e) = Np$, then $\text{rank}(A_i^{p+1}) = \text{rank}(A_i^p)$, and $\text{ind}(A_i) \leq p, i \in \mathbb{Z}$. Thus, we have a relationship between the index of the matrix A_e and the indices of the monodromy matrices: if $\text{ind}(A_e) = Np$ for some $p \in \mathbb{Z}$, then $\text{ind}(A_s) \leq p, s \in \mathbb{Z}$.

In the following result the Drazin inverses of the monodromy matrices A_s at consecutive times are related.

PROPOSITION 3.1. *Consider an N -periodic collection $\{A(k) \in \mathbb{R}^{n \times n}, k \in \mathbb{Z}\}$; then the Drazin inverses of the monodromy matrices A_s at consecutive times satisfy*

- (i) $A_{s+1}^D = A(s)A_s^D A_s^D \Phi_A(s+N, s+1)$,
- (ii) $A_s A_s^D = \Phi_A(s+N, s+1)A_{s+1}^D A(s)$,
- (iii) $\Phi_A(s+N, s+1)A_{s+1}^D = A_s^D \Phi_A(s+N, s+1)$.

Proof. (i) As $A_{s+1} = \phi_A(s+N+1, s+1)$, it follows that

$$A_{s+1}^D = [\phi_A(s+N+1, s+1)]^D = [A(s)\phi_A(s+N, s+1)]^D.$$

Using (2.1) we obtain the Drazin inverse

$$\begin{aligned} A_{s+1}^D &= A(s) \left[[\phi_A(s+N, s+1)A(s)]^2 \right]^D \phi_A(s+N, s+1) \\ &= A(s)A_s^D A_s^D \phi_A(s+N, s+1). \end{aligned}$$

(ii) Taking $A_{s+1} = \phi_A(s+N+1, s+N-1)\phi_A(s+N-1, s+1)$ and using (2.1), we have

$$A_{s+1}^D = \phi_A(s+1, s-1) \left[(\phi_A(s+N-1, s-1))^2 \right]^D \phi_A(s+N-1, s+1).$$

Hence

$$\begin{aligned} &\phi_A(s+N, s+1)A_{s+1}^D A(s) \\ &= \phi_A(s+N, s)A(s-1) \left[[\phi_A(s+N-1, s)A(s-1)]^2 \right]^D \phi_A(s+N-1, s) \\ &= A_s [A(s-1)\phi_A(s+N-1, s)]^D = A_s A_s^D. \end{aligned}$$

(iii) By (i) and the Drazin inverse properties, we have

$$\begin{aligned}\phi_A(s+N, s+1)A_{s+1}^D &= \phi_A(s+N, s+1)A(s)A_s^D A_s^D \phi_A(s+N, s+1) \\ &= A_s A_s^D A_s^D \phi_A(s+N, s+1) \\ &= A_s^D \phi_A(s+N, s+1). \quad \square\end{aligned}$$

PROPOSITION 3.2. *Consider an N -periodic collection $\{A(k) \in \mathbb{R}^{n \times n}, k \in \mathbb{Z}\}$ and the matrix S_n given in (2.3); then*

- (i) $\text{diag}[A(i)]_{i=0}^{N-1} S_n^k = S_n^k \text{diag}[A(N+i-k)]_{i=0}^{N-1}$,
- (ii) $A_e = \text{diag}[A(i)]_{i=0}^{N-1} S_n^{-1} = S_n^{-1} \text{diag}[A(i+1)]_{i=0}^{N-1}$,
- (iii) $A_e^k = \text{diag}[\Phi_A(N+i+1, N+i+1-k)]_{i=0}^{N-1} S_n^{-k} = S_n^{-k} \text{diag}[\Phi_A(N+i+1+k, N+i+1)]_{i=0}^{N-1}, k \in \mathbb{Z}$.

Proof. Statements (i) and (ii) follow by straightforward algebraic manipulation.

(iii) Using (ii) and $k=2$, we have

$$\begin{aligned}A_e^2 &= S_n^{-1} \text{diag}[A(i+1)]_{i=0}^{N-1} \text{diag}[A(i)]_{i=0}^{N-1} S_n^{-1} \\ &= S_n^{-1} \text{diag}[A(i+1)A(i)]_{i=0}^{N-1} S_n^{-1} \\ &= S_n^{-2} \text{diag}[A(i+2)A(i+1)]_{i=0}^{N-1} \\ &= S_n^{-2} \text{diag}[\phi_A(i+3, i+1)]_{i=0}^{N-1}.\end{aligned}$$

By induction

$$\begin{aligned}A_e^k &= A_e^{k-1} A_e \\ &= S_n^{-(k-1)} \text{diag}[A(i+k-1) \cdots A(i+1)]_{i=0}^{N-1} \text{diag}[A(i)]_{i=0}^{N-1} S_n^{-1} \\ &= S_n^{-k} \text{diag}[\phi_A(i+k+1, i+1)]_{i=0}^{N-1}.\end{aligned}$$

Then

$$\begin{aligned}&\text{diag}[\phi_A(N+i+1, N+i+1-k)]_{i=0}^{N-1} S_n^{-k} \\ &= S_n^{-k} \text{diag}[\phi_A(N+i+1+k, N+i+1)]_{i=0}^{N-1}.\end{aligned} \quad \square$$

In the next result we give an expression of the Drazin inverse of a weakly N -cyclic matrix.

THEOREM 3.3. *Consider an N -periodic collection $\{A(k) \in \mathbb{R}^{n \times n}, k \in \mathbb{Z}\}$ and the matrix A_e given in (3.1). Then the Drazin inverse of the matrix A_e is given by*

$$\begin{aligned}(3.2) \quad A_e^D &= \text{diag}[A_{i+1}^D \phi_A(N+i+1, i+2)]_{i=0}^{N-1} S_n \\ &= S_n \text{diag}[A_i^D \phi_A(N+i, i+1)]_{i=0}^{N-1}.\end{aligned}$$

Proof. From [8] and [17] the N -periodic collection $\{A(k) \in \mathbb{R}^{n \times n}, k \in \mathbb{Z}\}$ is contragrediently equivalent to an N -periodic collection $\{\bar{A}(k) \in \mathbb{R}^{n \times n}, k \in \mathbb{Z}\}$ having the form

$$\begin{aligned}\bar{A}(0) &= T(0)A(0)T(N-1)^{-1} = \begin{bmatrix} J & 0 \\ 0 & H(0) \end{bmatrix}, \\ \bar{A}(k) &= T(k)A(k)T(k-1)^{-1} = \begin{bmatrix} I & 0 \\ 0 & H(k) \end{bmatrix}, \quad k=1, \dots, N-1,\end{aligned}$$

and $A(k+N) = A(k)$, $k \in \mathbb{Z}$, where $\{T(k) \in \mathbb{R}^{n \times n}, k \in \mathbb{Z}\}$ is an N -periodic collection of nonsingular matrices. The matrix J denotes the Jordan block of A_k corresponding to the nonzero eigenvalues, and $H(k)$, $k = 0, 1, \dots, N-1$, are block diagonal matrices except for some null rows and/or null columns, where the possible blocks are obtained from the construction made in [8]. Hence,

$$\bar{A}_k = T(k-1)A_kT(k-1)^{-1} = \begin{bmatrix} J & 0 \\ 0 & J_k(0) \end{bmatrix},$$

where $J_k(0) = \Phi_H(N+k, k)$ is the Jordan block of A_k corresponding to the zero eigenvalue. Taking $T = \text{diag}[T(i)]_{i=0}^{N-1}$, we have that the matrix A_e is similar to the matrix $\bar{A}_e$; that is,

$$\bar{A}_e = \text{diag}[\bar{A}(i)]_{i=0}^{N-1} S_n^{-1} = T A_e T^{-1}.$$

From the block diagonal structure of the matrices $\{\bar{A}_k \in \mathbb{R}^{n \times n}, k \in \mathbb{Z}\}$ and using a permutation matrix P , we have

$$\bar{\bar{A}}_e = P \bar{A}_e P^{-1} = \begin{bmatrix} J_e & 0 \\ 0 & H_e \end{bmatrix},$$

where

$$J_e = \begin{bmatrix} 0 & 0 & \cdots & 0 & J \\ I & 0 & \cdots & 0 & 0 \\ \vdots & \vdots & & \vdots & \vdots \\ 0 & 0 & \cdots & I & 0 \end{bmatrix}, \quad H_e = \begin{bmatrix} 0 & 0 & \cdots & 0 & H(0) \\ H(1) & 0 & \cdots & 0 & 0 \\ \vdots & \vdots & & \vdots & \vdots \\ 0 & 0 & \cdots & H(N-1) & 0 \end{bmatrix}.$$

As J_e is a nonsingular matrix and H_e is a nilpotent matrix, we have that the Drazin inverse of the matrix $\bar{\bar{A}}_e$ is given by

$$\bar{\bar{A}}_e^D = P^{-1} \bar{\bar{A}}_e P = P^{-1} \begin{bmatrix} J_e^{-1} & 0 \\ 0 & 0 \end{bmatrix} P.$$

By straightforward algebraic manipulation we obtain

$$\bar{\bar{A}}_e^D = S_n \text{diag}[\bar{\bar{A}}_i^D \phi_{\bar{\bar{A}}}(N+i, i+1)]_{i=0}^{N-1},$$

since

$$\bar{\bar{A}}_i^D \phi_{\bar{\bar{A}}}(N+i, i+1) = \begin{bmatrix} I & 0 \\ 0 & 0 \end{bmatrix}, \quad i \neq 0, \quad \text{and} \quad \bar{\bar{A}}_0^D \phi_{\bar{\bar{A}}}(N, 1) = \begin{bmatrix} J^{-1} & 0 \\ 0 & 0 \end{bmatrix}.$$

Hence,

$$A_e^D = T^{-1} \bar{\bar{A}}_e^D T = S_n \text{diag}[A_i^D \phi_A(N+i, i+1)]_{i=0}^{N-1}. \quad \square$$

Following the construction given in the proof of the theorem it is necessary to compute the Jordan block of the monodromy matrix $\bar{A}_k$ corresponding to the nonzero eigenvalues. Some algorithms to obtain the N -periodic collection $\{\bar{A}(k) \in \mathbb{R}^{n \times n}, k \in \mathbb{Z}\}$ are given in [17, 19].

To clarify the obtained results in Theorem 3.3 we give the following example.

Example 3.1. Consider the 2-periodic collection $\{\bar{A}(k) \in \mathbb{R}^{5 \times 5}, k \in \mathbb{Z}\}$ given by

$$\bar{A}(0) = \begin{bmatrix} 2 & 1 & 0 & 0 & 0 \\ 0 & 2 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 & 0 \end{bmatrix}, \quad \bar{A}(1) = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 \end{bmatrix},$$

and $\bar{A}(k+2) = \bar{A}(k)$, $k \in \mathbb{Z}$. The corresponding 2-periodic collection of monodromy matrices $\{\bar{A}_s \in \mathbb{R}^{5 \times 5}, s \in \mathbb{Z}\}$ is given by

$$\bar{A}_0 = \bar{A}(1)\bar{A}(0) = \begin{bmatrix} 2 & 1 & 0 & 0 & 0 \\ 0 & 2 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 & 0 \end{bmatrix}, \quad \bar{A}_1 = \bar{A}(0)\bar{A}(1) = \begin{bmatrix} 2 & 1 & 0 & 0 & 0 \\ 0 & 2 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \end{bmatrix}.$$

Their Drazin inverses are

$$\bar{A}_0^D = \bar{A}_1^D = \begin{bmatrix} 1/2 & -1/4 & 0 & 0 & 0 \\ 0 & 1/2 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \end{bmatrix}.$$

From Theorem 3.3 it follows that $\bar{A}_e^D = S_n \text{diag} [\bar{A}_i^D \phi_{\bar{A}}(N+i, i+1)]_{i=0}^{N-1}$. In this case,

$$\bar{A}_e^D = \begin{bmatrix} 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 1/2 & -1/4 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1/2 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix}.$$

It is easy to prove that the above matrix satisfies the three properties of the Drazin inverse matrix, that is, $\bar{A}_e^D \bar{A}_e \bar{A}_e^D = \bar{A}_e^D$, $\bar{A}_e \bar{A}_e^D = \bar{A}_e^D \bar{A}_e$, and $\bar{A}_e^{k+1} \bar{A}_e^D = \bar{A}_e^k$, where $k = \text{ind}(\bar{A}_e)$ is the index of $\bar{A}_e$.

Finally, we give a result on the power of the Drazin inverse of a weakly N -cyclic matrix.

PROPOSITION 3.4. *Consider an N -periodic collection $\{A(k) \in \mathbb{R}^{n \times n}, k \in \mathbb{Z}\}$ and the matrix A_e given in (3.1). If the Drazin inverse of A_e is given by expression (3.2), then*

$$(A_e^D)^k = \text{diag}[A_{i+1}^D \Phi_A(N+i+1, i+1+k)]_{i=0}^{N-1} S_n^k, \quad k \in \mathbb{Z},$$

where S_n is the matrix given in (2.3).

Proof. For $k = 1$ using Theorem 3.3 we have $A_e^D = \text{diag}[A_{i+1}^D \Phi_A(N+i+1, i+2)]_{i=0}^{N-1} S_n$.

For $k = 2$ we obtain

$$\begin{aligned} (A_e^D)^2 &= \text{diag} [A_{i+1}^D \phi_A(N+i+1, i+2) A_{i+2}^D \phi_A(N+i+2, i+3)]_{i=0}^{N-1} S_n^2 \\ &= \text{diag} [A_{i+1}^D A_{i+1}^D A_{i+1}^D \phi_A(N+i+1, i+3)]_{i=0}^{N-1} S_n^2 \\ &= \text{diag} [A_{i+1}^D \phi_A(N+i+1, i+3)]_{i=0}^{N-1} S_n^2. \end{aligned}$$

By induction,

$$\begin{aligned} (A_e^D)^k &= (A_e^D)^{k-1} A_e^D \\ &= \text{diag} [A_{i+1}^D \phi_A(N+i+1, i+k)]_{i=0}^{N-1} S_n^{k-1} \\ &\quad \text{diag} [A_{i+1}^D \phi_A(N+i+1, i+2)]_{i=0}^{N-1} S_n \\ &= \text{diag} [A_{i+1}^D \phi_A(N+i+1, i+k) A_{i+k}^D \phi_A(N+i+k, i+k+1)]_{i=0}^{N-1} S_n^k \\ &= \text{diag} [A_{i+1}^D A_{i+1}^D A_{i+1}^D \phi_A(N+i+1, i+k+1)]_{i=0}^{N-1} S_n^k \\ &= \text{diag} [A_{i+1}^D \phi_A(N+i+1, i+k+1)]_{i=0}^{N-1} S_n^k. \quad \square \end{aligned}$$

In the next section we use the above result to obtain the solution of an N -periodic descriptor system of difference equations.

4. Application to the solution of an N -periodic descriptor system of difference equations. Periodic systems play an important role in modelling sampling control processes where the measurement samples and control calculations must be performed at different frequencies.

On the other hand, descriptor systems appear in the study of singularly perturbed systems; see [12]. These systems can have the following structure:

$$(4.1) \quad \begin{aligned} x(k+1) &= A_{11}x(k) + A_{11}z(k) + B_1u(k), \\ \varepsilon z(k+1) &= A_{21}x(k) + A_{22}z(k) + B_2u(k), \end{aligned}$$

where $x(k)$ and $z(k)$ are the n_1 and n_2 dimensional state vectors, respectively, $u(k)$ is an m dimensional control vector, and ε is a small positive parameter. This parameter ε tends to zero.

Control processes modelled by this kind of system can be found in several engineering applications such as electrical circuit networks, aerospace engineering, and chemical processing. In particular, a periodic system for modelling the dynamics of nitrogen absorption, distribution, and translocation in citrus trees can be found in [3]. The presence of some parameters such as small climatic changes or the existence of impurities in the soil generates a singular perturbation in the model. In this case, the singularly perturbed system will be a periodic descriptor system.

Consider an N -periodic descriptor system given by

$$(4.2) \quad E(k)x(k+1) = A(k)x(k) + B(k)u(k), \quad k \in \mathbb{Z},$$

where $E(k) \in \mathbb{R}^{n \times n}$, $k \in \mathbb{Z}$; $A(k) \in \mathbb{R}^{n \times n}$, $k \in \mathbb{Z}$; and $B(k) \in \mathbb{R}^{n \times m}$, $k \in \mathbb{Z}$, are N -periodic matrices, with $\ker(E(k)) \cap \ker(A(k)) = \{0\}$. The system (4.2) is denoted by $(E(\cdot), A(\cdot), B(\cdot))_N$.

A *time-invariant cyclically augmented descriptor system* (ICADS) is associated with the system (4.2) (see [15] for the standard case, where $E(k) = I$, $k \in \mathbb{Z}$, and see [18] for the descriptor case). This ICADS is given by

$$E_e z(k+1) = A_e z(k) + B_e u_e(k), \quad k \in \mathbb{Z},$$

where A_e is the matrix given in (3.1), $E_e = \text{diag}[E(i)]_{i=0}^{N-1}$, $B_e = \text{diag}[B(i)]_{i=0}^{N-1}$, and

$$z(k) = S_n^{-k+1} \hat{x}(k) \quad \text{and} \quad u_e(k) = S_m^{-k} \hat{u}(k),$$

where $\hat{x}(k) = \text{col}[x(k+j)]_{j=0}^{N-1}$, $\hat{u}(k) = \text{col}[u(k+j)]_{j=0}^{N-1}$, and S_j is the matrix given in (2.3).

If the pair (E_e, A_e) is regular (see [7]), that is, if there exists $\lambda \in \mathbb{C}$ such that $\det(\lambda E_e - A_e) \neq 0$ and $E_e A_e = A_e E_e$, then the trajectory of the ICADS is given by

$$\begin{aligned} z(k) &= (E_e^D A_e)^k E_e^D E_e z(0) \\ &\quad + \sum_{i=0}^{k-1} E_e^D (E_e^D A_e)^{k-i-1} B_e u_e(i) \\ &\quad - (I - E_e^D E_e) \sum_{i=0}^{q-1} (E_e A_e^D)^i A_e^D B_e u_e(k+i), \quad k \in \mathbb{Z} \end{aligned}$$

with $z(0) \in \mathcal{X}_0$, where $\mathcal{X}_0$ denotes the set of admissible initial conditions and $q = \text{ind}(E_e)$.

Let $\{E(k) \in \mathbb{R}^{n \times n}, k \in \mathbb{Z}\}$ and $\{A(k) \in \mathbb{R}^{n \times n}, k \in \mathbb{Z}\}$ be two N -periodic collections of matrices. Matrices $E(\cdot)$ and $A(\cdot)$ satisfy the σ -property if $E(k+1)A(k+1) = A(k+1)E(k)$, $k \in \mathbb{Z}$. If $E(\cdot)$ and $A(\cdot)$ satisfy the σ -property, it is straightforward to show that $E_e A_e = A_e E_e$, $E_e^D A_e = A_e E_e^D$, and $E_e A_e^D = A_e^D E_e$.

The solution of the ICADS involves the matrices $(E_e A_e^D)^k$ and $(E_e^D A_e)^k$, so in the next result we obtain expressions for these matrices in terms of the N -periodic collections of matrices $\{E(k) \in \mathbb{R}^{n \times n}, k \in \mathbb{Z}\}$ and $\{A(k) \in \mathbb{R}^{n \times n}, k \in \mathbb{Z}\}$.

PROPOSITION 4.1. *Consider two N -periodic collections $\{E(k) \in \mathbb{R}^{n \times n}, k \in \mathbb{Z}\}$ and $\{A(k) \in \mathbb{R}^{n \times n}, k \in \mathbb{Z}\}$, such that $E(\cdot)$ and $A(\cdot)$ satisfy the σ -property with $E_e = \text{diag}[E(s)]_{s=0}^{N-1}$ and the matrix A_e given in (3.1). Then*

$$(E_e A_e^D)^k = \text{diag}[E(s)^k A_{s+1}^D \phi_A(s+N+1, s+k+1)]_{s=0}^{N-1} S_n^k$$

and

$$(E_e^D A_e)^k = \text{diag}[E^D(s)^k \phi_A(s+N+1, s+N-k+1)]_{s=0}^{N-1} S_n^{-k}.$$

Proof. Since $E(\cdot)$ and $A(\cdot)$ satisfy the σ -property, we have $E_e A_e^D = A_e^D E_e$, and thus

$$(E_e A_e^D)^k = E_e^k (A_e^D)^k.$$

Hence,

$$\begin{aligned} (E_e A_e^D)^k &= (E_e)^k (A_e^D)^k \\ &= \text{diag}[(E(s))^k]_{s=0}^{N-1} \text{diag}[A_{s+1}^D \phi_A(s+N+1, s+k+1)]_{s=0}^{N-1} S_n^k \\ &= \text{diag}[E(s)^k A_{s+1}^D \phi_A(s+N+1, s+k+1)]_{s=0}^{N-1} S_n^k. \end{aligned}$$

On the other hand,

$$(E_e^D A_e)^k = (E_e^D)^k A_e^k.$$

Then,

$$(E_e^D A_e)^k = \text{diag} [E^D(s)^k \phi_A(s + N + 1, s + N - k + 1)]_{s=0}^{N-1} S_n^{-k}. \quad \square$$

Using the above result and the relation between the trajectory of the N -periodic descriptor system of difference equations and the trajectory of the ICADS given by

$$\begin{aligned} z(k) &= S_n^{-k+1} \text{col} [x(k + s)]_{s=0}^{N-1} \\ &= S_n^{-(k-1)} \text{col} [x(k + s)]_{s=0}^{N-1}, \end{aligned}$$

we can obtain the expression of the solution of the N -periodic system in terms of the Drazin inverses of the matrices $\{E(k) \in \mathbb{R}^{n \times n}, k \in \mathbb{Z}\}$ and the Drazin inverses of the N -periodic collection of monodromy matrices $\{A_s \in \mathbb{R}^{n \times n}, s \in \mathbb{Z}\}$.

By the expression

$$z(0) = S_n \hat{x}(0) = S_n \text{col} [x(s)]_{s=0}^{N-1},$$

we have

$$\begin{aligned} & (E_e^D A_e)^k E_e^D E_e z(0) \\ &= \text{diag} [E^D(s)^k \phi_A(s + N + 1, N - k + s + 1)]_{s=0}^{N-1} S_n^{-k} S_n \text{col} [x(s)]_{s=0}^{N-1} \\ &= S_n^{-(k-1)} \text{diag} [E^D(s + (k-1))^k \\ &\quad \cdot \phi_A(s + N + k - 1 + 1, N - k + k + s + 1 - 1)]_{s=0}^{N-1} \text{col} [x(s)]_{s=0}^{N-1} \\ &= S_n^{-(k-1)} \text{diag} [E^D(s + (k-1))^k \phi_A(s + N + k, s + N)]_{s=0}^{N-1} \text{col} [x(s)]_{s=0}^{N-1}. \end{aligned}$$

Using that $u_e(i) = S_m^{-i} \hat{u}(i)$ with $\hat{u}(i) = \text{col} [u(i + s)]_{s=0}^{N-1}$, we have

$$\begin{aligned} & E_e^D (E_e^D A_e)^{k-i-1} B_e u_e(i) \\ &= \text{diag} [E^D(s)^{k-i} \phi_A(s + N + 1, N - k + i + s + 2)]_{s=0}^{N-1} S_n^{-k+i+1} \text{diag} [B(s)]_{s=0}^{N-1} \\ &\quad \cdot S_m^{-i} \text{col} [u(i + s)]_{s=0}^{N-1} \\ &= S_n^{-(k-1)} \text{diag} [E^D(s + k - 1)^{k-i} \phi_A(N + k + s, N + i + s + 1) B(s + i)]_{s=0}^{N-1} \\ &\quad \cdot \text{col} [u(i + s)]_{s=0}^{N-1}. \end{aligned}$$

And since

$$u_e(k + i) = S_m^{-i} \hat{u}(k + i) = S_m^{-i} \text{col} [u(i + s)]_{s=0}^{N-1},$$

then

$$\begin{aligned} & (I - E_e^D E_e) (E_e A_e^D)^i A_e^D B_e u_e(k + i) = (I - E_e^D E_e) E_e^i (A_e^D)^{i+1} B_e u_e(k + i) \\ &= \text{diag} [I - E^D(s) E(s)]_{s=0}^{N-1} \text{diag} [E(s)^i A_{s+1}^D \phi_A(s + N + 1, i + s + 2)]_{s=0}^{N-1} \\ &\quad \cdot S_n^{i+1} \text{diag} [B(s)]_{s=0}^{N-1} S_m^{-k-i} \text{col} [u(k + i + s)]_{s=0}^{N-1} \\ &= \text{diag} [I - E^D(s) E(s)]_{s=0}^{N-1} \text{diag} [E(s)^i A_{s+1}^D \phi_A(s + N + 1, i + s + 2)]_{s=0}^{N-1} \\ &\quad \cdot S_n^{-(k-1)} \text{diag} [B(k + i + s)]_{s=0}^{N-1} \text{col} [u(k + i + s)]_{s=0}^{N-1} \\ &= S_n^{-(k-1)} \text{diag} [(I - E^D(k + s - 1) E(k + s - 1)) E(k + s - 1)^i A_{k+s}^D \\ &\quad \cdot \phi_A(N + k + s, k + i + s + 1) B(k + i + s)]_{s=0}^{N-1} \text{col} [u(k + i + s)]_{s=0}^{N-1}. \end{aligned}$$

Then

$$\begin{aligned}
& \text{col}[x(k+s)]_{s=0}^{N-1} \\
&= \text{diag}[E^D(s+(k-1))^k \phi_A(s+N+k, s+N)]_{s=0}^{N-1} \text{col}[x(s)]_{s=0}^{N-1} \\
&+ \sum_{i=0}^{k-1} \text{col}[E^D(s+k-1)^{k-i} \phi_A(N+k+s, N+i+s+1)B(s+i)]_{s=0}^{N-1} \\
&\cdot \text{col}[u(i+s)]_{s=0}^{N-1} - \sum_{i=0}^{q-1} \text{diag}\left[\left(I - E^D(k+s-1)E(k+s-1)\right) \right. \\
&\quad \cdot E(k+s-1)^i A_{k+s}^D \phi_A(N+k+s, k+i+s+1) \\
&\quad \left. \cdot B(k+i+s)\right]_{s=0}^{N-1} \text{col}[u(k+i+s)]_{s=0}^{N-1}.
\end{aligned}$$

The above result is summarized in the following theorem.

THEOREM 4.2. *Consider the system $(E(\cdot), A(\cdot), B(\cdot))_N$ given in (4.2). If $E(\cdot)$ and $A(\cdot)$ satisfy the σ -property, then the trajectory of the system is given by*

$$\begin{aligned}
x(k) &= E^D(k-1)^k \phi_A(N+k, N)x(0) \\
&+ \sum_{i=0}^{k-1} E^D(k-1)^{k-i} \phi_A(N+k, N+i+1)B(i)u(i) \\
&- (I - E^D(k-1)E(k-1)) \sum_{i=0}^{q-1} E(k-1)^i A_k^D \phi_A(N+k, k+i+1) \\
&\cdot B(k+i)u(k+i),
\end{aligned}$$

where $k \in \mathbb{Z}$, with $q = \max\{q_s, s = 0, 1, \dots, N-1\}$.

Acknowledgment. The authors are greatly indebted to the reviewers for their useful suggestions.

REFERENCES

- [1] A. BEN-ISRAEL AND T. GREVILLE, *Generalized Inverses: Theory and Applications*, 2nd ed., Springer-Verlag, New York, 2003.
- [2] A. BERMAN AND S. K. JAIN, *Nonnegative generalized inverses of powers of nonnegative matrices*, Linear Algebra Appl., 107 (1988), pp. 169–179.
- [3] R. BRU, R. CANTÓ, AND B. RICARTE, *Modelling nitrogen dynamics in citrus trees*, Math. Comput. Modelling, 38 (2003), pp. 975–987.
- [4] S. L. CAMPBELL, *Differentiation of the Drazin inverse*, SIAM J. Appl. Math., 30 (1976), pp. 703–707.
- [5] S. L. CAMPBELL AND C. D. MEYER, *Generalized Inverses of Linear Transformations*, Dover, New York, 1991.
- [6] J. J. CLIMENT, N. THOME, AND Y. WEI, *A geometrical approach on generalized inverses by Neumann-type series*, Linear Algebra Appl., 332/334 (2001), pp. 533–540.
- [7] L. DAI, *Singular Control Systems*, Springer-Verlag, New York, 1989.
- [8] J. GELONCH, *The contragredient equivalence for several matrices: A set of invariants*, Linear Algebra Appl., 291 (1999), pp. 37–49.
- [9] J. GELONCH, C. R. JOHNSON, AND P. RUBIÓ, *An extension of Flanders theorem to several matrices*, Linear and Multilinear Algebra, 43 (1997), pp. 181–200.
- [10] R. HARTWIG, G. WANG, AND Y. WEI, *Some additive results on Drazin inverse*, Linear Algebra Appl., 322 (2001), pp. 207–217.

- [11] T. KACZOREK, *Linear Control Systems, Vol. 1, Analysis of Multivariable Systems*, John Wiley and Sons, New York, 1992.
- [12] P. KOKOTOVIC, H. K. KHALIL, AND J. O'REILLY, *Singular Perturbation Methods in Control: Analysis and Design*, Classics Appl. Math. 25, SIAM, Philadelphia, 1999.
- [13] B. F. LAMOND AND M. L. PUTERMAN, *Generalized inverses in discrete time Markov decision processes*, SIAM J. Matrix Anal. Appl., 10 (1989), pp. 118–134.
- [14] M. NEUMANN AND H. J. WERNER, *Nonnegative group inverses*, Linear Algebra Appl., 151 (1991), pp. 85–96.
- [15] B. PARK AND E. I. VERRIEST, *Canonical forms on discrete lineal periodically time-varying system and a control application*, in Proceedings of the 28th IEEE Conference on Decision and Control, Trapa, FL, 1989.
- [16] S. P. SINGH AND R. W. LIU, *Existence of state equation representation of linear large-scale dynamical systems*, IEEE Trans. Circuit Theory, CT-20 (1973), pp. 239–246.
- [17] V. SERGEICHUK, *Computation of canonical matrices for chains and cycles of linear mappings*, Linear Algebra Appl., 376 (2004), pp. 235–263.
- [18] J. SREEDHAR AND P. VAN DOOREN, *Periodic descriptor systems: Solvability and conditionability*, IEEE Trans. Automat. Control, 44 (1999), pp. 310–313.
- [19] P. VAN DOOREN, *The computation of Kronecker's canonical form of a singular pencil*, Linear Algebra Appl., 27 (1979), pp. 103–140.
- [20] G. WANG, *The reverse order law for the Drazin inverses of multiple matrix products*, Linear Algebra Appl., 348 (2002), pp. 265–272.
- [21] G. WANG, Y. WEI, AND S. QIAO, *Generalized Inverses: Theory and Computation*, Science Press, Beijing, New York, 2004.
- [22] Y. WEI, *A characterization and representation of the Drazin inverse*, SIAM J. Matrix Anal. Appl., 17 (1996), pp. 744–747.

RELATIVE PERTURBATION BOUNDS FOR POSITIVE POLAR FACTORS OF GRADED MATRICES*

REN-CANG LI†

Abstract. Let B be an $m \times n$ ($m \geq n$) complex (or real) matrix. It is known that there is a unique *polar decomposition* $B = QH$, where $Q^*Q = I$, the $n \times n$ identity matrix, and H is positive definite, provided that B has full column rank. If B is perturbed to $\tilde{B}$, how do the polar factors Q and H change? This question has been investigated quite extensively, but most work so far has been on how the perturbation changed the unitary polar factor Q , with very little on the positive polar factor H , except $\|H - \tilde{H}\|_F \leq \sqrt{2}\|B - \tilde{B}\|_F$ in the Frobenius norm, due to [F. Kittaneh, *Comm. Math. Phys.*, 104 (1986), pp. 307–310], where $\tilde{Q}$ and $\tilde{H}$ are the corresponding polar factors of $\tilde{B}$. While this inequality of Kittaneh shows that H is always well behaved under perturbations, it does not tell much about smaller entries of H in the case when H 's entries vary a great deal in magnitude. This paper is intended to fill the gap by addressing the variations of H for a graded matrix $B = GS$, where S is a scaling matrix and usually diagonal (but may not be). The elements of S can vary wildly, while G is well conditioned. In such cases, the magnitudes of H 's entries indeed often vary a lot, and thus any bound on $\|H - \tilde{H}\|_F$ means little, if anything, to the accuracy of $\tilde{H}$'s smaller entries. This paper proposes a new way of measuring the errors in the H factor via bounding the scaled difference $(\tilde{H} - H)S^{-1}$, as well as accurately computing the factor when S is diagonal. Numerical examples are presented.

The results are also extended to the matrix square root of a graded positive definite matrix.

Key words. polar decomposition, positive polar factor, graded matrix, relative perturbation theory

AMS subject classifications. 15A18, 15A23, 65F15, 65F35

DOI. 10.1137/S0895479803437153

1. Introduction. Let B be an $m \times n$ ($m \geq n$) complex matrix. It is known that [11, page 152] there are Q with orthonormal column vectors, i.e., $Q^*Q = I$, and a unique positive semidefinite H such that

$$(1.1) \quad B = QH.$$

Hereafter I denotes an identity matrix with appropriate dimensions that will be clear from the context or specified. The decomposition (1.1) is called the *polar decomposition* of B . If, in addition, B has full column rank, then Q is uniquely determined, too. In fact,

$$(1.2) \quad H = (B^*B)^{1/2}, \quad Q = B(B^*B)^{-1/2},$$

where superscript “*” denotes conjugate transpose. The decomposition (1.1) can also be computed from the *singular value decomposition* (SVD) $B = U\Sigma V^*$ by

$$(1.3) \quad H = V\Sigma_1V^*, \quad Q = U_1V^*,$$

*Received by the editors October 31, 2003; accepted for publication (in revised form) by R. Mathias September 8, 2004; published electronically November 14, 2005. This material is based on work supported in part by the National Science Foundation CAREER award under grant CCR-9875201.

<http://www.siam.org/journals/simax/27-2/43715.html>

†Department of Mathematics, University of Kentucky, Lexington, KY 40506 (rccli@ms.uky.edu).

where $U = (U_1, U_2)$ and V are unitary, U_1 is $m \times n$, $\Sigma = \begin{pmatrix} \Sigma_1 \\ 0 \end{pmatrix}$ and $\Sigma_1 = \text{diag}(\sigma_1, \sigma_2, \dots, \sigma_n)$ is nonnegative.

Assume now that B is perturbed to $\tilde{B}$, and

$$(1.4) \quad B = QH \quad \text{and} \quad \tilde{B} = \tilde{Q}\tilde{H}$$

are their polar decompositions, respectively. There are many published bounds on how much the two factor matrices Q and H may change with respect to the perturbation to B being *additive* or *multiplicative*; see, e.g., [3, 6, 14, 15, 16, 17, 18, 19, 20, 22]. The additive perturbation refers to the situation when no assumption was made on how B was perturbed except possibly an assumption on the smallness of $\|\tilde{B} - B\|$ for some matrix norm $\|\cdot\|$. The multiplicative perturbation refers to the situation when $\tilde{B} = D_1^* B D_2$ such that D_1 and D_2 are assumed close to identity matrices. The perturbation for the *one-sided scaling* case (or the *graded* case) can be translated into a perturbation of this kind [16]. By one-sided scaling we mean that B and $\tilde{B}$ take the forms

$$(1.5) \quad B = GS, \quad \tilde{B} = \tilde{G}S \equiv (G + \Delta G)S,$$

where S is a scaling matrix and usually diagonal (but this is not necessary for some theorems below). The elements of S can vary wildly. G has full column rank and is usually better conditioned than B itself; i.e., the ratio of G 's largest singular value over its smallest one is much smaller than that for B when S has entries of widely varying magnitude.

Much work in the past was done for the additive perturbation (see, e.g., [3, 5, 6, 10, 12, 14, 15, 17, 18, 19, 20, 22]), except [16], which addressed the multiplicative perturbation. In these works, most attention was given to how the unitary factor Q changed with respect to the two different kinds of perturbations and to the number fields—real *vs.* complex. Perhaps this was due to the fact that the study on the perturbation of H could be considered *complete*, owing to the following result [13, Theorem 2]:

$$(1.6) \quad \|H - \tilde{H}\|_F \leq \sqrt{2}\|B - \tilde{B}\|_F,$$

where $\|\cdot\|_F$ denotes the Frobenius norm. This paper, motivated by the following example, concerns the H factor for the graded case.

Example 1.1. Let

$$(1.7) \quad B = GS = \begin{pmatrix} 6 & -2 & 14 & -5 \\ 8 & 5 & -7 & -8 \\ -2 & -11 & 2 & -3 \\ 5 & -8 & -16 & 9 \end{pmatrix} \begin{pmatrix} 10^6 & & & \\ & 10^4 & & \\ & & 10^2 & \\ & & & 1 \end{pmatrix}.$$

We take ΔG , a random complex matrix (in MATLAB: $10^{-5} * \text{randn}(4)$). Then (1.6) implies $\|H - \tilde{H}\|_F \leq \sqrt{2}\|\Delta GS\|_F \approx 24.7393$. However, the true H is (only the first five decimal digits in each entry are presented here)

$$(1.8) \quad H = \begin{pmatrix} 1.1358 \cdot 10^7 & 8.6928 \cdot 10^3 & -4.9320 \cdot 10^2 & -3.7828 \cdot 10^0 \\ 8.6928 \cdot 10^3 & 1.4603 \cdot 10^5 & 3.1908 \cdot 10^2 & -4.4827 \cdot 10^0 \\ -4.9320 \cdot 10^2 & 3.1908 \cdot 10^2 & 2.1691 \cdot 10^3 & -7.7287 \cdot 10^0 \\ -3.7828 \cdot 10^0 & -4.4827 \cdot 10^0 & -7.7287 \cdot 10^0 & 9.2121 \cdot 10^0 \end{pmatrix},$$

which, together with $\|H - \tilde{H}\|_F \leq 24.7393$, gives no assurance of the correctness of smaller entries in $\tilde{H}$! Actually, all the entries of H are determined to high relative accuracy by the data, as we shall see soon.

2. Main result. With (1.4) and (1.5), write $\tilde{Q} = Q + \Delta Q$. We have

$$\begin{aligned}\tilde{Q}\tilde{H} &= \tilde{G}S, \\ \tilde{H}S^{-1} &= \tilde{Q}^*\tilde{G} \\ &= \tilde{Q}^*(G + \Delta G) \\ &= (Q + \Delta Q)^*G + \tilde{Q}^*\Delta G \\ &= Q^*G + \Delta Q^*G + \tilde{Q}^*\Delta G \\ &= HS^{-1} + \Delta Q^*G + \tilde{Q}^*\Delta G.\end{aligned}$$

Thus $(\tilde{H} - H)S^{-1} = \Delta Q^*G + \tilde{Q}^*\Delta G$, and

$$(2.1) \quad \|(\tilde{H} - H)S^{-1}\|_F \leq \|\Delta Q^*\|_F \|G\|_2 + \|\Delta G\|_F,$$

where $\|\cdot\|_2$ denotes the spectral norm. It is proved in [16] that

$$(2.2) \quad \|\Delta Q\|_F \leq \sqrt{\|(\Delta G)G^\dagger\|_F^2 + \left\|I - (I + (\Delta G)G^\dagger)^{-1}\right\|_F^2}$$

$$(2.3) \quad \leq \sqrt{1 + \frac{1}{(1 - \|G^\dagger\|_2 \|\Delta G\|_2)^2}} \|G^\dagger\|_2 \|\Delta G\|_F,$$

where $G^\dagger = (G^*G)^{-1}G^*$ is the pseudoinverse of G . A consequence of (2.1), (2.2), and (2.3) is the following theorem, which states that up to the first order $\|(\tilde{H} - H)S^{-1}\|_F$ is bounded by $[\sqrt{2}\kappa(G) + 1]\|\Delta G\|_F$, where $\kappa(G) \stackrel{\text{def}}{=} \|G\|_2 \|G^\dagger\|_2$.

THEOREM 2.1. *Let $B = GS$ and $\tilde{B} = \tilde{G}S$ be two $m \times n$ matrices having full column rank with the polar decompositions (1.4). S is $n \times n$ and nonsingular. If $\|\Delta G\|_2 \|G^\dagger\|_2 < 1$, then*

$$(2.4) \quad \|(\tilde{H} - H)S^{-1}\|_F \leq \sqrt{\|(\Delta G)G^\dagger\|_F^2 + \left\|I - (I + (\Delta G)G^\dagger)^{-1}\right\|_F^2} \|G\|_2 + \|\Delta G\|_F$$

$$(2.5) \quad \leq \left(\sqrt{1 + \frac{1}{(1 - \|G^\dagger\|_2 \|\Delta G\|_2)^2}} \|G^\dagger\|_2 \|G\|_2 + 1 \right) \|\Delta G\|_F.$$

The scaled difference between H and $\tilde{H}$ in Theorem 2.1 is measured by the Frobenius norm of $\|(\tilde{H} - H)S^{-1}\|_F$. But since the H factors are Hermitian, some readers may prefer some kind of symmetric scaling, i.e., two-sided scaling. This can be easily done when S is diagonal. Let $|S|$ be the diagonal matrix obtained by taking entrywise absolute value on S . Then $|S|^{-1/2}$ is well defined, and $|S|^{-1/2}(\tilde{H} - H)|S|^{-1/2}$ is Hermitian and has the same eigenvalues as $(\tilde{H} - H)|S|^{-1}$. Therefore for diagonal S

$$(2.6) \quad \||S|^{-1/2}(\tilde{H} - H)|S|^{-1/2}\|_F \leq \|(\tilde{H} - H)|S|^{-1}\|_F = \|(\tilde{H} - H)S^{-1}\|_F.$$

More generally it can be proved that (2.6) holds for normal S with $|S|$ being interpreted as $(S^*S)^{1/2}$, which in the case of diagonal S is the same as taking entrywise

absolute values. We now outline a proof. Assume that S is normal and let $S = X\Lambda X^*$ be its eigendecomposition, where X is unitary and Λ is diagonal. Then $|S| = X|\Lambda|X^*$ and $|S|^{-1/2} = X|\Lambda|^{-1/2}X^*$. We have

$$\begin{aligned} \||S|^{-1/2}(\tilde{H} - H)|S|^{-1/2}\|_F &= \||\Lambda|^{-1/2}X^*(\tilde{H} - H)X|\Lambda|^{-1/2}\|_F \\ &\leq \|X^*(\tilde{H} - H)X|\Lambda|^{-1}\|_F \\ &= \|X^*(\tilde{H} - H)X\Lambda^{-1}\|_F \\ &= \|(\tilde{H} - H)X\Lambda^{-1}X^*\|_F \\ &= \|(\tilde{H} - H)S^{-1}\|_F, \end{aligned}$$

as expected. So we have¹ the following corollary.

COROLLARY 2.2. *To the conditions of Theorem 2.1 add that S is normal (and thus diagonal S is included). Then*

$$(2.7) \quad \begin{aligned} &\||S|^{-1/2}(\tilde{H} - H)|S|^{-1/2}\|_F \\ &\leq \sqrt{\|(\Delta G)G^\dagger\|_F^2 + \left\|I - (I + (\Delta G)G^\dagger)^{-1}\right\|_F^2} \|G\|_2 + \|\Delta G\|_F \end{aligned}$$

$$(2.8) \quad \leq \left(\sqrt{1 + \frac{1}{(1 - \|G^\dagger\|_2\|\Delta G\|_2)^2}} \|G^\dagger\|_2\|G\|_2 + 1 \right) \|\Delta G\|_F.$$

We caution the reader that the bounds on $\||S|^{-1/2}(\tilde{H} - H)|S|^{-1/2}\|_F$ in Corollary 2.2 can overestimate the errors in some entries of $\tilde{H}$ much more than the bounds on $\|(\tilde{H} - H)S^{-1}\|_F$ in Theorem 2.1 do, even though bounding $\||S|^{-1/2}(\tilde{H} - H)|S|^{-1/2}\|_F$ seems to be more mathematically elegant. In a moment we shall revisit Example 1.1 to show our point. For now, let us analyze the case when S is diagonal. Suppose

$$\||S|^{-1/2}(\tilde{H} - H)|S|^{-1/2}\|_F \leq \|(\tilde{H} - H)S^{-1}\|_F \leq \beta.$$

Write $S = \text{diag}(s_1, s_2, \dots)$, $H = (h_{ij})$, and $\tilde{H} = (\tilde{h}_{ij})$. Entrywise, $\||S|^{-1/2}(\tilde{H} - H)|S|^{-1/2}\|_F \leq \beta$ gives

$$(2.9) \quad |h_{ij} - \tilde{h}_{ij}| \leq \beta \sqrt{|s_i s_j|},$$

and $\|(\tilde{H} - H)S^{-1}\|_F \leq \beta$ gives

$$(2.10) \quad |h_{ij} - \tilde{h}_{ij}| = |h_{ji} - \tilde{h}_{ji}| \leq \beta \min\{|s_i|, |s_j|\}.$$

If $|s_i| \gg |s_j|$ or $|s_i| \ll |s_j|$, (2.9) will be much less sharp than (2.10). Accordingly, the numbers of correct significant decimal digits defined as

$$(2.11) \quad -\log_{10}(|h_{ij} - \tilde{h}_{ij}|/|h_{ij}|)$$

in $\tilde{H}$ are at least

$$-\log_{10}(\beta \sqrt{|s_i s_j|}/|h_{ij}|) \text{ by (2.9), or } -\log_{10}(\beta, \min\{|s_i|, |s_j|\}/|h_{ij}|) \text{ by (2.10).}$$

¹One may also see that $\|S^{-1/2}(\tilde{H} - H)S^{-1/2}\|_F = \||S|^{-1/2}(\tilde{H} - H)|S|^{-1/2}\|_F$ for any square root S of a normal S . But then $S^{-1/2}(\tilde{H} - H)S^{-1/2}$ is no longer Hermitian.

The case $h_{ij} = 0$ exactly is special and an exception in the sense that either $\tilde{h}_{ij} = 0 = h_{ij}$ exactly or it has no correct significant decimal digits at all. Certainly $\|(\tilde{H} - H)S^{-1}\|_F \leq \beta$ does not guarantee $\tilde{h}_{ij} = 0$ in the event that $h_{ij} = 0$ no matter how tiny β may be, but rather it says that $|\tilde{h}_{ij}|$ is scaledly tiny in the sense of (2.10). Nonetheless, the definition (2.11) of the number of correct significant decimal digits in $\tilde{h}_{ij}$ remains mathematically valid by setting $0/0 = 0$. Then in the event that $h_{ij} = 0$, $-\log_{10}(|h_{ij} - \tilde{h}_{ij}|/|h_{ij}|) = \pm\infty$, corresponding to two possible cases: $\tilde{h}_{ij} = 0$ or not.

Example 1.1 (continued). Use the same ΔG . Note that S is diagonal and positive definite. Corollary 2.2 implies

$$\begin{aligned} \|S^{-1/2}(\tilde{H} - H)S^{-1/2}\|_F &\leq \|(\tilde{H} - H)S^{-1}\|_F \\ &\leq 1.4298 \cdot 10^{-4} && \text{(by (2.7))} \\ &\leq 2.0756 \cdot 10^{-4} && \text{(by (2.8)).} \end{aligned}$$

Take $\beta = 1.4298 \cdot 10^{-4}$ in (2.9) and (2.10). Two sets of lower bounds on the numbers of correct significant decimal digits in $\tilde{h}_{ij}$ obtained from using, respectively, $\|S^{-1/2}(\tilde{H} - H)S^{-1/2}\|_F \leq \beta$ and $\|(\tilde{H} - H)S^{-1}\|_F \leq \beta$, are, entrywise,

$$\begin{pmatrix} 4.9 & 2.8 & 2.5 & 1.4 \\ 2.8 & 5.0 & 3.3 & 2.5 \\ 2.5 & 3.3 & 5.2 & 3.7 \\ 1.4 & 2.5 & 3.7 & 4.8 \end{pmatrix}, \quad \begin{pmatrix} 4.9 & 3.8 & 4.5 & 4.4 \\ 3.8 & 5.0 & 4.3 & 4.5 \\ 4.5 & 4.3 & 5.2 & 4.7 \\ 4.4 & 4.5 & 4.7 & 4.8 \end{pmatrix}.$$

These bounds yield two conclusions. First, all entries of $\tilde{H}$ have at least one correct decimal digit, and the diagonal entries have at least four correct decimal digits, much sharper than the bound (1.6) indicates. Second, $\|(\tilde{H} - H)S^{-1}\|_F \leq \beta$ can be potentially sharper than $\|S^{-1/2}(\tilde{H} - H)S^{-1/2}\|_F \leq \beta$ when it comes to off-diagonal entries.

3. Stable computation of H . Given the perturbation theory we have developed in section 2, how do we compute the factor H as accurately as predicted? It is not clear if this can be done for any arbitrary S . But we shall show it is always possible for *diagonal* S :

$$(3.1) \quad \boxed{\begin{array}{l} 1. \text{ Compute SVD of } B \equiv GS = U\Sigma V^* \text{ by one-sided Jacobi [8];} \\ 2. \text{ Set } Q = UV^*, W = Q^*G, \text{ and then } H = WS. \end{array}}$$

It is also tempting to compute H as $V\Sigma V^*$. But as we shall show by Example 3.1 below, doing so can destroy the high relative accuracy in V and Σ delivered by the one-sided Jacobi method. The following theorem shows that (3.1) will deliver H with a small scaled error on machines having the floating point arithmetic model

$$(3.2) \quad \text{computed } \alpha \odot \beta \text{ is } (\alpha \odot \beta)(1 + \delta), \quad |\delta| \leq \epsilon_m \text{ for } \odot \in \{+, -, \times, \div\},$$

where ϵ_m is the machine roundoff for the working precision. All current commercially significant machines run the IEEE floating point arithmetic [1, 9] and thus satisfy (3.2).

THEOREM 3.1. *The computed $\tilde{H}$ by (3.1) satisfies $\|(\tilde{H} - H)S^{-1}\|_F = \mathcal{O}(\epsilon_m \kappa(G)\|G\|_F)$.*

Proof. Denote all corresponding computed matrices by the same symbols except with *tildes*. It is proved in Algorithm 3.1 and Theorems 3.1 and 3.2 and their proofs in [8] that the computed SVD $\tilde{U}\tilde{\Sigma}\tilde{V}^*$ is the exact SVD of a nearby $\tilde{B}$:

$$\tilde{B} = (I + E)B(I + F) \quad \text{satisfying} \quad \|E\|_F = \mathcal{O}(\epsilon_m \kappa(G)), \quad \|F\|_F = \mathcal{O}(\epsilon_m \kappa(G)).$$

Note also that hiding in the two $\mathcal{O}(\cdot)$ are some modest increasing functions of n . It can be seen that the computed $\tilde{Q}$, $\tilde{W}$, and $\tilde{H}$ satisfy exactly

$$(3.3) \quad \tilde{Q} = \tilde{U}\tilde{V}^* + E_1, \quad \tilde{W} = \tilde{Q}^*G + E_2, \quad \tilde{H} = (\tilde{W}S) \circ M,$$

where $\circ$ denote the entrywise Hadamard product,

$$\|E_1\|_F = \mathcal{O}(\epsilon_m), \quad \|E_2\|_F = \mathcal{O}(\epsilon_m \|G\|_F),$$

and each M 's (i, j) th entry $m_{ij} = 1 + \mathcal{O}(\epsilon_m)$ since S is diagonal. The first two equations in (3.3) are results of analysis in [24, page 115]; the last equation is a consequence of the model (3.2) and the entrywise Hadamard product. With those equations, we get

$$\begin{aligned} \tilde{H}S^{-1} &= \tilde{W} \circ M \\ &= (\tilde{Q}^*G + E_2) \circ M \\ &= [(\tilde{U}\tilde{V}^* + E_1)^*G + E_2] \circ M \\ &= [(\tilde{U}\tilde{V}^*)^*G] \circ M + (E_1^*G + E_2) \circ M. \end{aligned}$$

Now by [16, Theorem 1], we have $\tilde{U}\tilde{V}^* = Q + E_3$ satisfying

$$\|E_3\|_F = \mathcal{O}(\|E\|_F + \|F\|_F) = \mathcal{O}(\epsilon_m \kappa(G)).$$

Write $M = J + E_4$, where J 's entries are all 1's and $\|E_4\|_F = \mathcal{O}(\epsilon_m)$. Therefore noticing $Q^*G = HS^{-1}$, we have

$$\begin{aligned} \tilde{H}S^{-1} &= (Q^*G) \circ M + (E_3^*G + E_1^*G + E_2) \circ M \\ &= (HS^{-1}) \circ (J + E_4) + (E_3^*G + E_1^*G + E_2) \circ M \\ &= HS^{-1} + (Q^*G) \circ E_4 + (E_3^*G + E_1^*G + E_2) \circ M, \end{aligned}$$

which gives

$$\|(\tilde{H} - H)S^{-1}\|_F \leq \|(Q^*G) \circ E_4 + (E_3^*G + E_1^*G + E_2) \circ M\|_F = \mathcal{O}(\epsilon_m \kappa(G) \|G\|_F),$$

as expected. $\square$

Example 3.1. $n = 10$, and

$$G = \begin{pmatrix} 4.656 & 7.220 & 3.831 & 2.924 & 7.556 & 7.329 & 4.105 & 2.827 & 6.787 & 7.7860 \\ 4.187 & 2.644 & 5.986 & 9.774 & 7.660 & 7.170 & 1.042 & 2.240 & 5.235 & 6.1470 \\ 7.518 & 6.780 & 5.691 & 1.071 & 8.527 & 2.231 & 0.220 & 7.386 & 9.997 & 4.8770 \\ 3.829 & 7.801 & 2.164 & 1.158 & 1.907 & 4.992 & 7.945 & 9.693 & 7.529 & 2.1330 \\ 3.515 & 1.907 & 9.999 & 0.686 & 2.950 & 2.240 & 5.921 & 9.097 & 5.180 & 3.4390 \\ 0.871 & 4.511 & 4.406 & 3.102 & 4.731 & 1.793 & 3.100 & 9.608 & 6.400 & 6.9290 \\ 8.565 & 8.088 & 9.318 & 3.557 & 6.034 & 9.012 & 1.087 & 8.167 & 1.351 & 4.6020 \\ 1.351 & 5.289 & 3.025 & 3.591 & 1.210 & 2.123 & 8.771 & 4.650 & 8.002 & 9.6420 \\ 2.408 & 2.745 & 3.893 & 4.202 & 5.845 & 3.501 & 0.603 & 2.775 & 0.912 & 9.1680 \\ 0.231 & 2.726 & 6.898 & 9.243 & 3.813 & 5.065 & 0.594 & 8.415 & 5.140 & 1.9450 \end{pmatrix},$$

$$S = \text{diag}(10^3, 10^8, 10^5, 10^4, 1, 10^4, 10^9, 10^8, 10^3, 10^8).$$

For this example, $\kappa(B) = 5.37 \cdot 10^{10}$, $\kappa(G) = 4.60 \cdot 10^2$, and $\|G\|_F = 5.02 \cdot 10^1$. Throughout computations, IEEE single precision was used and thus $\epsilon_m = 2^{-24} \approx 5.96 \times 10^{-8}$. We compared accuracy of numerical results via the one-sided Jacobi SVD code provided to us by Dr. Z. Drmač and the SVD code `sgesvd` by the QR algorithm from LAPACK [2]. The exact Q and H are obtained by Maple² with `Digits:=50`. The following table lists various errors by the two methods, with Q and H computed as in (3.1).

	$\ (\tilde{H} - H)S^{-1}\ _F$	$\ \tilde{H} - H\ _F$	$\ \tilde{Q} - Q\ _F$	$\max \tilde{\sigma} - \sigma /\sigma$
Jacobi	1.19e-5	8.83e+2	1.75e-6	4.55e-6
QR	8.67e+1	2.47e+3	2.00e+0	2.19e+1

Here the last column is for the maximum relative errors among all computed singular values. The third column does not really suggest any accuracy advantage of the one-sided Jacobi SVD over `sgesvd` due to the inadequacy of $\|\tilde{H} - H\|_F$ in this graded example because it merely reflects errors in the largest entries of $\tilde{H}$. All errors associated with QR are unacceptably large, as expected; this is because SVD by QR can obscure all singular values by as much as $\mathcal{O}(\epsilon_m \|B\|_2)$, which in this example is bigger than the smallest singular values, and consequently, corresponding computed singular vectors are very much inaccurate.

We note in passing that H , if computed as $V\Sigma V^*$, can be very inaccurate even with highly relatively accurate V and Σ by one-sided Jacobi SVD. In fact for this example, we get $\|(\tilde{H} - H)S^{-1}\|_F = 1.68e + 1$ if doing so.

4. Extensions to the matrix square root of a positive definite matrix.

There is a natural extension of the theory in section 2 to the perturbation of the matrix square root of a positive definite matrix that allows some kind of symmetric scaling for better conditioning. By this we mean $A = S^*TS$, where S , as above, is a scaling matrix (and usually diagonal), and T is positive definite and well conditioned; i.e., $\|T\|_2\|T^{-1}\|_2$ is of moderate magnitude. Then we have

$$A = S^*T^{1/2}T^{1/2}S = B^*B,$$

where B takes the form in (1.5) with $G = T^{1/2}$. Assume now that A is perturbed to $\tilde{A} = S^*\tilde{T}S$ such that $\Delta T \stackrel{\text{def}}{=} \tilde{T} - T$ is sufficiently tiny. Then

$$\tilde{A} = S^*(T + \Delta T)S = S^*T^{1/2} \left[I + T^{-1/2}(\Delta T)T^{-1/2} \right] T^{1/2}S = \tilde{B}^*\tilde{B},$$

where $\tilde{B}$ takes the form in (1.5) with

$$\tilde{G} = (I + \hat{T})^{1/2}T^{1/2}, \quad \hat{T} \stackrel{\text{def}}{=} T^{-1/2}(\Delta T)T^{-1/2}.$$

We have

$$\begin{aligned} \Delta G &\stackrel{\text{def}}{=} \tilde{G} - G = [(I + \hat{T})^{1/2} - I]G, \\ (\Delta G)G^{-1} &= (I + \hat{T})^{1/2} - I, \\ I - (I + (\Delta G)G^{-1})^{-1} &= I - (I + \hat{T})^{-1/2}. \end{aligned}$$

²<http://www.maplesoft.com/>

Let $\delta_p = \|\hat{T}\|_p$, where $p = 2, F$. It can be verified that

$$(4.1) \quad \|(I + \hat{T})^{1/2} - I\|_p \leq \frac{\delta_p}{1 + \sqrt{1 - \delta_2}},$$

$$(4.2) \quad \|I - (I + \hat{T})^{-1/2}\|_p \leq \frac{\delta_p}{(1 + \sqrt{1 - \delta_2}) \sqrt{1 - \delta_2}}.$$

Let the polar decompositions of B and $\tilde{B}$ be given as (1.4). Then

$$A^{1/2} = H, \quad \tilde{A}^{1/2} = \tilde{H}.$$

Now apply Theorem 2.1 to get the following theorem.

THEOREM 4.1. *Let $A = S^*TS$ and $\tilde{A} = S^*\tilde{T}S$ be two $n \times n$ positive definite matrices, where S is $n \times n$ and nonsingular, and let $\hat{T} \stackrel{\text{def}}{=} T^{-1/2}(\Delta T)T^{-1/2}$ and $\delta_p = \|\hat{T}\|_p$, where $p = 2, F$. If $\delta_2 < 1$, then*

$$(4.3) \quad \|(\tilde{A}^{1/2} - A^{1/2})S^{-1}\|_F \leq \left(\sqrt{\frac{2 - \delta_2}{1 - \delta_2}} + 1 \right) \|T^{1/2}\|_2 \frac{\delta_F}{1 + \sqrt{1 - \delta_2}},$$

$$(4.4) \quad \approx \frac{\sqrt{2} + 1}{2} \|T^{1/2}\|_2 \delta_F.$$

A corollary of this theorem similar to Corollary 2.2 can be stated by noting that for normal S ,

$$\| |S|^{-1/2}(\tilde{A}^{1/2} - A^{1/2})|S|^{-1/2} \|_F \leq \|(\tilde{A}^{1/2} - A^{1/2})S^{-1}\|_F.$$

But we shall omit the detail. Stronger bounds than those in Theorem 4.1 can be proved if ΔT is itself positive semidefinite due to stronger bounds than (4.1) and (4.2). In fact, it can be proved that *if ΔT is positive semidefinite, then*

$$(4.5) \quad \|(I + \hat{T})^{1/2} - I\|_2 = \frac{\delta_2}{1 + \sqrt{1 + \delta_2}},$$

$$(4.6) \quad \|I - (I + \hat{T})^{-1/2}\|_2 = \frac{\delta_2}{(1 + \sqrt{1 + \delta_2}) \sqrt{1 + \delta_2}},$$

$$(4.7) \quad \|(I + \hat{T})^{1/2} - I\|_F, \|I - (I + \hat{T})^{-1/2}\|_F \leq \frac{\delta_F}{2}.$$

Denote the set of eigenvalues of ΔT by $\lambda(\Delta T)$, and notice $0 \leq \mu \in \lambda(\Delta T)$ and $\delta_2 = \max_{\mu \in \lambda(\Delta T)} \mu$ if ΔT is positive semidefinite. We have

$$\|(I + \hat{T})^{1/2} - I\|_2 = \max_{\mu \in \lambda(\Delta T)} \left(\sqrt{1 + \mu} - 1 \right) = \sqrt{1 + \delta_2} - 1,$$

$$\|I - (I + \hat{T})^{-1/2}\|_2 = \max_{\mu \in \lambda(\Delta T)} \left(1 - \frac{1}{\sqrt{1 + \mu}} \right) = 1 - \frac{1}{\sqrt{1 + \delta_2}},$$

which are (4.5) and (4.6), respectively. To prove (4.7), we have

$$\|(I + \hat{T})^{1/2} - I\|_F^2 = \sum_{\mu \in \lambda(\Delta T)} \left(\sqrt{1 + \mu} - 1 \right)^2 = \sum_{\mu \in \lambda(\Delta T)} \frac{\mu^2}{(1 + \sqrt{1 + \mu})^2} \leq \frac{\delta_F^2}{4},$$

$$\|I - (I + \hat{T})^{-1/2}\|_2 = \sum_{\mu \in \lambda(\Delta T)} \left(1 - \frac{1}{\sqrt{1 + \mu}} \right)^2 \leq \frac{\delta_F^2}{4}.$$

Now apply Theorem 2.1 to get $\|(\tilde{A}^{1/2} - A^{1/2})S^{-1}\|_F \leq \frac{\sqrt{2}+1}{2}\|T^{1/2}\|_2\delta_F$, which is the right-hand side of (4.4).

For the same reason as for the positive polar factor, Theorem 4.1 can provide a much sharper bound than the existing one [4, 23],

$$(4.8) \quad \|\tilde{A}^{1/2} - A^{1/2}\|_F \leq \frac{1}{\|\tilde{A}^{-1/2}\|_2^{-1} + \|A^{-1/2}\|_2^{-1}} \|\tilde{A} - A\|_F.$$

Example 4.1. Take $A = B^*B$ with B as in (1.7), i.e.,

$$T = \begin{pmatrix} 129 & 10 & -56 & -43 \\ 10 & 214 & 43 & -69 \\ -56 & 43 & 505 & -164 \\ -43 & -69 & -164 & 179 \end{pmatrix}, \quad S = \begin{pmatrix} 10^6 & & & \\ & 10^4 & & \\ & & 10^2 & \\ & & & 1 \end{pmatrix}.$$

Then $A^{1/2}$ is B 's positive polar factor, which is given as in (1.8). Take $E = 10^{-5} * \text{randn}(4)$ and $\Delta T = E + E^* + EE^*$. Then (4.8) yields a bound

$$\|\tilde{A}^{1/2} - A^{1/2}\|_F \leq 1.4288 \cdot 10^6,$$

which is too big to be of any use. However, our new bound (4.3) produces

$$\|(\tilde{A}^{1/2} - A^{1/2})S^{-1}\|_F \leq 1.17363 \cdot 10^{-5}.$$

But can we compute $A^{1/2}$ as accurately as predicted by Theorem 4.1? Indeed we can. Analogously to section 3, we have the following algorithm that will deliver a highly accurately computed $A^{1/2}$:

$$(4.9) \quad \begin{array}{l} 1. \text{ Decompose } T = G^*G \text{ (e.g., Cholesky decomposition [7]);} \\ 2. \text{ Compute polar decomposition } GS = QH \text{ by (3.1), and} \\ \text{ then return } A^{1/2} = H. \end{array}$$

Step 2 of (4.9) is guaranteed to compute a highly accurate H for $\tilde{G}S$, given the computed $\tilde{G}$ by Step 1. Thus it suffices to show that $\tilde{G}$ is the Cholesky factor of a nearby T in order to convince ourselves that (4.9) will work. This has been done in [7, 21] from which one can conclude that the computed $\tilde{G}$ indeed satisfies $\tilde{G}^*\tilde{G} = T + \Delta T$ with $\|\Delta T\|_F = \mathcal{O}(\epsilon_m \|A\|_F)$.

Acknowledgments. The author wishes to thank Dr. Z. Drmač for his one-sided Jacobi SVD code and Mr. James Money for running numerical Example 3.1. He is also grateful to an anonymous referee for his many detailed comments that improved the presentation of this paper.

REFERENCES

- [1] INSTITUTE OF ELECTRICAL AND ELECTRONIC ENGINEERS, *IEEE Standard for Binary Floating-Point Arithmetic*, IEEE Std 754-1985, IEEE, New York, 1985.
- [2] E. ANDERSON, Z. BAI, C. BISCHOF, J. DEMMEL, S. BLACKFORD, J. DONGARRA, J. DU CROZ, A. GREENBAUM, S. HAMMARLING, A. MCKENNEY, AND D. SORENSSEN, *LAPACK Users' Guide*, Software Environ. Tools 9, 3rd ed., SIAM, Philadelphia, 1999.
- [3] A. BARRLUND, *Perturbation bounds on the polar decomposition*, BIT, 30 (1990), pp. 101–113.
- [4] R. BHATIA, *Some inequalities for norm ideals*, Comm. Math. Phys., 111 (1987), pp. 33–39.

- [5] F. CHATELIN AND S. GRATTON, *On the condition numbers associated with the polar factorization of a matrix*, Numer. Linear Algebra Appl., 7 (2000), pp. 337–354.
- [6] C.-H. CHEN AND J.-G. SUN, *Perturbation bounds for the polar factors*, J. Comput. Math., 7 (1989), pp. 397–401.
- [7] J. W. DEMMEL, *Applied Numerical Linear Algebra*, SIAM, Philadelphia, 1997.
- [8] J. W. DEMMEL, M. GU, S. C. EISENSTAT, I. SLAPNIČAR, K. VESELIĆ, AND Z. DRMAČ, *Computing the singular value decomposition with high relative accuracy*, Linear Algebra Appl., 299 (1999), pp. 21–80.
- [9] D. GOLDBERG, *What every computer scientist should know about floating-point arithmetic*, Comput. Surveys, 232 (1991), pp. 5–48.
- [10] N. J. HIGHAM, *Computing the polar decomposition—with applications*, SIAM J. Sci. Statist. Comput., 7 (1986), pp. 1160–1174.
- [11] R. A. HORN AND C. R. JOHNSON, *Topics in Matrix Analysis*, Cambridge University Press, Cambridge, UK, 1991.
- [12] C. KENNEY AND A. J. LAUB, *Polar decomposition and matrix sign function condition estimates*, SIAM J. Sci. Statist. Comput., 12 (1991), pp. 488–504.
- [13] F. KITTANEH, *Inequalities for the Schatten p -norm. III*, Comm. Math. Phys., 104 (1986), pp. 307–310.
- [14] R.-C. LI, *A perturbation bound for the generalized polar decomposition*, BIT, 33 (1993), pp. 304–308.
- [15] R.-C. LI, *New perturbation bounds for the unitary polar factor*, SIAM J. Matrix Anal. Appl., 16 (1995), pp. 327–332.
- [16] R.-C. LI, *Relative perturbation bounds for the unitary polar factor*, BIT, 37 (1997), pp. 67–75.
- [17] W. LI AND W. SUN, *Perturbation bounds of unitary and subunitary polar factors*, SIAM J. Matrix Anal. Appl., 23 (2002), pp. 1183–1193.
- [18] W. LI AND W. SUN, *New perturbation bounds for unitary polar factors*, SIAM J. Matrix Anal. Appl., 25 (2003), pp. 362–372.
- [19] J.-Q. MAO, *The perturbation analysis of the product of singular vector matrices UV^H* , J. Comput. Math., 4 (1986), pp. 245–248.
- [20] R. MATHIAS, *Perturbation bounds for the polar decomposition*, SIAM J. Matrix Anal. Appl., 14 (1993), pp. 588–597.
- [21] J.-G. SUN, *Rounding-error and perturbation bounds for the Cholesky and LDL^T factorizations*, Linear Algebra Appl., 173 (1992), pp. 77–97.
- [22] J.-G. SUN AND C.-H. CHEN, *Generalized polar decomposition*, Math. Numer. Sin., 11 (1989), pp. 262–273 (in Chinese).
- [23] J. L. VAN HEMMEN AND T. ANDO, *An inequality for trace ideals*, Comm. Math. Phys., 76 (1980), pp. 143–148.
- [24] J. H. WILKINSON, *The Algebraic Eigenvalue Problem*, Clarendon Press, Oxford, UK, 1965.

A NEW APPROACH TO GENERALIZED SINGULAR VALUE DECOMPOSITION*

SHMUEL FRIEDLAND†

Abstract. We propose a new algorithm to find the generalized singular value decompositions of two matrices with the same number of columns. We discuss in detail the sensitivity of our algorithm to errors in the entries of the matrices and suggest a way to suppress this sensitivity.

Key words. gene expression matrix, singular value decomposition, extended singular value decomposition, generalized singular value decomposition

AMS subject classifications. 15A18, 92D10

DOI. 10.1137/S0895479804439791

1. Introduction. The singular value decomposition (SVD) used in mathematics [6] and in numerical computations [5] is a very useful and versatile tool. It is used in statistics in principal component analysis (PCA) [7] and recently became very useful in the analysis of DNA microarrays [1].

Let $\mathbb{R}^m$, $M_{mn}(\mathbb{R})$, $O_{md}(\mathbb{R})$, $M_n(\mathbb{R})$, $S_n(\mathbb{R})$, $GL_n(\mathbb{R})$, $O_n(\mathbb{R})$ be the linear space of real column vectors with m coordinates, the linear space of real $m \times n$ matrices, the subset of $m \times d$ matrices whose d ($\leq m$) columns form an orthonormal system, the algebra of $n \times n$ real matrices, the subspace of $n \times n$ real symmetric matrices, the group of $n \times n$ real invertible matrices, and the subgroup of $n \times n$ orthogonal matrices, respectively. As usual, for $S \in S_n(\mathbb{R})$ we write $S \geq 0$ if S is nonnegative definite and $S > 0$ if S is positive definite. As in [5], denote by $\text{diag}(d_1, \dots, d_{\min(m,n)}) \in M_{mn}(\mathbb{R})$ the diagonal matrix with the diagonal entry d_i on the (i, i) position, where $i = 1, \dots, \min(m, n)$ and all other entries are equal to zero.

For $A \in M_{mn}(\mathbb{R})$ the SVD of A is given by $A = U\Sigma V^T$, where the following variations are popular: For the standard SVD $U \in O_m(\mathbb{R})$, $V \in O_n(\mathbb{R})$, $\Sigma = (s_{ij})_{i,j=1}^{m,n} \in M_{mn}(\mathbb{R})$, where Σ is a diagonal matrix with $s_{11} = \sigma_1 \geq \dots \geq s_{dd} = \sigma_d > 0 = s_{ii} = \sigma_i = 0$, $i = d+1, \dots, \min(m, n)$. For the reduced SVD

$$(1.1) \quad A = U\Sigma V^T, \quad U \in O_{md}(\mathbb{R}), \quad V \in O_{nd}(\mathbb{R}), \quad 0 < \Sigma = \text{diag}(\sigma_1, \dots, \sigma_d) \in S_d(\mathbb{R}).$$

Note that d is the rank of A , denoted by $\text{rank } A$. The columns of U and the rows of V^T form an orthonormal basis of the column and the row spaces of A , respectively. In this paper we will mainly use the reduced SVD, which has all the information available in A . In many applications such as DNA analysis, image processing, or data analysis, one can compress the data by observing that there is a small number of *significant* singular values $\sigma_1 \geq \dots \geq \sigma_t > 0$, while other singular values $\sigma_{t+1}, \dots$, are much smaller than σ_t or equal to zero numerically. (We call t the numerical rank of A .) Then the *compressed* SVD of A will be given by

$$(1.2) \quad A_1 = U_1 \Sigma_1 V_1^T, \quad U_1 \in O_{mt}(\mathbb{R}), \quad V_1 \in O_{nt}(\mathbb{R}), \quad \Sigma_1 = \text{diag}(\sigma_1, \dots, \sigma_t) > 0.$$

*Received by the editors January 10, 2004; accepted for publication (in revised form) by R. Nabben March 9, 2005; published electronically November 14, 2005. This work was supported in part by the NSF.

<http://www.siam.org/journals/simax/27-2/43979.html>

†Department of Mathematics, Statistics and Computer Science, University of Illinois at Chicago, Chicago, IL 60607-7045 and Institute for Mathematics and Its Applications, 400 Lind Hall, 207 Church St., S.E., Minneapolis, MN 55455-0436 (friedlan@uic.edu).

A_1 also can be viewed as the *noise reduction* of A . This approach is used in [4] for finding the missing entries of the matrix A .

Let $A \in M_{mn}(\mathbb{R})$ and $B \in M_{ln}(\mathbb{R})$. Then the *generalized singular value decomposition* (GSVD) of A and B [5] is given by

$$(1.3) \quad A = F\Gamma R, \quad B = G\Delta R, \quad F \in O_m(\mathbb{R}), \quad G \in O_l(\mathbb{R}), \quad R \in GL(n, \mathbb{R}),$$

and $\Gamma \in M_{mn}(\mathbb{R})$, $\Delta \in M_{ln}(\mathbb{R})$ are diagonal matrices with the diagonal elements, called the generalized singular values, $\gamma_1, \dots, \gamma_{\min(m,n)} \geq 0$ and $\delta_1, \dots, \delta_{\min(l,n)} \geq 0$, respectively. (In general it is impossible to arrange both sets of the singular values in a decreasing order.) GSVD became important recently in DNA microarray analysis as a tool to compare two sets of DNA microarrays of different organisms [2] and [3]. (See section 2 for more details.)

The numerical difficulties in obtaining a stable GSVD decomposition can be partially attributed to the fact that the nonzero pair $(\gamma_i, \delta_i) \neq (0, 0)$ is determined up to a multiple by a positive scalar [5], [8], and [9]. The main feature of the existing algorithms for GSVD is the observation that the eigenvalues of the pencil $\delta^2 A^T A - \gamma^2 B^T B$ are the generalized singular values of the pair A, B .

The aim of this paper is to give a new robust algorithm to compute the *reduced* GSVD of A and B . The main feature of our approach is to consider the eigenvalues of the pencil $A^T A - \phi^2(A^T A + B^T B)$. Furthermore, to reduce the “noise” in the data represented by A, B we have to replace $P = A^T A + B^T B$ by a lower rank matrix $\tilde{P}$, using the *significant* singular values of P . Once $\tilde{P}$ is chosen, the low rank approximations $\tilde{A}, \tilde{B}$ of A, B , such that $\tilde{A}^T \tilde{A} + \tilde{B}^T \tilde{B} = \tilde{P}$, are determined. Thus our algorithm computes GSVD of the pencil $\tilde{A}^T \tilde{A} - \tilde{\phi}^2 \tilde{P}$. In applications such as microarray analysis, the choice of $\tilde{P}$ is left to the judgment of the user.

We now briefly survey the contents of our paper. In section 2 we discuss our main algorithm for GSVD. It is given in terms of matrices and is self-contained. In section 3 we discuss two numerical algorithms used to obtain stable GSVDs of A, B . In section 4 we discuss a random example of $A_0 \in M_{8,7}(\mathbb{R})$, $B_0 \in M_{9,7}(\mathbb{R})$ both of rank 2, such that the intersection of the row space of A_0 and B_0 is a subspace of dimension 1. Then the rank of $P_0 = A_0^T A_0 + B_0^T B_0$ is 3, and A_0, B_0 have three nonzero pairs of singular values: $(1.0, 0.0)$, $(0.681, 0.732)$, $(0.0, 1.0)$, up to three significant digits. In the terminology of microarrays we deduce that the two different organisms have three distinct functions, with exactly one common function.

Next we consider the random perturbation of A_0, B_0 given by matrices A, B . We replace this perturbation by A_1, B_1 of rank 2 using the SVD of A_1, B_1 . The matrix $P := A_1^T A_1 + B_1^T B_1$ has three large singular values of magnitude 10^8 , the fourth singular value is of order 10^4 , and the other three singular values are less than 10^{-2} . (We used floating point precision rounded off to 10 digits.) We first assumed that P has $r = 3$ significant singular values. Then GSVD of the appropriate approximations $\tilde{A}, \tilde{B}$ of rank 2 of A_0, B_0 is reasonably close to the original GSVD of A_0, B_0 . Next we assume that P has $r = 4$ significant singular values, the maximal possible number of nonzero singular values. Then the four pairs of generalized singular values of A_1, B_1 are $(1, 0)$, $(1, 0)$, $(0, 1)$, $(0, 1)$; i.e., the pair $(0.681, 0.732)$ of generalized singular values of A_0, B_0 split into the two pairs $(1, 0)$, $(0, 1)$ of generalized singular values of A_1, B_1 . In this case the microarray interpretation yields that the two organisms each have two distinct functions and no function in common! We explain the reason for this phenomenon in section 3.

In the appendix we give a short summary of the SVD for a linear operator that maps one finite dimensional inner product space to another finite dimensional inner

product space, over the complex numbers $\mathbb{C}$. This approach can be considered as a *base free* approach to the SVD. For one matrix $A \in M_{mn}(\mathbb{C})$ it yields an extended singular value decomposition, called ESVD, obtained by introducing any two inner products on the linear space $\mathbb{C}^m$ and $\mathbb{C}^n$. Then the GSVD of $A \in M_{mn}(\mathbb{C}), B \in M_{ln}(\mathbb{C})$ described in section 2 is the ESVD of A and B obtained by choosing a special inner product on $\mathbb{C}^n$ and the standard inner products on $\mathbb{C}^m$ and $\mathbb{C}^l$. We hope that the ESVD introduced here will have more applications in the near future.

2. An exact algorithm for GSVD. We now briefly describe the main steps in our algorithm for the reduced GSVD for a given pair $A \in M_{mn}(\mathbb{R}), B \in M_{ln}(\mathbb{R})$. (In this section we assume that A, B have no “noise.”) First we compute the three symmetric nonnegative definite matrices

$$(2.1) \quad P_A := A^T A, \quad P_B := B^T B, \quad P := P_A + P_B \in S_n(\mathbb{R}).$$

Assume that the reduced SVD of P , having rank r , is

$$(2.2) \quad P = O\Omega^2 O^T, \quad O \in O_{nr}(\mathbb{R}), \quad \Omega = \text{diag}(\omega_1, \dots, \omega_r), \quad \omega_1 \geq \dots \geq \omega_r > 0.$$

(The k th column of O is an eigenvector of P corresponding to the eigenvalue ω_k^2 for $k = 1, \dots, r$.) Then

$$(2.3) \quad Q_A := \Omega^{-1} O^T P_A O \Omega^{-1}, \quad Q_B := \Omega^{-1} O^T P_B O \Omega^{-1} \in S_r(\mathbb{R})$$

are nonnegative definite, and $Q_A + Q_B$ is the $r \times r$ identity matrix I_r . The spectral decompositions of Q_A and Q_B are given by

$$(2.4) \quad Q_A = T\Phi^2 T^T, \quad \Phi = \text{diag}(\phi_1, \dots, \phi_r), \quad \phi_i \geq 0, \quad i = 1, \dots, r, \quad T \in O_r(\mathbb{R}),$$

$$(2.5) \quad Q_B = T\Psi^2 T^T, \quad \Psi = \text{diag}(\psi_1, \dots, \psi_r), \quad \psi_i \geq 0, \quad i = 1, \dots, r,$$

$$(2.6) \quad \phi_i^2 + \psi_i^2 = 1, \quad \phi_i \geq 0, \psi_i \geq 0, \quad i = 1, \dots, r.$$

Let

$$(2.7) \quad V = O\Omega T \in M_{nr}(\mathbb{R}).$$

Then the GSVD of A and B is given by

$$(2.8) \quad A = U\Phi V^T, \quad U \in O_{mr}(\mathbb{R}), \quad B = W\Psi V^T, \quad W \in O_{lr}(\mathbb{R}).$$

The matrices U and W are easily obtained from the equalities

$$(2.9) \quad U\Phi = A O \Omega^{-1} T, \quad W\Psi = B O \Omega^{-1} T.$$

THEOREM 2.1. *Let $A \in M_{mn}(\mathbb{R}), B \in M_{ln}(\mathbb{R})$ and assume that $r = \text{rank}(A^T A + B^T B)$. Then there exists a GSVD given by (2.8), where $\Phi, \Psi \in M_r(\mathbb{R})$ are nonnegative diagonal matrices satisfying $\Phi^2 + \Psi^2 = I_r$ and $V \in M_{nr}(\mathbb{R}), \text{rank } V = r$. Every such GSVD can be found as follows: Let P be defined by (2.1); let O and Ω be given by the reduced SVD of P (2.2); let Q_A and Q_B be defined in (2.3); let the spectral decomposition of Q_A and Q_B be given by (2.4)–(2.6); and let V be given by (2.7). The columns of U and W in (2.8) corresponding to the nonzero ϕ_i and ψ_j are uniquely determined by (2.9). Other columns of U and W are any orthonormal systems in the orthogonal complement of the subspaces spanned by the determined columns of U and W , respectively.*

Proof. Clearly the matrices P_A , P_B , and P are nonnegative definite. The reduced SVD of P is the spectral decomposition of P corresponding to the positive eigenvalues of P . Observe

$$\Omega^2 = O^T P O = O^T (P_A + P_B) O = O^T P_A O + O^T P_B O.$$

Hence

$$(2.10) \quad I_r = Q_A + Q_B, \quad Q_A \geq 0, \quad Q_B \geq 0.$$

Let (2.4) be the spectral decomposition of Q_A . Then (2.10) implies (2.5) and (2.6). Let $\hat{U} := A O \Omega^{-1} T$. Then $\hat{U}^T \hat{U} = T^T Q_A T = \Phi^2$. Assume for simplicity of the exposition that

$$(2.11) \quad \phi_1 \geq \dots \geq \phi_{r_A} > 0 = \phi_{r_A+1} = \dots = \phi_r = 0,$$

where $r_A = \text{rank } A$. Then the last $r - r_A$ columns of $\hat{U}$ are zero, while the first r_A columns of $\hat{U}$ form an orthogonal system, where the norm of column i is ϕ_i for $i = 1, \dots, r_A$. We define $U \in M_{nr}(\mathbb{R})$ as follows. The i th column of U is the i th column of $\hat{U}$ divided by ϕ_i for $i = 1, \dots, r_A$. The last $r - r_A$ columns of U can be any orthonormal system in the orthogonal complement of the column space of $\hat{U}$. Hence $U \in O_{nr}(\mathbb{R})$, $\hat{U} = U \Phi$, and $A = U \Phi V^T$. The second equality in (2.8) is established similarly.

We now show that any GSVD given (2.8), with the conditions that $\Phi, \Psi \in M_r(\mathbb{R})$ are nonnegative diagonal matrices satisfying $\Phi^2 + \Psi^2 = I_r$ and $V \in M_{nr}(\mathbb{R})$, $\text{rank } V = r$, can be obtained by using the algorithm given in the theorem. Clearly $P = V V^T$. So the column space of V is equal to the column space of O . Since $\text{rank } V = \text{rank } O = r$, it follows that $V = O \Omega T$ for some invertible $T \in M_r(\mathbb{R})$. Use (2.2) and the equality $P = V V^T$ to deduce that $T T^T = I_r$. Hence $Q_A = T \Phi^2 T^T$, $Q_B = T \Psi^2 T^T$, and the rest of the theorem follows. $\square$

We conclude this section with the following remarks. First, we consider the interpretation of GSVD in comparing two sets of DNA microarrays of different organisms as discussed in [2]. As mentioned in section 1, a_{ij}, b_{pj} entries of $A = (a_{ij})_{i,j=1}^{m,n}$, $B = (b_{pj})_{p,j=1}^{l,n}$ represent the measurements for the i and p genomes of the two different organisms, represented by the matrices A and B , in the measurement $j = 1, \dots, n$. (We assume for the simplicity of the exposition that A, B do not have errors (noise). We address the problem of noise filtering of A, B in the next section.) The number r appearing in the GSVD (2.8) is the number of total functions of the DNA of the two organisms observed in the n experiments. If $\frac{\phi_i}{\psi_i} \approx 1$, then the function i is similarly expressed in both organisms. If $\frac{\phi_i}{\psi_i} \gg 1$ (resp., $\frac{\psi_i}{\phi_i} \gg 1$), then the function i is expressed only in the first organism (resp., the second organism).

Second, from the proof of Theorem 2.1 it follows that the information given in (2.8) is equivalent to the following:

$$(2.12) \quad \begin{aligned} A &= U_1 \Phi_1 V_1^T, \quad U_1 \in O_{mr_A}(\mathbb{R}), \quad V_1 \in O_{nr_A}(\mathbb{R}), \quad 0 < \Phi_1 = \text{diag}(\phi_{i_1}, \dots, \phi_{i_{r_A}}) \in S_{r_A}(\mathbb{R}), \\ B &= W_1 \Psi_1 V_2^T, \quad W_1 \in O_{mr_B}(\mathbb{R}), \quad V_2 \in O_{nr_B}(\mathbb{R}), \quad 0 < \Psi_1 = \text{diag}(\psi_{j_1}, \dots, \psi_{j_{r_B}}) \in S_{r_B}(\mathbb{R}). \end{aligned}$$

Here $\phi_{i_1}, \dots, \phi_{i_{r_A}}$ and $\psi_{j_1}, \dots, \psi_{j_{r_B}}$ are all positive entries of Φ and Ψ appearing in (2.8). (Note that $r_A = \text{rank } A$, $r_B = \text{rank } B$.) V_1 and V_2 are the submatrices of V

corresponding to the columns $i_1, \dots, i_{r_A}$ and $j_1, \dots, j_{r_B}$, respectively. U_1 and W_1 are the submatrices of U and W corresponding to the columns $i_1, \dots, i_{r_A}$ and $j_1, \dots, j_{r_B}$. Equivalently, (2.12) can be viewed as the reduced ESVD of A and B with respect to the standard inner products in $\mathbb{R}^m$ and $\mathbb{R}^l$ and the inner product in $\mathbf{V}$, the subspace spanned by the rows of A and B , induced by P . See section 6.

3. Two numerical algorithms. In this section we point out two numerical algorithms to compute GSVD for $A \in M_{mn}(\mathbb{R})$, $B \in M_{ln}(\mathbb{R})$. In the first algorithm we replace A, B by A_1, B_1 using the significant singular values of A, B of their corresponding SVD as explained in section 1. (A_1, B_1 are the compressed SVD of A, B .) Then we form the matrices

$$(3.1) \quad P_A := A_1^T A_1, \quad P := A_1^T A_1 + B_1^T B_1.$$

Next we consider the SVD of P , i.e., its spectral decomposition, and assume that P has $\tilde{r}$ *significant* singular values $\tilde{r} \leq \text{rank } P$. Let $\tilde{P}$ be the *compressed* SVD of P :

$$(3.2) \quad \tilde{P} = \tilde{O} \tilde{\Omega}^2 \tilde{O}^T, \quad \tilde{O} \in O_{n\tilde{r}}(\mathbb{R}), \quad \tilde{\Omega} = \text{diag}(\tilde{\omega}_1, \dots, \tilde{\omega}_{\tilde{r}}), \quad \tilde{\omega}_1 \geq \dots \geq \tilde{\omega}_{\tilde{r}} > 0.$$

(The k th column of $\tilde{O}$ is an eigenvector of P corresponding to the eigenvalue $\tilde{\omega}_k^2$ for $k = 1, \dots, \tilde{r}$.) Form $\tilde{Q}_A$ and compute its spectral decomposition, which determines $\tilde{\Phi}$ and $\tilde{\Psi}$:

$$(3.3) \quad \tilde{Q}_A := \tilde{\Omega}^{-1} \tilde{O}^T P_A \tilde{O} \tilde{\Omega}^{-1} \in S_{\tilde{r}}(\mathbb{R}),$$

$$(3.4) \quad \tilde{Q}_A = \tilde{T} \tilde{\Phi}^2 \tilde{T}^T, \quad \tilde{T} \in O_{\tilde{r}}(\mathbb{R}), \quad \tilde{\Phi} = \text{diag}(\tilde{\phi}_1, \dots, \tilde{\phi}_{\tilde{r}}), \quad \tilde{\phi}_1 \geq \dots \geq \tilde{\phi}_{\tilde{r}} \geq 0,$$

$$(3.5) \quad \tilde{\Psi} = \text{diag}(\tilde{\psi}_1, \dots, \tilde{\psi}_{\tilde{r}}), \quad 0 \leq \tilde{\psi}_1 \leq \dots \leq \tilde{\psi}_{\tilde{r}}, \quad \tilde{\Phi}^2 + \tilde{\Psi}^2 = I_{\tilde{r}}.$$

Then the first rank $\tilde{\Phi}$ columns of $\tilde{U}$ and the last rank $\tilde{\Psi}$ columns of $\tilde{W}$ are effectively determined by the equalities

$$(3.6) \quad \tilde{U} \tilde{\Phi} := A_1 \tilde{O} \tilde{\Omega}^{-1} \tilde{T}, \quad \tilde{U} \in O_{m\tilde{r}}(\mathbb{R}), \quad \tilde{W} \tilde{\Psi} := B_1 \tilde{O} \tilde{\Omega}^{-1} \tilde{T}, \quad \tilde{W} \in O_{l\tilde{r}}(\mathbb{R}).$$

We now explain why we need the additional noise reduction in P , after we already performed the noise reduction in A, B . Assume that the ideal data was given by $A_0 \in M_{mn}(\mathbb{R})$, $B_0 \in M_{ln}(\mathbb{R})$. Suppose that $\dim A_0^T \mathbb{R}^m \cap B_0^T \mathbb{R}^l = c > 0$. Let $P_0 := A_0^T A_0 + B_0^T B_0$. Then $\text{rank } P_0 = \text{rank } A_0 + \text{rank } B_0 - c$. Assume the interesting case $\text{rank } P_0 < n$.

The given matrices A, B are perturbations of A_0, B_0 . Suppose that $\text{rank } A_1 = \text{rank } A_0$, $\text{rank } B_1 = \text{rank } B_0$. However, as we shall see in the next section, with high probability $\text{rank } P = \min(\text{rank } A_0 + \text{rank } B_0, n) > \text{rank } P_0$. Still, P will have rank P_0 significant singular values. Thus $\tilde{P}$ is the correct approximation of P_0 , and $\tilde{A}, \tilde{B}$ are the approximation of A_0, B_0 with the property $\dim \tilde{A}^T \mathbb{R}^m \cap \tilde{B}^T \mathbb{R}^l = c$.

In the second algorithm we simply let $A_1 = A$ and $B_1 = B$; i.e., we do not perform the noise reduction in A, B . We just perform the noise reduction in P . Then $\tilde{A}, \tilde{B}$ are the noise reductions of A, B with the properties as above.

4. Numerical examples. One way to obtain a random matrix $E \in M_{pq}(\mathbb{R})$ of rank $t \leq \min(p, q)$, in general, is to use the following procedure:

$$(4.1) \quad E := \sum_{i=1}^t \mathbf{x}_i \mathbf{y}_i^T, \quad \mathbf{x}_1, \dots, \mathbf{x}_t \in \mathbb{R}^p, \quad \mathbf{y}_1, \dots, \mathbf{y}_t \in \mathbb{R}^q,$$

where $\mathbf{x}_1, \dots, \mathbf{x}_t, \mathbf{y}_1, \dots, \mathbf{y}_t$ are chosen at random. Suppose we use the procedure (4.1) to generate random matrix $A_0 \in M_{mn}(\mathbb{R})$ with $t := a_0 \leq \min(m, n)$ and random matrix $B_0 \in M_{ln}(\mathbb{R})$ with $t := b_0 \leq \min(l, n)$. Then generically $r_{A_0} = a_0$ and $r_{B_0} = b_0$, respectively. Assume that $a_0 + b_0 \leq n$. Then almost surely P_{A_0} and P_{B_0} do not have a common nullspace, and the rank of the matrix $P = A_0^T A_0 + B_0^T B_0$ is $r = r_{A_0} + r_{B_0}$. In order to generate random matrices $A_0 \in M_{mn}(\mathbb{R}), B_0 \in M_{ln}(\mathbb{R})$ of ranks r_{A_0}, r_{B_0} , such that the matrix P will have in general rank $r \in [\max(r_{A_0}, r_{B_0}), r_{A_0} + r_{B_0}]$, we modify procedure (4.1) as follows. Given $rc \geq 1$ we generate two random matrices $F \in M_n(\mathbb{R}), G \in M_{ln}(\mathbb{R})$ both of rank rc such that $F^T \mathbb{R}^m \cap G^T \mathbb{R}^l$ have, in general, dimension rc :

$$F := \sum_{i=1}^{rc} \mathbf{x}_i \mathbf{y}_i^T, G := \sum_{i=1}^{rc} \mathbf{z}_i \mathbf{y}_i^T, \mathbf{x}_1, \dots, \mathbf{x}_{rc} \in \mathbb{R}^m, \mathbf{y}_1, \dots, \mathbf{y}_{rc} \in \mathbb{R}^n, \mathbf{z}_1, \dots, \mathbf{z}_{rc} \in \mathbb{R}^l, \quad (4.2)$$

where $\mathbf{x}_1, \dots, \mathbf{x}_{rc}, \mathbf{y}_1, \dots, \mathbf{y}_{rc}, \mathbf{z}_1, \dots, \mathbf{z}_{rc}$ are chosen at random. After that we generate the random matrices $E_1 \in M_{mn}(\mathbb{R}), E_2 \in M_{ln}(\mathbb{R})$ of ranks $r_{A_0} - rc, r_{B_0} - rc$ using procedure (4.1). Then $A_0 = F + E_1, B_0 = G + E_2$ are of the ranks r_{A_0}, r_{B_0} and rank $P = r_{A_0} + r_{B_0} - rc$. (If $r_c = 0$, then $F = 0, G = 0$.)

We first generated $A_0 \in M_{8,7}(\mathbb{R}), B_0 \in M_{9,7}(\mathbb{R})$ with $rc = 1, r_{A_0} = r_{B_0} = 2$ as explained above:

$$A_0 = \begin{pmatrix} 1826 & 846 & 1516 & 1831 & 3060 & -577 & 1368 \\ -3452 & -1752 & -2182 & -2827 & -5970 & 1199 & -2236 \\ 5765 & 3573 & 745 & 2032 & 10755 & -2461 & 2250 \\ -202 & -1818 & 7558 & 6964 & -2430 & 1286 & 3804 \\ 3873 & 1353 & 5193 & 5718 & 5955 & -911 & 3914 \\ -5206 & -2862 & -2306 & -3350 & -9270 & 1964 & -2868 \\ -2060 & 1224 & -11470 & -11119 & -810 & -893 & -6540 \\ -2630 & -726 & -4390 & -4684 & -3810 & 482 & -3100 \end{pmatrix},$$

$$B_0 = \begin{pmatrix} -3652 & -3486 & 640 & 2833 & -321 & 1424 & -1731 \\ -8657 & -7471 & -2665 & 3283 & 1354 & 2669 & -6371 \\ 2420 & 2122 & 568 & -1063 & -289 & -776 & 1685 \\ -3927 & -4161 & 2865 & 4833 & -1446 & 1899 & -681 \\ 253 & -873 & 5837 & 4631 & -2952 & 895 & 3309 \\ -4620 & -2044 & -11676 & -6664 & 5908 & -308 & -8960 \\ 2596 & 2388 & 20 & -1624 & -12 & -932 & 1488 \\ -8624 & -7722 & -1180 & 4481 & 603 & 2908 & -5547 \\ -7964 & -5438 & -10024 & -3195 & 5075 & 1176 & -9967 \end{pmatrix}.$$

(Since we used Maple routine to generate random vectors and matrices with integer entries in the range $[-99, 99]$, the matrices A_0 and B_0 have integer entries.) The first three singular values of A_0 and B_0 are

$$27455.5092631633888, \quad 17374.6830503566089, \quad 3.14050409246786192 \times 10^{-12},$$

$$29977.5429571960522, \quad 19134.3838220483449, \quad 3.52429226420727071 \times 10^{-12},$$

i.e., the ranks of A_0 and B_0 are 2. The first four singular values of P are

$$\begin{aligned} 1.32179857269680762 \times 10^9, & \quad 6.04366385186753988 \times 10^8, \\ 3.94297368116438210 \times 10^8, & \quad 1.34609524647135614 \times 10^{-7}. \end{aligned}$$

The three generalized singular values of A_0 and B_0 given by Theorem 2.1 are

$$\begin{aligned} \phi_1 = 1, \quad \phi_2 = 0.6814262563, \quad \phi_3 = 3.777588180 \times 10^{-9}, \\ \psi_1 = 0, \quad \psi_2 = 0.7318867789, \quad \psi_3 = 1. \end{aligned}$$

The matrix $V \in M_{7,3}(\mathbb{R})$ given by Theorem 2.1 is

$$V = \begin{pmatrix} 9975.15570726070292 & 2218.77772608917530 & -16518.7090029761894 \\ 4258.08009519515508 & 5446.09078420315382 & -13181.0851653484933 \\ 9910.16789114064704 & -17951.9288836113556 & -10755.8389196556800 \\ 11513.1007966750912 & -16136.5652888949844 & 1610.05490646888757 \\ 16275.4438487944208 & 9076.81797277147780 & 5451.31112120093894 \\ -2894.44180376545456 & -3832.43425556036664 & 4134.75009336168569 \\ 8306.91358787289937 & -8471.69677689182754 & -15231.5628696020595 \end{pmatrix}.$$

U_1 , the first two columns of $U \in M_{8,3}(\mathbb{R})$, and W_1 , the last two columns of $W \in M_{9,3}(\mathbb{R})$ in the decomposition (2.8), are given by

$$U_1 = \begin{pmatrix} 0.179765121701646796 & 0.0217039456257290890 \\ -0.332293103754918662 & -0.0908271155631820566 \\ 0.522953081333910608 & 0.362755122878894332 \\ 0.0653691351259460541 & -0.564884208286237532 \\ 0.403109666836316383 & -0.0979419423157340264 \\ -0.490268513741756395 & -0.208670062565787018 \\ -0.310503391981747257 & 0.686088211972394224 \\ -0.283266252367416982 & 0.129387295795073296 \end{pmatrix},$$

$$W_1 = \begin{pmatrix} -0.212587926393571491 & 0.200183908075783706 \\ -0.209341218852211240 & 0.503492859705758855 \\ 0.0709866110946431456 & -0.139522117749788622 \\ -0.381933946402143232 & 0.200183908075783734 \\ -0.399563630497640044 & -0.0545956112933954560 \\ 0.610572448157044589 & 0.339706025825572078 \\ 0.117661178466181738 & -0.145588296782388132 \\ -0.312433467319077507 & 0.491360501640559999 \\ 0.340824522437284283 & 0.515625217770957822 \end{pmatrix}.$$

To show the robustness of our GSVD we perturb the matrices A_0 and B_0 by random matrices of the maximal rank by generating matrices $X \in M_{8,7}(\mathbb{R})$, $Y \in M_{9,7}(\mathbb{R})$ with random entries and relatively small ℓ_2 norm (the first singular value) with respect to the ℓ_2 norms of A and B , respectively. The random matrices X, Y , with integer

entries in $[-99, 99]$, are

$$X = \begin{pmatrix} -14 & -73 & 65 & 3 & -14 & 16 & 9 \\ -10 & 8 & 90 & -94 & -22 & -24 & 0 \\ 78 & 32 & -48 & -6 & 80 & -18 & -63 \\ 66 & -13 & 88 & -45 & -92 & -69 & -43 \\ 32 & 9 & 41 & -95 & -28 & -90 & -63 \\ -23 & -72 & -84 & -84 & -58 & -37 & -40 \\ 35 & 14 & -29 & 76 & -62 & -82 & -5 \\ 18 & -40 & -51 & 11 & 87 & 66 & -46 \end{pmatrix},$$

$$Y = \begin{pmatrix} -14 & -83 & 65 & -22 & -80 & 43 & -56 \\ -55 & -50 & 68 & 66 & 9 & -26 & -58 \\ -63 & -32 & -25 & 96 & 90 & -5 & 28 \\ -49 & 31 & -17 & -27 & 46 & -5 & 37 \\ 81 & -99 & -98 & 22 & 58 & -68 & -37 \\ 59 & 57 & 65 & -64 & 65 & 84 & 41 \\ -68 & 36 & -63 & 7 & -58 & 53 & 90 \\ 95 & -27 & 54 & -16 & -46 & -18 & 46 \\ 23 & 10 & -64 & 58 & 58 & -73 & 97 \end{pmatrix}.$$

The singular values of X, Y rounded off to three significant digits are

$$(266, 183, 165, 151, 99.1, 36.0, 14.1), (259, 229, 198, 153, 116, 86.8, 46.2).$$

Note that $\|X\| \sim 0.01\|A_0\|$, $\|Y\| \sim 0.01\|B_0\|$. Form the matrices $A := A_0 + X$, $B := B_0 + Y$. These matrices have the full ranks with corresponding singular values rounded off to at least three significant digits:

$$(27490, 17450, 233, 130, 119, 70.0, 18.2), (29884, 19183, 250, 187, 137, 102, 19.7).$$

We now replace A, B by A_1, B_1 of rank 2 using the first two singular values and the corresponding singular vectors in the SVDs of A, B . (We do the noise reduction described in the introduction.) Then two nonzero singular values of A_1, B_1 are $(27490, 17450), (29883, 19183)$, rounded to five significant digits. The singular values of the corresponding $P = A_1^T A_1 + B_1^T B_1$ are up to three significant digits:

$$(4.3) \quad (1.32 \times 10^9, 6.07 \times 10^8, 3.96 \times 10^8, 1.31 \times 10^4, 0.068, 9.88 \times 10^{-3}, 6.76 \times 10^{-3}).$$

Assume that $\tilde{r} = 3$; i.e., P has three significant singular values. We now apply our first numerical algorithm for GSVD, as described in section 3. The three generalized singular values of A_1, B_1 are

$$(1.000000000, 0.6814704276, 0.7582758358 \times 10^{-8}), \quad (0., 0.7318456506, 1.0).$$

These results match the generalized singular values of A_0, B_0 up to at least four significant digits. Let $\hat{V}, \hat{U}_1, \hat{W}_1$ be the matrix V , the first two columns of U , the last two columns of W , which are computed for A_1, B_1 . Then

$$\frac{\|V - \hat{V}\|}{\|V\|} \sim 0.0061, \quad \|U_1 - \hat{U}_1\| \sim 0.0093, \quad \|W_1 - \hat{W}_1\| \sim 0.0098.$$

We now apply the second numerical algorithm to the matrices $A \in M_{8,7}, B \in M_{9,7}(\mathbb{R})$, as described in section 3. The singular values of $P = A^T A + B^T B$, up to three significant digits, are

$$(1.31 \times 10^9, 6.07 \times 10^8, 3.96 \times 10^8, 6.91 \times 10^4, 6.61 \times 10^4, 2.84 \times 10^4, 1.19 \times 10^4).$$

As expected, P has $\tilde{r} = 3$ significant singular values. Then the three generalized singular eigenvalues of $\tilde{A}, \tilde{B}$ are

$$(0.9999667639, 0.6814699415, 0.005726580138), \\ (0.008152974917, 0.7318461033, 0.9999836030).$$

Let $\tilde{V}, \tilde{U}_1, \tilde{W}_1$ be the corresponding matrices for GSVD of $\tilde{A}, \tilde{B}$. Then

$$\frac{\|V - \tilde{V}\|}{\|V\|} \sim 0.0061, \quad \|U_1 - \tilde{U}_1\| \sim 0.0091, \quad \|W_1 - \tilde{W}_1\| \sim 0.011.$$

This shows that the two “noise reduction” algorithms are comparable in these examples.

Finally we discuss the critical issue of correctly choosing the number of significant singular values of noised matrices A, B and the corresponding matrix P . First we revisit our example with A_1, B_1 . Recall the values of the singular values of $P = A_1^T A_1 + B_1^T B_1$ given by (4.3). Assume now that $\tilde{r} = 4$. Then the four generalized singular values of $\tilde{A}, \tilde{B}$ up to six significant digits are $(1, 1, 0, 0), (0, 0, 1, 1)!$

We now replace A, B by A_2, B_2 of rank 3 using the first three singular values and the corresponding singular vectors in the SVD decompositions of A, B . The singular values of the matrix P up to three significant digits are

$$(1.32 \times 10^9, 6.07 \times 10^8, 3.96 \times 10^8, 4.74 \times 10^4, 3.83 \times 10^4, 5.39 \times 10^3, 9.70 \times 10^{-3}).$$

Assume first that $\tilde{r} = 6$. It then turns out that the generalized singular values of A_2, B_2 up to six significant digits are $(1, 1, 1, 0, 0, 0)$ and $(0, 0, 0, 1, 1, 1)!$ Assume finally in this example that $\tilde{r} = 3$. It then turns out that the generalized singular values of $\tilde{A}, \tilde{B}$ are

$$(0.9999796224, 0.6814701987, 0.005232470265), \\ (0.006383948621, 0.7318458638, 0.9999863106).$$

Thus, the most important step in finding the GSVD of noised data given by $\tilde{A}, \tilde{B}$ is the choice of $\tilde{r}$, the significant number of singular values of P .

5. Conclusions. In this paper we discuss a new algorithm to find the GSVD of two real matrices A, B with the same number of rows. The main novelty of our approach is to consider the spectral decomposition of the nonnegative symmetric matrix $P := A^T A + B^T B$ and use it to derive the GSVD of the pair A, B . Roughly speaking, the GSVD is closely related to the spectral decomposition of the pencils $A^T A - \phi^2 P, B^T B - \psi^2 P$.

To obtain stable numerical computations of GSVD of A, B one needs to reduce the “noise” in P by approximating P with a low rank matrix $\tilde{P}$, which is called the *numerical* rank of P . The numerical examples clearly showed that the stability of the GSVD is greatly influenced by the choice of the numerical rank of P . This may have a significant impact on the interpretation of the comparative analysis of genome-scale expression data sets of two different organisms.

6. Appendix: SVD on inner product spaces. In this appendix we briefly discuss the standard notion of the SVD of a linear operator that maps one finite dimensional inner product space to another finite dimensional inner product space. For the utmost generality we consider here the inner products over the complex numbers $\mathbb{C}$. The proofs of the facts stated here are either standard or straightforward, and are left to the reader.

Let $\mathbf{U}_i$ be an m_i -dimensional inner product space over $\mathbb{C}$, given by $\langle \cdot, \cdot \rangle_i$ for $i = 1, 2$. Let $T : \mathbf{U}_1 \rightarrow \mathbf{U}_2$ be a linear operator. Let $T^* : \mathbf{U}_2 \rightarrow \mathbf{U}_1$ be the adjoint operator of T , i.e., $\langle T\mathbf{x}, \mathbf{y} \rangle = \langle \mathbf{x}, T^*\mathbf{y} \rangle$ for all $\mathbf{x} \in \mathbf{U}_1, \mathbf{y} \in \mathbf{U}_2$. Equivalently, let $[\mathbf{a}_1, \dots, \mathbf{a}_{m_1}]$ and $[\mathbf{b}_1, \dots, \mathbf{b}_{m_2}]$ be orthonormal bases of $\mathbf{U}_1$ and $\mathbf{U}_2$, respectively. Let $A \in M_{m_2 m_1}(\mathbb{C})$ be the representation matrix of T in these bases: $[T\mathbf{a}_1, \dots, T\mathbf{a}_{m_1}] = [\mathbf{b}_1, \dots, \mathbf{b}_{m_2}]A$. Then the matrix $A^* := \overline{A}^T$ represents T^* in these bases. The SVD decomposition of $A = U\Sigma V^*$, where U, V are unitary matrices of dimensions m_2, m_1 , respectively, and $\Sigma \in M_{m_2 m_1}(\mathbb{R})$ is a diagonal matrix with the diagonal entries $\sigma_1 \geq \dots \geq \sigma_{\min(m_2, m_1)} \geq 0$, corresponds to the following *base free* concepts of T .

Consider the operators $S_1 := T^*T : \mathbf{U}_1 \rightarrow \mathbf{U}_1$ and $S_2 := TT^* : \mathbf{U}_2 \rightarrow \mathbf{U}_2$. Then S_1, S_2 are self-adjoint, i.e., $S_1^* = S_1, S_2^* = S_2$, and nonnegative definite, i.e., $\langle S_i \mathbf{x}_i, \mathbf{x}_i \rangle \geq 0$ for all $\mathbf{x}_i \in \mathbf{U}_i$ for $i = 1, 2$. The positive eigenvalues of S_1 and S_2 , counted with their multiplicities and arranged in a decreasing order, are $\sigma_1^2 \geq \dots \geq \sigma_r^2 > 0$, where $r = \text{rank } T = \text{rank } T^*$. Let

$$S_1 \mathbf{v}_i = \sigma_i^2 \mathbf{v}_i, \quad i = 1, \dots, r, \quad \langle \mathbf{v}_i, \mathbf{v}_j \rangle_1 = \delta_{ij}, \quad i = 1, \dots, r.$$

Then $\mathbf{u}_i := \sigma_i^{-1} T \mathbf{v}_i$ for $i = 1, \dots, r$ is an orthonormal set of the eigenvectors of S_2 corresponding to the eigenvalue σ_i^2 for $i = 1, \dots, r$. Complete the orthonormal systems $\{\mathbf{v}_1, \dots, \mathbf{v}_r\}$ and $\{\mathbf{u}_1, \dots, \mathbf{u}_r\}$ to orthonormal bases $[\mathbf{v}_1, \dots, \mathbf{v}_{m_1}]$ and $[\mathbf{u}_1, \dots, \mathbf{u}_{m_2}]$ in $\mathbf{U}_1$ and $\mathbf{U}_2$, respectively. Then the unitary matrices U, V are the transition matrices from basis $[\mathbf{b}_1, \dots, \mathbf{b}_{m_2}]$ to $[\mathbf{u}_1, \dots, \mathbf{u}_{m_2}]$ and basis $[\mathbf{a}_1, \dots, \mathbf{a}_{m_1}]$ to $[\mathbf{v}_1, \dots, \mathbf{v}_{m_1}]$:

$$[\mathbf{u}_1, \dots, \mathbf{u}_{m_2}] = [\mathbf{b}_1, \dots, \mathbf{b}_{m_2}]U, \quad [\mathbf{v}_1, \dots, \mathbf{v}_{m_1}] = [\mathbf{a}_1, \dots, \mathbf{a}_{m_1}]V.$$

Let $A \in M_{m_2 m_1}(\mathbb{C})$. Then A can be viewed as a linear operator $A : \mathbb{C}^{m_1} \rightarrow \mathbb{C}^{m_2}$, where $\mathbf{x} \rightarrow A\mathbf{x}$ for any $\mathbf{x} \in \mathbb{C}^{m_1}$. Let P_i be an $m_i \times m_i$ hermitian positive definite matrix for $i = 1, 2$. We define the following inner product on $\mathbb{C}^{m_1}$ and $\mathbb{C}^{m_2}$:

$$(6.1) \quad \langle \mathbf{x}, \mathbf{y} \rangle_i := \mathbf{y}^* P_i \mathbf{x}, \quad \mathbf{x}, \mathbf{y} \in \mathbb{C}^{m_i}, \quad i = 1, 2.$$

It is straightforward to show that the SVD of A , viewed as the above operator, with respect to the inner products given by (6.1) is

$$(6.2) \quad A = U\Sigma V^*, \quad U^* P_2 U = I_{m_2}, \quad V^* P_1^{-1} V = I_{m_1},$$

where Σ is an $m_2 \times m_1$ diagonal matrix with nonnegative diagonal entries in decreasing order. A simple way to deduce this decomposition is to observe that $\langle \mathbf{x}, \mathbf{y} \rangle_i = (P_i^{\frac{1}{2}} \mathbf{y})^* (P_i^{\frac{1}{2}} \mathbf{x})$, where $P_i^{\frac{1}{2}}$ is the unique hermitian positive definite square root of P_i . The decomposition (6.2) is called the ESVD of A . The diagonal entries of Σ are called the *extended singular values* of A . ESVD of A corresponds to the standard SVD of $P_2^{\frac{1}{2}} A P_1^{-\frac{1}{2}}$.

We conclude this section by considering the GSVD of $A \in M_{m_2 m_1}(\mathbb{C})$ and $B \in M_{m_3 m_1}(\mathbb{C})$ given by

$$(6.3) \quad A = U\Phi V^*, \quad U \in U_{m_2 r}(\mathbb{C}), \quad B = W\Psi V^*, \quad W \in U_{m_3 r}(\mathbb{C}), \quad V \in U_{m_1 r}(\mathbb{C}).$$

Here $U_{mr}(\mathbb{C}) := \{Z \in M_{mr}(\mathbb{C}) : Z^*Z = I_r\}$, and the matrices U, V, W, Φ, Ψ are given by the formulas as in section 2, except that the transposed matrices appearing there are replaced by the conjugate transposed matrices. We claim that (6.3) are the ESVD of A and B with respect to the following corresponding hermitian positive definite matrices $P_i \in M_{m_i}(\mathbb{C}), i = 1, 2, 3$. First let $P_2 = I_{m_2}$ and $P_3 = I_{m_3}$. Let $P = A^*A + B^*B$ be an $m_1 \times m_1$ hermitian nonnegative definite matrix. Assume first that $P > 0$. Then choose $P_1 = P$. Let $E := AP^{-\frac{1}{2}}, F := BP^{-\frac{1}{2}}$. Observe that $E^*E + F^*F = I_{m_1}$. Then the spectral decomposition of E^*E and F^*F is given by

$$E^*E = V_0\Phi^2V_0^*, F^*F = V_0\Psi^2V_0^*, \quad V_0^*V_0 = I_{m_1}, \quad \Phi^2 + \Psi^2 = I_{m_1},$$

where Φ, Ψ are diagonal nonnegative definite matrices. This establishes the decomposition (6.3) with $V^* = V_0^*P^{\frac{1}{2}}$.

Now assume that P is not positive definite; i.e., $\text{rank } P < m_1$. Let $\mathbf{V} \subset \mathbb{C}^{m_1}$ be the range of P of dimension r . Note that $AC^{m_1} = A\mathbf{V}$, $BC^{m_1} = B\mathbf{V}$. Hence we can view the matrices A, B as the operators $A : \mathbf{V} \rightarrow \mathbb{C}^{m_2}$. Let $\langle \mathbf{x}, \mathbf{y} \rangle := \mathbf{y}^*P\mathbf{x}$ be an inner product on $\mathbf{V}$. Then the ESVD of the operators A, B gives the GSVD (6.3). Alternatively, let P_1 be any $m_1 \times m_1$ hermitian matrix such that $\mathbf{V}$ is an invariant subspace of P_1 and the restriction of P_1 to $\mathbf{V}$ is equal to the restriction of P to $\mathbf{V}$. Let E, F be defined as above. Then $E^*E + F^*F$ is the identity operator on $\mathbf{V}$ and is equal to the zero operator on the orthogonal complement of $\mathbf{V}$ in $\mathbb{C}^{m_1}$. Hence E^*E and F^*F commute. Thus E^*E and F^*F have a common orthonormal basis of eigenvectors. Then the reduced ESVD of A and B yields the decomposition (6.3).

REFERENCES

- [1] O. ALTER, P. O. BROWN, AND D. BOTSTEIN, *Singular value decomposition for genome-wide expression data processing and modelling*, in Proc. Nat. Acad. Sci. USA, 97 (2000), pp. 10101–10106.
- [2] O. ALTER, P. O. BROWN, AND D. BOTSTEIN, *Generalized singular decomposition for comparative analysis of genome-scale expression data sets of two different organisms*, in Proc. Nat. Acad. Sci. USA, 100 (2003), pp. 3351–3356.
- [3] O. ALTER, G. H. GOLUB, P. O. BROWN, AND D. BOTSTEIN, *Novel genome-scale correlation between DNA replication and RNA transcription during the cell cycle in yeast is predicted by data-driven models*, in Proceedings of the Miami Nature Biotechnology Winter Symposium on the Cell Cycle, Chromosomes and Cancer, Vol. 15, M. P. Deutscher et al., eds., University of Miami School of Medicine, Miami, 2004.
- [4] S. FRIEDLAND, A. NIKNEJAD, AND L. CHIHARA, *A simultaneous reconstruction of missing data in DNA microarrays*, Linear Algebra Appl., to appear; available online at <http://www.ima.umn.edu/preprints/dec2003/dec2003.html>.
- [5] G. H. GOLUB AND C. F. VAN LOAN, *Matrix Computation*, 3rd ed., The Johns Hopkins University Press, Baltimore, MD, 1996.
- [6] R. A. HORN AND C. R. JOHNSON, *Matrix Analysis*, Cambridge University Press, Cambridge, UK, 1987.
- [7] R. A. JOHNSON AND D. W. WICHERN, *Applied Multivariate Statistical Analysis*, 4th ed., Prentice-Hall, Englewood Cliffs, NJ, 1998.
- [8] C. C. PAIGE AND M. A. SAUNDERS, *Towards a generalized singular value decomposition*, SIAM J. Numer. Anal., 18 (1981), pp. 398–405.
- [9] G. W. STEWART, *A method for computing the generalized singular value decomposition*, in Matrix Pencils, B. Kagström and A. Ruhe, eds., Lecture Notes in Math. 973, 1983, pp. 207–220.

ADDITIVE SCHWARZ ITERATIONS FOR MARKOV CHAINS*

RAFAEL BRU[†], FRANCISCO PEDROCHE[†], AND DANIEL B. SZYLD[‡]

Abstract. A convergence analysis is presented for additive Schwarz iterations when applied to consistent singular systems of equations of the form $Ax = b$. The theory applies to singular M -matrices with one-dimensional null space and is applicable in particular to systems representing ergodic Markov chains, and to certain discretizations of partial differential equations. Additive Schwarz can be seen as a generalization of block Jacobi, where the set of indices defining the diagonal blocks have nonempty intersection; this is called the overlap. The presence of overlap is known to accelerate the convergence of the methods in the nonsingular case. By providing convergence results, as well as some characteristics of the induced splitting, we hope to encourage the use of this additional computational tool for the solution of Markov chains and other singular systems. We present several numerical examples showing that additive Schwarz performs better than block Jacobi. For completeness, a few numerical experiments with block Gauss–Seidel and multiplicative Schwarz are also included.

Key words. Markov chains, singular linear systems, additive Schwarz iterations, overlap

AMS subject classifications. 65F10, 65F15, 15A48

DOI. 10.1137/040616541

1. Introduction. Stationary classical iterative methods such as block Jacobi and block Gauss–Seidel currently are extensively used for the numerical solution of Markov chains; see, e.g., [20]. Schwarz methods can be seen as generalizations of these methods as they introduce overlap between the blocks. It is the purpose of this paper to show theoretically and experimentally that additive Schwarz iterations can be a viable alternative.

Schwarz iterative methods were developed to solve linear systems of algebraic equations derived from discretizations of partial differential equations; see, e.g., [17], [18], [23]. Schwarz methods are usually used as preconditioners, though in many cases they have been used as stationary iterative methods associated with a physical domain decomposed in p overlapping subdomains, especially for nonsymmetric problems; see, e.g., [14]. In this paper we focus on additive Schwarz when used as a stationary iterative method for the solution of an $n \times n$ consistent singular linear system of the form

$$(1.1) \quad Ax = b,$$

where we assume that the null space of A is one-dimensional. This situation applies in particular to matrices $A = I - B$, where B is irreducible and column stochastic, representing an ergodic Markov chain, and $b = 0$. In this case, the solution of (1.1) corresponds to the stationary probability distribution of the Markov chain [20]. More

*Received by the editors October 7, 2004; accepted for publication (in revised form) by A. Frommer March 25, 2005; published electronically November 14, 2005.
<http://www.siam.org/journals/simax/27-2/61654.html>

[†]Institut de Matemàtica Multidisciplinar, Universitat Politècnica de València, Camí de Vera s/n. 46022 València, Spain (rbbru@mat.upv.es, pedroche@mat.upv.es). This work was supported by the Spanish DGI grants BFM2001-2641 and MTM2004-02998 and by the Oficina de Ciència i Tecnologia de la Presidència de La Generalitat Valenciana under project GRUPOS03/062.

[‡]Department of Mathematics, Temple University, Philadelphia, PA 19122-6094 (szyld@math.temple.edu). The work of this author was supported in part by the U.S. National Science Foundation under grant DMS-0207525.

generally, we assume that A is a singular M -matrix with each principal submatrix being nonsingular. Other situations, such as discretization of certain differential equations with Neumann boundary conditions, can also be represented with such matrices.

The convergence of Schwarz iterations in the framework of an algebraic formulation was studied in [2], [7], [10], where theorems on convergence were presented for the case of A being symmetric positive definite, nonsingular M -matrix, and H -matrix. Recently, the convergence of multiplicative Schwarz iterations for singular systems was proved [13]. In the present paper, we prove the convergence of additive Schwarz iterations for singular systems and study certain properties of the induced splitting. When no overlap is present, additive Schwarz reduces to the classical block Jacobi method, while multiplicative Schwarz reduces to block Gauss–Seidel. We review these concepts later and we give definitions and theorems relating those matrices describing the overlap. Overlap has played a major role in the convergence of Schwarz methods for discretizations of differential equations. Here we present numerical evidence that overlap can indeed improve the convergence of iterative methods for singular systems, and specifically for ergodic Markov chains.

We mention that the use of Krylov subspace methods for these problems has not turned out to be so competitive; see, e.g., [16].

The paper is structured as follows. In section 2 we review the algebraic formulation of additive Schwarz iterations and give some preliminaries, especially concerning the overlap. In section 3 we give conditions for the convergence of this iterative method when used with Markov chains. In section 4 we prove, for certain important cases, the existence of a splitting induced by the iteration matrix of additive Schwarz. In section 5 we show some numerical experiments comparing the convergence factor of additive Schwarz and block Jacobi. We also provide execution times. For completeness we include a few numerical experiments with block Gauss–Seidel and multiplicative Schwarz.

2. Overlapping blocks. Algebraic representation. To formally describe the overlapping blocks, let V_i be subspaces of $V = \mathbb{R}^n$ of dimension n_i , $i = 1, \dots, p$ ($p > 1$), such that the sum of these subspaces spans the whole space. These subspaces are not pairwise disjoint; on the contrary, their intersection is precisely the overlap, and thus $\sum_{i=1}^p n_i > n$.

The restriction and prolongation operators map vectors from V to V_i , and vice versa. The restriction operators used here are of the form

$$R_i = [I_i | O] \pi_i, \quad i = 1, \dots, p,$$

where I_i is the identity on $\mathbb{R}^{n_i}$, and π_i is a permutation matrix on $\mathbb{R}^n$. The prolongation operators considered here are R_i^T , $i = 1, \dots, p$. Define the diagonal matrices

$$(2.1) \quad E_i = R_i^T R_i, \quad i = 1, \dots, p,$$

which have nonzero diagonal elements (with value one) only in the columns of the matrix R_i which have a nonzero element. We denote by

$$(2.2) \quad S_i = \{i_1, i_2, \dots, i_{n_i}\}, \quad i_1 < i_2 < \dots < i_{n_i},$$

the set of indices corresponding to the columns of R_i which have nonzero elements, and by q the *measure of overlap*, i.e., the maximum over all possible rows, of the number of matrices E_i with a nonzero in the row. Thus

$$(2.3) \quad \sum_{i=1}^p E_i \leq qI,$$

and usually $q \ll p$.

The restriction of the matrix A to the subspace V_i is

$$(2.4) \quad A_i = R_i A R_i^T,$$

which is a symmetric permutation of an $n_i \times n_i$ principal submatrix of A , $i = 1, \dots, p$. We assume throughout the paper that the matrix A is such that A_i is nonsingular whenever $n_i < n$, $i = 1, \dots, p$. We note that the indices on S_i in (2.2) are those corresponding to the rows and columns of A that contribute to A_i , while the arrangement of these rows and columns is given by R_i (or, equivalently, by π_i). These matrices A_i are just the p overlapping blocks; see [2], [10] for more details.

With these definitions we can briefly describe the Schwarz iterations for the solution of (1.1); more detail can be found, e.g., in [2], [10]. The damped additive Schwarz iteration for the solution of (1.1) has the form

$$x^{k+1} = x^k + \theta \sum_{i=1}^p R_i^T A_i^{-1} R_i (b - A x^k),$$

where x^0 is an initial vector and $0 < \theta \leq 1$ is the *damping* factor. Thus, additive Schwarz iterations can be expressed as most classical methods by the iteration

$$(2.5) \quad x^{k+1} = T x^k + c, \quad k = 0, 1, \dots$$

The iteration matrix for this scheme is thus

$$(2.6) \quad T = T_\theta = I - \theta \sum_{i=1}^p R_i^T A_i^{-1} R_i A = I - \theta \sum_{i=1}^p P_i,$$

where $P_i = R_i^T A_i^{-1} R_i A$ is a projection onto V_i . The iteration matrix for the multiplicative Schwarz iterations is

$$T = T_\mu = (I - P_p)(I - P_{p-1}) \cdots (I - P_1) = \prod_{i=p}^1 (I - P_i).$$

Following [10], we use a different algebraic representation of the iteration matrix (2.6). To that end, let $A_{-i} = [O | I_{-i}] \pi_i A \pi_i^T [O | I_{-i}]^T$, where I_{-i} is the $(n - n_i) \times (n - n_i)$ identity matrix, and let

$$(2.7) \quad M_i = \pi_i^T \begin{bmatrix} A_i & O \\ O & \text{diag}(A_{-i}) \end{bmatrix} \pi_i.$$

We observe that this matrix M_i is just a copy of A with some coefficients changed to zero. In fact, we can express the (combinatorial) zero structure of M_i explicitly as

$$(2.8) \quad M_i(j, k) = 0 \quad \text{if } j \neq k \text{ and } j \notin S_i \text{ or } k \notin S_i$$

for $j, k = 1, \dots, n$. Of course there might be more zeros in the other locations corresponding to A_i , but we do not use this fact.

It follows then from the form of the matrices (2.1) and (2.7) that

$$(2.9) \quad G_i := E_i M_i^{-1} = R_i^T A_i^{-1} R_i = \pi_i^T \begin{bmatrix} A_i^{-1} & O \\ O & O \end{bmatrix} \pi_i.$$

Using this equality, the iteration matrix T_θ can be expressed as

$$(2.10) \quad T_\theta = I - \theta \sum_{i=1}^p E_i M_i^{-1} A.$$

Our proof of convergence of additive Schwarz iterations consists of showing that if $\theta < 1/q$, then the limit $\lim_{k \rightarrow \infty} T_\theta^k$ exists. We do this in section 3.

The existence of a splitting $A = M - N$ so that $T_\theta = M^{-1}N$ is based on showing that the matrix $G = \sum_{i=1}^p E_i M_i^{-1}$ is nonsingular. As we shall see, this issue is not trivial, and one needs to consider some restrictions on the sets S_i . We do this in section 4.

3. Convergence of additive Schwarz for Markov chains. Convergence and comparison theorems for the classical stationary iterations (2.5) in the case of A being a nonsingular matrix are well established, and the rate of convergence of such iterative methods is given by the spectral radius of T , $\rho(T)$; see, e.g., [4], [5], [24], [25]. For the present case where A is singular the rate of convergence is given by the *convergence factor* $\gamma(T)$ defined as

$$\gamma(T) = \max\{|\lambda|, \lambda \in \sigma(T), \lambda \neq 1\},$$

where $\sigma(T)$ denotes the spectrum of T . Comparison theorems for splittings of singular matrices have appeared in [11], [12].

We consider $A = I - B$, with B an $n \times n$ irreducible column stochastic matrix. We wish to show that $\lim_{k \rightarrow \infty} T_\theta^k$ exists, i.e., it is *convergent*¹. This limit if it exists is a projection onto the null space of the singular matrix A . In this case, the additive Schwarz iteration converges to a nontrivial solution of $Ax = 0$ for any initial vector $x^0 \notin \mathcal{R}(I - T)$, the range of $I - T$ [4], [5].

As is well known [4], [5], a matrix T with $\rho(T) = 1$ is convergent if and only if each of the following conditions hold:

- (i) $\lambda \in \sigma(T)$ with $|\lambda| = 1$ implies $\lambda = 1$; i.e., $\gamma(T) < 1$.
- (ii) $\text{ind}_1 T = 1$; i.e., $\text{rank}(I - T) = \text{rank}(I - T)^2$.

When $T \geq O$, condition (i) can be replaced with T having positive diagonal entries [1]. Equivalent conditions for (ii) can be found in [22]. Here we will use the following lemma, whose proof can be found, e.g., in [6].

LEMMA 3.1. *Let T be a nonnegative square matrix such that $Tv \leq v$ with $v > 0$. Then $\rho(T) \leq 1$. If, furthermore, $\rho(T) = 1$, then $\text{ind}_1 T = 1$.*

In our convergence proofs, we will show that the iteration matrix satisfies $T \geq O$ and that there is a positive vector v for which $Tv = v$, and thus conclude using this lemma that $\rho(T) = 1$ and $\text{ind}_1 T = 1$; i.e., (ii) holds. We will also show that the diagonals of T are positive, and thus (i) also holds.

We classify splittings $A = M - N$ as being of *nonnegative type* if the matrix $M^{-1}N$ is nonnegative [13], as *weak regular* if both M^{-1} and $M^{-1}N$ are nonnegative, and as *regular* if both M^{-1} and N are nonnegative [4], [5], [24], [25]. In [2], it was shown that if M_i is given by (2.7), then each splitting $A = M_i - N_i$ is a regular splitting, and thus a splitting of nonnegative type. The same proof applies here.

We begin our analysis by studying the diagonals of certain matrices.

LEMMA 3.2. *Let M_i be given by (2.7) and let $A = M_i - N_i$; then all entries in the diagonals of $T_i = M_i^{-1}N_i$, $i = 1, \dots, p$, are zero, and thus $\text{diag}(\sum_{i=1}^p E_i T_i) = O$, where E_i is given by (2.1).*

¹Some authors, e.g., [4], [5], call this *semiconvergent*.

Proof. Let us fix i as any of the indices from 1 to p . Recall the combinatorial zero structure of M_i (2.8) and note that from (2.7) we have that M_i^{-1} has zero blocks in the same locations as M_i . Therefore we have

$$(3.1) \quad M_i^{-1}(j, k) = 0 \quad \text{if } j \neq k \text{ and } j \notin S_i \text{ or } k \notin S_i.$$

The structure of $N_i = M_i - A$, taking into account (2.7), is given by

$$(3.2) \quad N_i(j, k) = \begin{cases} 0 & \text{if } j \in S_i \text{ and } k \in S_i, \\ 0 & \text{if } j = k, \\ |A(j, k)| & \text{otherwise.} \end{cases}$$

We study the (j, j) entry of $T_i = M_i^{-1}N_i$. To that end, denote $a_{jk} = M_i^{-1}(j, k)$, $b_{jk} = N_i(j, k)$ and write

$$T_i(j, j) = \sum_{k=1}^n a_{jk} b_{kj}.$$

Let us assume first that $j \neq k$. If one of the indices j or k (or both) does not belong to S_i then from (3.1) we obtain $a_{jk} = 0$. If, on the contrary, k and j belong to S_i , then from (3.2) we get $b_{kj} = 0$. In the case $j = k$ we also have that $b_{jj} = 0$. Then we conclude that $a_{jk} b_{kj} = 0$ for all $j, k = 1, \dots, n$ and therefore $T_i(j, j) = 0$ for all $j = 1, \dots, n$. Since i is an arbitrary index, $i = 1, \dots, p$, the lemma follows. $\square$

We are ready to prove the main convergence theorem.

THEOREM 3.3. *Let $A = I - B$, where B is an irreducible $n \times n$ column stochastic matrix. Let $p > 1$ be a positive integer and $A = M_i - N_i$ be splittings of nonnegative type. Then, if $\theta < 1/q$, the iteration matrix of the additive Schwarz given by (2.10) is convergent and $T_\theta \geq 0$.*

Proof. Since B is irreducible and column stochastic, it has a positive Perron vector v , i.e., $Bv = v$. We will show the following:

- (a) $T_\theta v = v > 0$,
- (b) $T_\theta \geq 0$,
- (c) $\text{diag}(T_\theta) > (0, 0, \dots, 0)$.

As discussed after Lemma 3.1, we will then have that T_θ satisfies (i) and (ii) and is thus convergent.

To prove (a) we write $Av = (I - B)v = v - Bv = 0$ and from (2.10) we conclude that $T_\theta v = v > 0$.

To prove (b) we use (2.3) and the hypothesis that $\theta < 1/q$ and write

$$I \geq \frac{1}{q} \sum_{i=1}^p E_i \geq \theta \sum_{i=1}^p E_i,$$

from which we have

$$T_\theta = I - \theta \sum_{i=1}^p E_i M_i^{-1} A \geq \theta \sum_{i=1}^p E_i - \theta \sum_{i=1}^p E_i M_i^{-1} A = \theta \sum_{i=1}^p E_i (I - M_i^{-1} A) \geq 0,$$

where we have used that $(I - M_i^{-1} A) = M_i^{-1} N_i \geq 0$, since by hypothesis the splittings are of nonnegative type.

Finally, to prove (c) we have that if $\theta < 1/q$, $\text{diag}(I) > \text{diag}(\theta \sum_{i=1}^p E_i)$, and then

$$\begin{aligned} \text{diag}(T_\theta) &= \text{diag} \left(I - \theta \sum_{i=1}^p E_i M_i^{-1} A \right) > \text{diag} \left(\theta \sum_{i=1}^p E_i - \theta \sum_{i=1}^p E_i M_i^{-1} A \right) \\ &= \theta \text{diag} \left(\sum_{i=1}^p E_i M_i^{-1} N_i \right) = O, \end{aligned}$$

where we have used Lemma 3.2. $\square$

We mention that Theorem 3.3 is valid for any singular M -matrix A with a one-dimensional null space spanned by a positive vector v . In particular it applies to ergodic Markov chains.

We also mention that there is an induced splitting $A = M - N$ with $T_\theta = M^{-1}N$ such that it is weak regular. The next section is devoted to showing this existence.

4. Nonsingularity and induced splitting. In this section we show that for the case $q = 2$, the matrix $G = \sum_{i=1}^p E_i M_i^{-1}$ is nonsingular, and we use this fact to analyze the induced splitting. We begin by laying the groundwork with some definitions and preliminary results. We then analyze the case of A_i , $i = 1, \dots, p$, being consecutive principal submatrices of A . This leads the way to the general case when each A_i is any symmetric permutation of a principal submatrix of A , i.e., of the form (2.4).

The structure of the overlap is quite simple when the blocks are consecutive overlapping diagonal blocks. To handle this situation we consider submatrices of A defined by the subscripts in the sets S_i given by (2.2). We denote by $A(S_i)$ a submatrix of A with rows and columns given by a set S_i . In particular, when the subscripts in S_i are consecutive, i.e., of the form $\{i_1, i_1 + 1, i_1 + 2, \dots, i_1 + k - 1\}$, the submatrix $A(S_i)$ is called a *consecutive principal submatrix* of A and is denoted by $A([i, j])$, where $i = i_1$, $j = i_1 + k - 1$. Moreover, denoting by F_{qr} an *elementary permutation matrix* obtained by permuting the rows q and r of an identity matrix I , a permutation matrix given by

$$(4.1) \quad \pi = F_{k,j+k-1} \cdots F_{2,j+1} F_{1,j}$$

is called a *consecutive permutation matrix*.

The following two lemmas are direct consequences of the above definitions and therefore their proofs are omitted.

LEMMA 4.1. *If π_i is a consecutive permutation matrix given by (4.1) with $j = j_i$ and $k = n_i$, then A_i given by (2.4) is the consecutive principal submatrix of order n_i of A , given by*

$$(4.2) \quad A_i = A([j_i, j_i + n_i - 1]).$$

LEMMA 4.2. *For each $i = 1, \dots, p$, let the matrices π_i be consecutive permutation matrices given by (4.1) with $j = j_i$ and $k = n_i$. Let $m_i = j_i + n_i$. Then the rows and columns indexed by $\ell = 1, \dots, j_i - 1$ and $\ell = m_i, \dots, n$ of the matrix G_i given by (2.9) with $A_i = A([j_i, j_i + n_i - 1])$ are zero rows and columns.*

From Lemma 4.2 and the definition of consecutive principal submatrix we have

$$A_i^{-1} = G_i([j_i, j_i + n_i - 1])$$

and, as long as each A_i is given by (4.2), we obtain that the position of each consecutive principal submatrix A_i in A is the same as the position of each A_i^{-1} in G_i .

We note that given p subdomains that cover the whole main diagonal of A and provided that each A_i is a consecutive principal submatrix and $j_1 = 1$, $j_{i+1} \geq j_i$, it is easy to see that the size of the overlapping block between A_i and A_{i+1} is given by

$$(4.3) \quad p_i = n_i + j_i - j_{i+1}, \quad i = 1, 2, \dots, p-1.$$

THEOREM 4.3. *Let $A \in \mathbb{R}^{n \times n}$ be such that any principal submatrix is a non-singular M -matrix. Let $p \geq 2$ be the number of subdomains and let the matrices A_i with $n_i < n$, $i = 1, \dots, p$, be defined by consecutive sets of indices S_i , such that $S_1 = \{1, \dots, n_1\}$, $S_i = \{j_i, \dots, j_i + n_i - 1\}$, $i = 2, \dots, p-1$, and $S_p = \{j_p, \dots, n\}$, verifying that $S_i \cap S_{i+k} = \emptyset$, $k \geq 2$, and such that each index j belongs to some set S_i for $j = 1, \dots, n$. Then $G = \sum_{i=1}^p E_i M_i^{-1}$ is nonsingular.*

Proof. Let v be a vector in the null space of G , i.e., $Gv = 0$. We want to show that $v = 0$. We assume for now that $p > 3$ and comment on the cases $p = 2, 3$ later. If there is no overlap, i.e., $q = 1$, we have a block diagonal matrix, and the theorem holds. We consider the case where there is some overlap, i.e., $q = 2$.

We partition each consecutive principal submatrix A_i , $i = 2, \dots, p-1$, into a 3×3 block corresponding to the two overlap portions and one without overlap as follows:

$$(4.4) \quad A_i = \begin{bmatrix} A_{i,11} & A_{i,12} & A_{i,13} \\ A_{i,21} & A_{i,22} & A_{i,23} \\ A_{i,31} & A_{i,32} & A_{i,33} \end{bmatrix},$$

where $A_{i,11}$ is $p_{i-1} \times p_{i-1}$, $A_{i,33}$ is $p_i \times p_i$ correspond to the two overlap portions, and $A_{i,22}$ is square of order $n_i - p_i - p_{i-1}$. A_1 and A_p are similarly partitioned as

$$(4.5) \quad A_1 = \begin{bmatrix} A_{1,22} & A_{1,23} \\ A_{1,32} & A_{1,33} \end{bmatrix}, \quad A_p = \begin{bmatrix} A_{p,11} & A_{p,12} \\ A_{p,21} & A_{p,22} \end{bmatrix}.$$

The important observation is that because of the overlap, we have that

$$(4.6) \quad A_{i,33} = A_{i+1,11}, \quad i = 1, \dots, p-1.$$

We partition v conformally to the partition of these blocks as

$$v^T = [v_{1,2}, v_{2,1}, v_{2,2}, v_{3,1}, v_{3,2}, \dots, v_{p-1,2}, v_{p,1}, v_{p,2}],$$

where $v_{i,1} \in \mathbb{R}^{p_{i-1}}$, $i = 2, \dots, p$, are the overlapping variables.

Using the structure of G_i as in (2.9) as described in Lemma 4.2, we write

$$(4.7) \quad Gv = \sum_{i=1}^p G_i v = \begin{bmatrix} A_1^{-1} \begin{bmatrix} v_{1,2} \\ v_{2,1} \end{bmatrix} \\ 0 \end{bmatrix} + \begin{bmatrix} 0 \\ A_2^{-1} \begin{bmatrix} v_{2,1} \\ v_{2,2} \\ v_{3,1} \end{bmatrix} \\ 0 \end{bmatrix} + \begin{bmatrix} 0 \\ 0 \\ A_3^{-1} \begin{bmatrix} v_{3,1} \\ v_{3,2} \\ v_{4,1} \end{bmatrix} \\ 0 \end{bmatrix} \\ + \dots + \begin{bmatrix} 0 \\ A_{p-1}^{-1} \begin{bmatrix} v_{p-1,1} \\ v_{p-1,2} \\ v_{p,1} \end{bmatrix} \\ 0 \end{bmatrix} + \begin{bmatrix} 0 \\ A_p^{-1} \begin{bmatrix} v_{p,1} \\ v_{p,2} \end{bmatrix} \end{bmatrix} = 0.$$

We partition each of these vectors in a similar manner, so that

$$(4.8) \quad \begin{bmatrix} x_1 \\ z_1 \end{bmatrix} = A_1^{-1} \begin{bmatrix} v_{1,2} \\ v_{2,1} \end{bmatrix}, \quad \begin{bmatrix} x_i \\ y_i \\ z_i \end{bmatrix} = A_i^{-1} \begin{bmatrix} v_{i,1} \\ v_{i,2} \\ v_{i+1,1} \end{bmatrix}, \quad i = 2, \dots, p-1,$$

$$(4.9) \quad \begin{bmatrix} x_p \\ z_p \end{bmatrix} = A_p^{-1} \begin{bmatrix} v_{p,1} \\ v_{p,2} \end{bmatrix}.$$

With this notation, it follows from (4.7) that

$$(4.10) \quad x_1 = 0, \quad y_i = 0, \quad i = 2, \dots, p-1, \quad z_p = 0,$$

$$(4.11) \quad \text{and } z_i + x_{i+1} = 0, \quad i = 1, \dots, p-1.$$

Using the notation (4.4), (4.5), and the identities (4.6) and (4.10), we obtain from (4.8)–(4.9)

$$(4.12) \quad \begin{bmatrix} v_{1,2} \\ v_{2,1} \end{bmatrix} = A_1 \begin{bmatrix} x_1 \\ z_1 \end{bmatrix} = \begin{bmatrix} A_{1,23}z_1 \\ A_{2,11}z_1 \end{bmatrix},$$

$$(4.13) \quad \begin{bmatrix} v_{i,1} \\ v_{i,2} \\ v_{i+1,1} \end{bmatrix} = A_i \begin{bmatrix} x_i \\ y_i \\ z_i \end{bmatrix} = \begin{bmatrix} A_{i,11}x_i + A_{i,13}z_i \\ A_{i,21}x_i + A_{i,23}z_i \\ A_{i,31}x_i + A_{i+1,11}z_i \end{bmatrix}, \quad i = 2, \dots, p-1,$$

$$(4.14) \quad \begin{bmatrix} v_{p,1} \\ v_{p,2} \end{bmatrix} = A_p \begin{bmatrix} x_p \\ z_p \end{bmatrix} = \begin{bmatrix} A_{p,11}x_p \\ A_{p,21}x_p \end{bmatrix}.$$

Equating the two occurrences of $v_{i,1}$, for each $i = 2, \dots, p$, in (4.12)–(4.14) and using (4.11), we obtain the following $p-1$ equations:

$$(4.15) \quad \begin{cases} 2A_{2,11}x_2 & + A_{2,13}z_2 & & = 0, \\ A_{i-1,31}z_{i-2} & + 2A_{i,11}x_i & + A_{i,13}z_i & = 0, \quad i = 3, \dots, p-1, \\ A_{p-1,31}z_{p-2} & + 2A_{p,11}x_p & & = 0. \end{cases}$$

The coefficient matrix of the system (4.15) is the block tridiagonal matrix T given by

$$T = \begin{bmatrix} 2A_{2,11} & A_{2,13} & & & & \\ A_{2,31} & 2A_{3,11} & A_{3,13} & & & \\ & A_{3,31} & 2A_{4,11} & A_{4,13} & & \\ & & A_{4,31} & 2A_{5,11} & A_{5,13} & \\ & & & \ddots & \ddots & \ddots \\ & & & & A_{p-2,31} & 2A_{p-1,11} & A_{p-1,13} \\ & & & & & A_{p-1,31} & 2A_{p,11} \end{bmatrix}.$$

We now show that T is nonsingular. To that end consider first the matrix U obtained from A by removing the first and last block of rows and columns, and all rows and columns corresponding to nonoverlapping blocks, i.e., the rows and columns corresponding to the matrices $A_{i,22}$, $i = 1, \dots, p$; cf. (4.4), (4.5), (4.12)–(4.14). Since U is a principal submatrix of A , it is a nonsingular M -matrix, and thus we can write

$U = sI - B$ with $B \geq O$, and $\rho(B) < s$. Consider now the matrix

$$S = \begin{bmatrix} A_{2,11} & A_{2,13} & & & & & \\ A_{2,31} & A_{3,11} & A_{3,13} & & & & \\ & A_{3,31} & A_{4,11} & A_{4,13} & & & \\ & & A_{4,31} & A_{5,11} & A_{5,13} & & \\ & & & \ddots & \ddots & \ddots & \\ & & & & A_{p-2,31} & A_{p-1,11} & A_{p-1,13} \\ & & & & & A_{p-1,31} & A_{p,11} \end{bmatrix}.$$

It follows that $S = sI - B'$ with $O \leq B' \leq B$, and therefore, $\rho(B') \leq \rho(B) < s$, implying that S is a nonsingular M -matrix. Let $S = M - N$ be the block Jacobi splitting, which is a convergent regular splitting, so that M is nonsingular and $\rho(M^{-1}N) < 1$. We have that T is a Z -matrix, and that the splitting $T = 2M - N$ is a regular splitting. Since $\rho(\frac{1}{2}M^{-1}N) = \frac{1}{2}\rho(M^{-1}N) < 1$, we have that $I - \frac{1}{2}M^{-1}N$ is nonsingular, and therefore $T = 2M(I - \frac{1}{2}M^{-1}N)$ is also nonsingular.

Thus, from (4.15) and (4.11), $x_i = 0$, $i = 2, \dots, p$, and $z_i = 0$, $i = 1, \dots, p-1$. This, together with (4.10), is used in (4.12)–(4.14) to conclude that $v = 0$, and thus G is nonsingular.

For the cases $p = 2$ and $p = 3$ one uses the same general argument, with only one or two sets of overlapping variables, respectively. In the case of $p = 2$, (4.12) and (4.13), together with (4.11), lead to $2A_{2,11}x_2 = 0$. In the case of $p = 3$, (4.14) is also used, thus obtaining the 2×2 block system which is the upper-left part of (4.15). For $p = 2$ the nonsingularity of $A_{2,11}$ leads to $x_2 = 0$ and thus $z_1 = 0$, from which $v = 0$. For $p = 3$, the matrix T is shown to be nonsingular in the same manner as in the general case, so that $x_2 = 0$, $z_2 = 0$, and thus $v = 0$. $\square$

We remark that the case $q = 2$ considered here is of practical importance. Theorem 4.3 is valid with some milder hypothesis, namely, that in addition $S_i \subset S_j$ for some i, j , although this case seldom appears in practice. We can also consider sets S_i with nonconsecutive indices, as shown in the following theorem.

THEOREM 4.4. *Let $A \in \mathbb{R}^{n \times n}$ be such that any principal submatrix is a nonsingular M -matrix. Let $p \geq 2$ be the number of subdomains and let the matrices A_i be defined by a set of indices S_i verifying that $S_i \cap S_{i+k} = \emptyset$, $k \geq 2$, and such that each index j belongs to some set S_i , for $j = 1, \dots, n$. Then $G = \sum_{i=1}^p E_i M_i^{-1}$ is nonsingular.*

Proof. We mentioned in section 2 that A_i is a symmetric permutation of a principal submatrix of A . In fact, we can say that any matrix A_i of the form (2.4) is a consecutive principal submatrix of $Q_i A Q_i^T$, a symmetric permutation of A , for each $i = 1, \dots, p$. The key observation here is that we can choose the same permutation $Q = Q_i$ for all $i = 1, \dots, p$. This follows from the fact that the variables corresponding to the overlap have to match in their location once permuted. Therefore, we can apply Theorem 4.3 to $Q A Q^T$ and obtain that $\hat{G} = \sum_{i=1}^p Q^T R_i^T A_i^{-1} R_i Q$ is nonsingular, from which it follows that $G = Q \hat{G} Q^T$ is also nonsingular. $\square$

With the nonsingularity of G established, we can show the existence of an induced splitting.

THEOREM 4.5. *Let $A \in \mathbb{R}^{n \times n}$ be such that any principal submatrix is a nonsingular M -matrix. Let $p \geq 2$ be the number of subdomains and let the matrices A_i be given by (2.4) with $n_i < n$, and defined by a set of indices S_i verifying that $S_i \cap S_{i+k} = \emptyset$, $k \geq 2$, and such that each index j belongs to some set S_i , for $j = 1, \dots, n$. Then there*

exists a splitting $A = M - N$ such that the iteration matrix T_θ is given by $T_\theta = M^{-1}N$, where

$$(4.16) \quad M^{-1} = \theta \sum_{i=1}^p E_i M_i^{-1}.$$

Furthermore, this splitting is a convergent weak regular splitting.

Proof. From Theorem 4.4 we have that $G = \sum_{i=1}^p E_i M_i^{-1}$ is nonsingular. We can then define $M = (1/\theta)G^{-1}$ and N such that $A = M - N$. It follows that

$$(4.17) \quad T_\theta = I - M^{-1}A$$

is the iteration matrix of the additive Schwarz iterations. For the second part, it is easy to see that all the conditions of Theorem 3.3 hold and therefore $T_\theta = M^{-1}N \geq O$. As we also have from (4.16) that $M^{-1} \geq O$, we conclude that $A = M - N$ is a weak regular splitting. Using Theorem 3.3 again, T_θ is convergent. $\square$

We comment here that the splitting defined by (4.16) is not the only splitting associated with T_θ ; there are infinitely many splittings $A = M - N$ for which (4.17) holds [3].

5. Illustrative examples. We begin with an example in which the induced splitting $A = M - N$ is (convergent) weak regular for $\theta = 0.3$ (in accordance with Theorem 4.5) but is not a regular splitting.

Example 5.1. Consider the matrix

$$(5.1) \quad A = \begin{bmatrix} 4/7 & -6/23 & -6/23 & -5/7 & -18/31 \\ -2/21 & 83/92 & -9/92 & 0 & -2/93 \\ -4/21 & -27/92 & 65/92 & 0 & -4/93 \\ -1/7 & -9/46 & -9/46 & 6/7 & -8/31 \\ -1/7 & -7/46 & -7/46 & -1/7 & 28/31 \end{bmatrix},$$

which is a singular M -matrix, and $A = I - B$ with B an $n \times n$ irreducible column stochastic matrix. Let us consider two subdomains such that $p = q = 2$. We use

$$R_1 = \begin{bmatrix} 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 1 & 0 & 0 \\ 1 & 0 & 0 & 0 & 0 \end{bmatrix}, \quad R_2 = \begin{bmatrix} 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 1 \\ 0 & 1 & 0 & 0 & 0 \end{bmatrix}.$$

Then, we have $S_1 = \{1, 3, 5\}$, $S_2 = \{2, 3, 4, 5\}$, and

$$A_1 = \begin{bmatrix} 28/31 & -7/46 & -1/7 \\ -4/93 & 65/92 & -4/21 \\ -18/31 & -6/23 & 4/7 \end{bmatrix},$$

$$A_2 = \begin{bmatrix} 65/92 & 0 & -4/93 & -27/92 \\ -9/46 & 6/7 & -8/31 & -9/46 \\ -7/46 & -1/7 & 28/31 & -7/46 \\ -9/92 & 0 & -2/93 & 83/92 \end{bmatrix}.$$

It follows that the splittings $A = M_1 - N_1$ and $A = M_2 - N_2$ are regular splittings, as expected, and one has

$$T_1 = M_1^{-1}N_1 = \begin{bmatrix} 0 & 9702/7157 & 0 & 15313/7157 & 0 \\ 184/1743 & 0 & 9/83 & 0 & 184/7719 \\ 0 & 5815/7157 & 0 & 30728/50099 & 0 \\ 1/6 & 21/92 & 21/92 & 0 & 28/93 \\ 0 & 3720/7157 & 0 & 8587/14314 & 0 \end{bmatrix},$$

$$T_2 = M_2^{-1}N_2 = \begin{bmatrix} 0 & 21/46 & 21/46 & 5/4 & 63/62 \\ 13861/91938 & 0 & 0 & 0 & 0 \\ 32231/91938 & 0 & 0 & 0 & 0 \\ 74/199 & 0 & 0 & 0 & 0 \\ 4619/15323 & 0 & 0 & 0 & 0 \end{bmatrix}.$$

Note that $\text{diag}(T_i) = O$, according to Lemma 3.2. For the splitting $A = M - N$, with M given by (4.16) with $\theta = 0.3$, we have

$$M = \begin{bmatrix} 89825/56742 & 1865/4439 & -10/23 & 3005/18914 & -30/31 \\ 9085/170226 & 155995/53268 & -15/92 & -415/56742 & -10/279 \\ -20/63 & -45/92 & 325/276 & 0 & -20/279 \\ 5120/28371 & -7955/8878 & -15/46 & 26340/9457 & -40/93 \\ -5/21 & -35/138 & -35/138 & -5/21 & 140/93 \end{bmatrix},$$

for which $M^{-1} \geq O$,

$$N = \begin{bmatrix} 57401/56742 & 3023/4439 & -4/23 & 16515/18914 & -12/31 \\ 25297/170226 & 53969/26634 & -3/46 & -415/56742 & -4/279 \\ -8/63 & -9/46 & 65/138 & 0 & -8/279 \\ 9173/28371 & -3109/4439 & -3/23 & 18234/9457 & -16/93 \\ -2/21 & -7/69 & -7/69 & -2/21 & 56/93 \end{bmatrix},$$

and $T_\theta = M^{-1}N \geq O$ with $\sigma(T_\theta) = \{1, 2/5, 2/5, 2/5, 7/10\}$.

For the matrix (5.1), we present in Table 2 the value of the convergence factor $\gamma(T)$ for several cases of (damped) block Jacobi and (damped) additive Schwarz. The different variables in each of the blocks are given in Table 1. In all the cases considered, we have that the maximum overlap is $q = 2$.

It can be readily appreciated that for the same damping factor θ additive Schwarz has a lower convergence factor than block Jacobi.

TABLE 1

Method	p	Variables in each block S_i
Block Jacobi (BJ1)	2	$S_1 = \{1, 3\}, S_2 = \{2, 4, 5\}$
Block Jacobi (BJ2)	2	$S_1 = \{1, 5\}, S_2 = \{2, 3, 4\}$
Additive Schwarz (AS1)	2	$S_1 = \{1, 3, 5\}, S_2 = \{2, 3, 4, 5\}$
Block Jacobi (BJ3)	3	$S_1 = \{1, 3\}, S_2 = \{2, 5\}, S_3 = \{4\}$
Additive Schwarz (AS2)	3	$S_1 = \{1, 3, 5\}, S_2 = \{2, 3, 5\}, S_3 = \{1, 4\}$

TABLE 2
Convergence factor $\gamma(T)$ for blocks described in Table 1. Matrix A is (5.1).

θ	BJ1	BJ2	AS1	BJ3	AS2
0.1	0.90003	0.90011	0.90000	0.8996	0.8892
0.2	0.80014	0.80051	0.80000	0.7992	0.7784
0.3	0.70036	0.70131	0.70000	0.6989	0.6675
0.4	0.60074	0.60271	0.60000	0.5987	0.5567
0.5	0.50139	0.50507	0.50000	0.4986	0.4459

Example 5.2. We consider matrices representing a model of an interactive computer system, described as Example 1 in [16] or in [21]. We used the matrices of orders $n = 286, 1771$, and 5456 , with $1606, 11011$, and 35216 nonzeros, respectively. These matrices have been used for numerical experiments by several authors, including [8], [9], [15], [16], [21], and can be obtained from [19]. We ran several experiments with varying blocks using (damped) block Jacobi and (damped) additive Schwarz. Our initial vector is $x^0 = (1, 1, \dots, 1)$. We report the convergence factor $\gamma(T_\theta)$, the number of iterations, and the CPU time for convergence, i.e., for the 2-norm of the residual $r = Ax^k$ to be reduced to less than 10^{-10} . The numerical experiments are executed in only one processor of an IBM-1350 at the Universitat Politècnica de València, running MATLAB 6. The times are in seconds. For comparison, we also report these three quantities for block Gauss–Seidel (GS) for multiplicative Schwarz (MS), i.e, using T_μ in selected cases. For block Jacobi, the best results are for $\theta = 0.90$, and experiments with this value of θ are the only ones we report in Table 3. Table 4 shows the structure of the subdomains. These experiments illustrate the potential advantage of Schwarz methods, i.e., of the overlap, over the widely used classical iterative methods for Markov chains.

TABLE 3

n		p	q	θ	$\gamma(T_\theta)$	Iter.	Time	$\gamma(T_\mu)$	Iter.	Time
286	BJ	3	1	0.90	0.9992	13028	29.57			
286	BJ	5	1	0.90	0.9997	32439	86.79			
286	AS	3	2	0.49	0.9498	278	0.81			
286	AS	5	3	0.49	0.9914	1392	3.93			
286	AS	5	2	0.49	0.9316	164	0.40			
286	AS	3	2	0.81	0.9170	166	0.45			
286	AS	5	3	0.66	0.9885	1032	3.12			
286	AS	5	2	0.63	0.9120	127	0.31			
1771	BJ/GS	6	1	0.90	0.9978	2178	58.00	0.9951	1209	106.7
1771	AS/MS	6	2	0.49	0.9882	217	9.39	0.9522	107	9.90
1771	AS	6	2	0.90	0.9782	118	5.29			
5456	BJ/GS	10	1	0.90	0.9988	9592	1636.5	0.9972	4551	3292.
5456	AS/MS	10	2	0.49	0.9716	175	45.09	0.8876	45	32.00
5456	AS	10	2	0.90	0.9479	95	32.15			

6. Conclusion. We have presented a convergence theory for additive Schwarz methods for linear systems whose matrix is a singular M -matrix with a one-dimensional null space. This applies in particular to ergodic Markov chains. The convergence theory is illustrated with several numerical experiments, where it is shown that using additive Schwarz, i.e., incorporating overlap variables common to more than one block, improves the asymptotic convergence factor as well as execution times.

TABLE 4

n		p	q	S_i
286	BJ	3	1	$\{1 : 100\}, \{101 : 200\}, \{201 : 286\}$
286	BJ	5	1	$\{1 : 66\}, \{67 : 139\}, \{140 : 216\}, \{217 : 273\}, \{274 : 286\}$
286	AS	3	2	$\{1 : 100\}, \{80 : 200\}, \{120 : 286\}$
286	AS	5	3	$\{1 : 30\}, \{25 : 70\}, \{25 : 80\}, \{40 : 150\}, \{100 : 286\}$
286	AS	5	2	$\{1 : 83\}, \{47 : 131\}, \{84 : 178\}, \{132 : 225\}, \{179 : 286\}$
286	AS	3	2	$\{1 : 100\}, \{80 : 200\}, \{120 : 286\}$
286	AS	5	3	$\{1 : 30\}, \{25 : 70\}, \{25 : 80\}, \{40 : 150\}, \{100 : 286\}$
286	AS	5	2	$\{1 : 83\}, \{47 : 131\}, \{84 : 178\}, \{132 : 225\}, \{179 : 286\}$
1771	BJ/GS	6	1	$\{1 : 300\}, \{301 : 601\}, \{602, 902\}, \{903 : 1203\},$ $\{1204 : 1503\}, \{1504 : 1771\}$
1771	AS/MS	6	2	$\{1 : 300\}, \{281 : 600\}, \{500 : 900\}, \{800 : 1200\},$ $\{1000 : 1500\}, \{1400 : 1771\}$
1771	AS	6	2	$\{1 : 300\}, \{281 : 600\}, \{500 : 900\}, \{800 : 1200\},$ $\{1000 : 1500\}, \{1400 : 1771\}$
5456	BJ/GS	10	1	$\{1 : 600\}, \{601 : 1200\}, \{1201 : 1800\}, \{1801 : 2400\},$ $\{2401 : 3000\}, \{3001 : 3600\}, \{3601 : 4200\},$ $\{4201 : 4800\}, \{4801 : 5400\}, \{5401 : 5456\}$
5456	AS/MS	10	2	$\{1 : 900\}, \{601 : 1500\}, \{1201 : 2100\}, \{1801 : 2700\},$ $\{2401 : 3300\}, \{3001 : 3900\}, \{3601 : 4500\},$ $\{4201 : 5100\}, \{4801 : 5455\}, \{5401 : 5456\}$
5456	AS	10	2	$\{1 : 900\}, \{601 : 1500\}, \{1201 : 2100\}, \{1801 : 2700\},$ $\{2401 : 3300\}, \{3001 : 3900\}, \{3601 : 4500\},$ $\{4201 : 5100\}, \{4801 : 5455\}, \{5401 : 5456\}$

Acknowledgment. We thank Michele Benzi for his comments on an earlier manuscript.

REFERENCES

- [1] G. ALEFELD AND N. SCHNEIDER, *On square roots of M -matrices*, Linear Algebra Appl., 42 (1982), pp. 119–132.
- [2] M. BENZI, A. FROMMER, R. NABBEN, AND D. B. SZYLD, *Algebraic theory of multiplicative Schwarz methods*, Numer. Math., 89 (2001), pp. 605–639.
- [3] M. BENZI AND D. B. SZYLD, *Existence and uniqueness of splittings for stationary iterative methods with applications to alternating methods*, Numer. Math., 76 (1997), pp. 309–321.
- [4] A. BERMAN AND R. J. PLEMMONS, *Nonnegative Matrices in the Mathematical Sciences*, Academic Press, New York, 1979.
- [5] A. BERMAN AND R. J. PLEMMONS, *Nonnegative Matrices in the Mathematical Sciences*, Classics Appl. Math. 9, SIAM, Philadelphia, 1994.
- [6] E. BOHL AND I. MAREK, *A model of amplification*, J. Comput. Appl. Math., 63 (1995), pp. 27–47.
- [7] R. BRU, F. PEDROCHE, AND D. B. SZYLD, *Overlapping additive and multiplicative Schwarz iterations for H -matrices*, Linear Algebra Appl., 393 (2004), pp. 91–105.
- [8] H. CHOI AND D. B. SZYLD, *Application of threshold partitioning of sparse matrices to Markov chains*, in Proceedings of the IEEE International Computer Performance and Dependability Symposium (IPDS'96), Urbana-Champaign, IL, 1996, IEEE Computer Society Press, Los Alamitos, CA, pp. 158–165.
- [9] T. DAYAR AND W. J. STEWART, *Comparison of partitioning techniques for two-level iterative solvers on large, sparse Markov chains*, SIAM J. Sci. Comput., 21 (2000), pp. 1691–1705.
- [10] A. FROMMER AND D. B. SZYLD, *Weighted max norms, splittings, and overlapping additive Schwarz iterations*, Numer. Math., 83 (1999), pp. 259–278.
- [11] I. MAREK AND D. B. SZYLD, *Comparison theorems for the convergence factor of iterative methods for singular matrices*, Linear Algebra Appl., 316 (2000), pp. 67–87.
- [12] I. MAREK AND D. B. SZYLD, *Comparison of convergence of general stationary iterative methods for singular matrices*, SIAM J. Matrix Anal. Appl., 24 (2002), pp. 68–77.

- [13] I. MAREK AND D. B. SZYLD, *Algebraic Schwarz methods for the numerical solution of Markov chains*, Linear Algebra Appl., 386 (2004), pp. 67–81.
- [14] T. P. MATHEW, *Uniform convergence of the Schwarz alternating method for solving singularly perturbed advection-diffusion equations*, SIAM J. Numer. Anal., 35 (1998), pp. 1663–1683.
- [15] V. MIGALLÓN, J. PENADÉS, AND D. B. SZYLD, *Experimental study of parallel iterative solutions of Markov chains with block partitions*, in Numerical Solution of Markov Chains (NSMC'99), B. Plateau, W. J. Stewart, and M. Silva, eds., Prensas Universitarias de Zaragoza, Zaragoza, Spain, 1999, pp. 96–110.
- [16] B. PHILIPPE, Y. SAAD, AND W. J. STEWART, *Numerical methods in Markov chain modeling*, Oper. Res., 40 (1992), pp. 1156–1179.
- [17] A. QUARTERONI AND A. VALLI, *Domain Decomposition Methods for Partial Differential Equations*, Oxford Science Publications, Clarendon Press, Oxford, UK, 1999.
- [18] B. F. SMITH, P. E. BJØRSTAD, AND W. D. GROPP, *Domain Decomposition: Parallel Multi-level Methods for Elliptic Partial Differential Equations*, Cambridge University Press, Cambridge, UK, 1996.
- [19] W. J. STEWART, *MARCA Models: A Collection of Markov Chain Models*, http://www.csc.ncsu.edu/faculty/stewart/MARCA_Models/MARCA_Models.html.
- [20] W. J. STEWART, *Introduction to the Numerical Solution of Markov Chains*, Princeton University Press, Princeton, NJ, 1994.
- [21] W. J. STEWART AND W. WU, *Numerical experiments with iteration and aggregation for Markov chains*, ORSA J. Comput., 4 (1992), pp. 336–350.
- [22] D. B. SZYLD, *Equivalence of convergence conditions for iterative methods for singular equations*, Numer. Linear Algebra Appl., 1 (1994), pp. 151–154.
- [23] A. TOSELLI AND O. B. WIDLUND, *Domain Decomposition—Algorithms and Theory*, Springer Ser. Comput. Math. 34, Springer, Berlin, 2005.
- [24] R. S. VARGA, *Matrix Iterative Analysis*, Prentice-Hall, Englewood Cliffs, NJ, 1962.
- [25] R. S. VARGA, *Matrix Iterative Analysis*, Springer Ser. Comput. Math. 27, 2nd ed., Springer, Berlin, 2000.

CRITICAL POINTS OF THE SINGULAR VALUE DECOMPOSITION*

KEVIN A. O'NEIL[†]

Abstract. The singular value decomposition (SVD) is a factorization that is discontinuous on the subset of matrices having repeated singular values. In this paper the SVD is studied in the vicinity of this critical set. Each one-parameter C^k perturbation transversal to the critical set is shown to uniquely determine an SVD at the critical point that extends to an SVD along the perturbation path that is C^{k-1} in the perturbation parameter. Derivatives of the singular vectors at the critical point are found explicitly. Application is made to the effect on the singular vectors of perturbations from a matrix in the critical set and compared to the information provided by the $\sin(\theta)$ theorem. Estimates of the derivative of the singular vectors are applied to inequalities involving the matrix absolute value, such as the generalized Araki–Yamagami inequality.

Key words. singular value decomposition, singular vector perturbation, matrix absolute value

AMS subject classifications. 15A, 65F

DOI. 10.1137/040611719

1. Introduction. The singular value decomposition (SVD) of a complex or real matrix M is a useful tool in linear algebra. If M has full rank and the singular values of M are distinct, then the singular values and subspaces are smooth functions of the elements of M ; one can say that the SVD is smooth on the subset of such matrices. If the singular values of M are not distinct, or if M is not square and has less than full rank, then the SVD is discontinuous at M , and such matrices may be called *critical points* of the SVD.

There are many results in the literature concerning the effect of perturbations from a regular point on the singular vectors. These range from the very general $\sin(\theta)$ theorems [4, 12], which give inequalities based on only the norm of the perturbation, to differentials for the singular values and vectors [6, 9, 12], which show the first-order effect of a movement in a given direction in matrix space. Discontinuities of the SVD obviously raise a barrier to the analysis of perturbations. One line of attack is to study the SVD along a path in matrix space. Rellich [11] showed that if the perturbation of a real symmetric matrix is analytic in the single perturbation parameter t , then so are the eigenvectors. Bunse-Gerstner et al. [2] showed that for analytic perturbations the SVD is analytic and essentially unique, modulo permutations and changes of sign of the singular values and choice of basis vectors for the singular subspaces. Wright [13] noted that the singular vectors of such an analytic SVD satisfied differential equations even at critical points, and indicated how to find these equations in the case of two and three identical singular values. Some applications of the SVD (e.g., control of mechanisms [8]) produce paths that are C^k rather than analytic. For such paths, Dieci and Eirola [5] obtained smoothness results based on the order of coalescing of the singular values at the critical point; extensions may be found in [3]. Lax [7] obtained differential-like relations linking the limit of the eigenvectors of a self-adjoint

*Received by the editors July 15, 2004; accepted for publication (in revised form) by M. Hanke December 2, 2004; published electronically November 22, 2005.

<http://www.siam.org/journals/simax/27-2/61171.html>

[†]Department of Mathematical and Computer Sciences, The University of Tulsa, 600 S. College Ave., Tulsa, OK 74104 (koneil@utulsa.edu).

matrix $A(t)$ as a critical point is approached to eigenvectors of the first derivative of the perturbation restricted to the eigenspace of the repeated eigenvalue.

In this paper an analysis of the SVD in the neighborhood of a critical point is presented. The structure of the discontinuity is clarified by expressing the SVD of a matrix as a point on an affine variety, i.e., the solution to a system of polynomial equations in which the elements of M appear as parameters. This variety is shown to be smooth at all points. Through desingularization of the polynomial system, the singular vectors of matrices in a neighborhood of a matrix on the critical set are shown to depend smoothly on direction from the critical point, at least in the transversal directions (directions not tangent to the critical set.) This provides differential-like information about perturbations of the SVD in the vicinity of a critical point and may prove useful in numerical algorithms for computation of the SVD along a path which begins at a critical point. A by-product of the analysis is a concise formulation of the differential equations discussed in [13] for analytic paths.

A limitation of this approach is the restriction to directions that are transversal to the critical set. This assumption allows many of the subtleties of the analytic case [2] to be avoided, yet is still generic [3]. An advantage of the analysis is that the cause of the loss of smoothness noted above is easily identified (Corollary 5 below). The analysis is somewhat complementary to perturbation inequalities such as the $\sin(\theta)$ theorem, as discussed in section 3.1. Application to inequalities involving the matrix absolute value such as the generalized Araki–Yamagami inequality is given in section 3.2.

2. The SVD along a path through a critical point. Let $M_{m,n}$ denote the space of $m \times n$ complex matrices. The case of square matrices is handled first. The SVD of a matrix $M \in M_{n,n}$ consists of unitary matrices U, V and numbers $d_1 \geq \dots \geq d_n \geq 0$ such that $U^*MV = \text{diag}(d_1, \dots, d_n) := (d_i\delta_{ij})$. The i th columns u_i and v_i of U and V , respectively, are the left and right singular vectors of M corresponding to the singular value d_i , i.e., $d_i u_i = M v_i$. There is a trivial nonuniqueness of the SVD, since u_i, v_i may both be multiplied by a complex number α of modulus 1. We will say two unitary matrices $U, \tilde{U}$ are equivalent if $U^*\tilde{U}$ is diagonal and will consider two SVDs of M to be equivalent if $U^*\tilde{U}$ and $V^*\tilde{V}$ are diagonal and equal. If the singular values of M are distinct, the SVD of M is unique up to equivalence. However, if any singular value is repeated, say d with multiplicity $k > 1$, then there is some freedom to choose bases for the right and left singular subspaces corresponding to d . Specifically, one may form matrices U_d and V_d out of the corresponding k columns of U and V , respectively, and choose any $k \times k$ unitary matrix Q . After replacing the block U_d of U by $U_d Q$ to form a new unitary matrix $\tilde{U}$, and doing the corresponding replacement in V to form $\tilde{V}$, it is clear that $\tilde{U}^*M\tilde{V} = U^*MV$, but U and $\tilde{U}$ are not equivalent in general. If the repeated value is zero, then $\tilde{U}$ and $\tilde{V}$ may be constructed using two different unitary matrices Q, Q' .

Let $\mathcal{S}$ denote the subset of $M_{n,n}$ consisting of matrices with repeated singular values. Suppose that u, v are singular vectors of $M \in \mathcal{S}$ corresponding to a repeated singular value d . Then they are also singular vectors of the matrix $M + \epsilon uv^*$ corresponding to the singular value $d + \epsilon$ and thus appear as columns in the SVD of $M + \epsilon uv^*$. Since u and v can be selected from subspaces of dimension greater than one, an arbitrarily small change in the matrix M can produce a large change in the SVD. In this sense, the SVD is discontinuous at M ; elements of $\mathcal{S}$ will be called *critical points* of the SVD.

In what follows, we will need a criterion that will ensure that motion from a point

in $\mathcal{S}$ in a given direction in matrix space escapes $\mathcal{S}$. Recall the matrices U_d, V_d defined above, and let $\mathcal{P}_H(M) := (M + M^*)/2$ and $\mathcal{P}_S(M) := (M - M^*)/2$ be the projections of M onto the spaces of Hermitian and skew-Hermitian matrices, respectively.

DEFINITION 1. *An $n \times n$ matrix X is transversal to $\mathcal{S}$ at $M \in \mathcal{S}$ if for every d in the set of singular values $\{d_1, \dots, d_n\}$ of M , the matrix $\mathcal{P}_H(U_d^* X V_d)$ has distinct eigenvalues (in the case $d > 0$), or the singular values of the matrix $U_d^* X V_d$ are distinct (in the case $d = 0$.)*

These conditions are trivial for singular values of multiplicity one, and it is readily seen that they are independent of the particular matrices U and V used for the SVD of M . It will be shown below that along the line $M + tX$ the derivatives of the repeated singular values with respect to t at $t = 0$ are given by the eigenvalues of $\mathcal{P}_H(U_d^* X V_d)$ or (if the repeated singular value is 0) the singular values of $U_d^* X V_d$. Therefore the conditions of the definition ensure that the singular values of $M + tX$ are distinct for small nonzero t , with differences that are at least $O(t)$. Hence the line $M + tX$ is indeed transversal to the critical set $\mathcal{S}$ in the geometric sense, i.e., not tangent to $\mathcal{S}$, if and only if X is transversal to $\mathcal{S}$ at M in the sense of Definition 1.

LEMMA 2. *Suppose X is transversal to $\mathcal{S}$ at M . Then there are unitary matrices U and V such that $U^* M V = \text{diag}(d_1, \dots, d_n)$, and the components P_{ij} of the matrix $P = U^* X V$ satisfy the following “alignment conditions”: for all d and for every pair $i < j$ such that $d_i = d_j = d$,*

(L2a) *if $d \neq 0$, then $P_{ij} = -(P_{ji})^*$ and $\text{Re}(P_{ii}) > \text{Re}(P_{jj})$;*

(L2b) *if $d = 0$, then $P_{ij} = P_{ji} = 0$ and $P_{ii} > P_{jj} \geq 0$.*

Furthermore, U and V are unique up to equivalence and depend continuously on the elements of X .

Proof. Choose unitary matrices U, V to begin with such that $U^* M V$ has diagonal elements $d_1 \geq \dots \geq d_n \geq 0$. For each repeated singular value d , let U_d and V_d be as above, and consider the $k \times k$ matrix $P_d = U_d^* X V_d$.

Suppose first that $d > 0$. There is a unitary matrix Q that diagonalizes $\mathcal{P}_H(P_d)$; thus $Q^*(\mathcal{P}_H(P_d))Q = \mathcal{P}_H((U_d Q)^* X (V_d Q))$ is zero off the diagonal and has distinct real diagonal elements. Because the eigenvalues are distinct, we may choose Q so that the diagonal elements are strictly decreasing, and this choice is unique up to equivalence class. If the block U_d of U is replaced by $U_d Q$ and corresponding replacement is made in V , then $U^* M V$ is unchanged, as noted above, and in addition (L2a) will be satisfied for all i, j corresponding to d .

Next consider the case $d = 0$. Let Q, Q' be unitary matrices that give the SVD of P_d , i.e., $Q^* P_d Q' = \text{diag}(\sigma_1, \dots, \sigma_k)$, with distinct $\sigma_i \geq 0$ that are strictly decreasing; Q and Q' are unique up to equivalence. Since $Q^* P_d Q' = (U_d Q)^* X (V_d Q')$ and $M V_d = 0$, replacing U_d by $U_d Q$ in U and V_d by $V_d Q'$ in V does not change the diagonal form of $U^* M V$, and (L2b) is satisfied.

It is clear in both cases that the aligned U and V are unique up to equivalence. Since the matrices $\mathcal{P}_H(P_d)$ and P_d have distinct eigenvalues and singular values, respectively, the corresponding matrices Q and Q' are smooth (indeed, real-analytic) functions of X [12]. The same is therefore true of U and V . $\square$

Now given a path $M(t)$ in $M_{n,n}$ passing through a matrix $M(0) \in \mathcal{S}$, we investigate the existence of an SVD along the trajectory. The basic technique is to observe that the matrices U and V of the SVD of a matrix M satisfy the system of polynomial equations (2.1) below. For M not in $\mathcal{S}$, these equations are nonsingular and solutions $U(t), V(t)$ can be extended from t_0 to a neighborhood of t_0 by the implicit function theorem. As the trajectory passes through $\mathcal{S}$, the system of equations becomes

singular; a new, equivalent system must be constructed that is nonsingular.

THEOREM 3. *Suppose $M(t)$ is a continuous complex $n \times n$ matrix-valued function of a real parameter t , twice differentiable at $t = 0$, and let $d_1, \dots, d_n$ be the singular values of $M(0) \in \mathcal{S}$. If $\dot{M}(0)$ is transversal to $\mathcal{S}$ at $M(0)$, then there are continuous unitary matrix-valued functions $U(t)$, $V(t)$ and continuous real-valued functions $d_1(t), \dots, d_n(t)$ defined for t in a neighborhood of 0, satisfying $d_k(0) = d_k$ for $1 \leq k \leq n$, $d_1(t) > \dots > d_n(t) \geq 0$ for $t > 0$, and*

$$U^*(t)M(t)V(t) = \text{diag}(d_1(t), \dots, d_n(t)).$$

Furthermore, $U(t)$ and $V(t)$ are differentiable at $t = 0$, while the $d_i(t)$ are twice differentiable there.

Proof. We seek matrices $U(t)$, $V(t)$, and $D(t)$ that satisfy the system of equations

$$(2.1) \quad \begin{aligned} D &= U^*MV, \quad U^*U = V^*V = I, \\ \mathcal{P}_S(D) &= 0, \quad \mathcal{P}_H(D)_{ij} = 0 \quad (i < j) \end{aligned}$$

for t in a neighborhood of 0. Due to the trivial nonuniqueness of the SVD, this system is underdetermined; it is convenient to fix the representative of the equivalence class by adding the condition that the matrix $U(0)^*U(t)$ has elements along the main diagonal that are positive reals.

There are unitary matrices $U(0)$, $V(0)$ such that $U(0)^*M(0)V(0) = D_0$, where $D_0 \equiv \text{diag}(d_1, \dots, d_n)$ and the alignment conditions of Lemma 2 hold for $X = \dot{M}(0)$. We will construct solutions to (2.1) by restating these equations in terms of new variables $\bar{U}(t)$, $\bar{V}(t)$, and $\bar{D}(t)$, connected to U , V , and D via the relations

$$(2.2) \quad U(t) = U(0) (I + t\bar{U}(t)), \quad V(t) = V(0) (I + t\bar{V}(t)), \quad D(t) = D_0 + t\bar{D}(t).$$

If we define $\bar{M}(t)$, $P(t)$ by the relations $M = M(0) + t\bar{M}$, $\bar{M}(0) = \dot{M}(0)$, and $P = U(0)^*\bar{M}V(0)$, then $U(0)^*MV(0) = D_0 + tP$ and the matrix $P(0) = U(0)^*\dot{M}(0)V(0)$ satisfies the alignment conditions of Lemma 2. These conditions can be restated as follows. Let $T(t) = \mathcal{P}_H(P(t))$ and $S(t) = \mathcal{P}_S(P(t))$. Then for $i < j$, $d_i = d_j = d$, we have $T_{ij}(0) = 0$ and $T_{ii}(0) > T_{jj}(0)$ in the case $d \neq 0$, and $T_{ij}(0) = S_{ij}(0) = 0$ and $T_{ii}(0) > T_{jj}(0) \geq 0$ in the case $d = 0$. Additionally, if $d_i = 0$, then $S_{ii} = 0$. Since M is twice differentiable at $t = 0$, we see that for these (i, j) , $L_{ij}(t) \equiv t^{-1}T_{ij}(t)$ can be defined at $t = 0$ so as to be a continuous function; if $d = 0$, then $K_{ij}(t) \equiv t^{-1}S_{ij}(t)$ can also be defined at $t = 0$ and this includes the case $i = j$. In fact, $2L_{ij}(0)$ and $2K_{ij}(0)$ are the ij components of the Hermitian and skew-Hermitian parts of $U(0)^*\dot{M}(0)V(0)$.

The following lemma is proved in section 4.

LEMMA 4. *For $t \neq 0$ the following system of equations in variables $\bar{U}$, $\bar{V}$ is equivalent to equations (2.1) via the relations (2.2):*

$$(L4a) \quad \bar{D}_{ii} = T_{ii} - t(T(D_0 - d_i I)^+ T + S(D_0 + d_i I)^+ S)_{ii} + O(t^2),$$

$$(L4b) \quad \bar{U}_{ii} = 0 + O(t),$$

$$(L4c) \quad (\bar{U} + \bar{V})_{ij} = -2T_{ij}/(d_i - d_j) + O(t) \quad (d_i \neq d_j),$$

$$(L4d) \quad (\bar{V} - \bar{U})_{ij} = -2S_{ij}/(d_i + d_j) + O(t) \quad (d_i + d_j \neq 0),$$

$$(L4e) \quad (T_{ii} - T_{jj})(\bar{U} + \bar{V})_{ij} = -2L_{ij} + 2(T(D_0 - dI)^+ T + S(D_0 + dI)^+ S)_{ij} + O(t) \\ (i \neq j; d_i = d_j = d),$$

$$(L4f) \quad (T_{ii} + T_{jj})(\bar{V} - \bar{U})_{ij} = -2K_{ij} + 2(SD_0^+ T + TD_0^+ S)_{ij} + O(t) \quad (d_i = d_j = 0).$$

The symbol $O(t^p)$ represents the product of t^p and some polynomial function of the elements of $\bar{U}$, $\bar{V}$, D_0 , and $\bar{M}$; A^+ denotes the Moore–Penrose pseudoinverse of a matrix A .

Since the quantities $T_{ii} - T_{jj}$ and $T_{ii} + T_{jj}$ are nonzero at $t = 0$ in equations (L4e) and (L4f), the system of equations in Lemma 4 can be solved for $\bar{U}$ and $\bar{V}$ at $t = 0$. Since the components of $\bar{U}$ and $\bar{V}$ appear only on the left-hand sides, it is clear that this solution is regular and the implicit function theorem implies that the solution at $t = 0$ may be extended to a continuous solution over a neighborhood of 0. Equations (L4a) also show that $\bar{D}(t)$ is differentiable at $t = 0$. Relations (2.2) then yield a solution to equations (2.1) valid on the same neighborhood; the same relations show that this solution is differentiable at $t = 0$ and that $D(t)$ is twice differentiable there. Finally, for each $i < j$ with $d_i = d_j$, we have $\dot{d}_i(0) - \dot{d}_j(0) = T_{ii} - T_{jj} > 0$ so that $d_i(t) > d_j(t)$ for small positive t . $\square$

COROLLARY 5. *In addition to the other hypotheses of Theorem 3, let $M(t)$ be C^k , $k \geq 2$. Then $D(t)$ is C^k and $U(t)$, $V(t)$ are C^{k-1} . If for each $i \neq j$ for which $d_i = d_j$ the function $P_{ij}(t)$ is C^k , then $U(t)$ and $V(t)$ are C^k as well.*

Proof. If $M(t)$ is k times continuously differentiable at $t = 0$, then $\bar{M}$ and hence P , T , and S are all C^{k-1} . It follows that all the right-hand sides of equations (L4a)–(L4e) are C^{k-1} functions except for the terms L_{ij} and K_{ij} , which are C^{k-2} . If P_{ij} is C^k for each $i \neq j$, $d_i = d_j$, then L_{ij} and K_{ij} are in fact C^{k-1} . The conclusion follows from relations (2.2). $\square$

Remark 1. If the terms L_{ij} and K_{ij} happen to be C^{k-1} , then U and V will have the same smoothness as M . These terms are related to the scalar functions $u^*(M(t) - M(0))v$, where u and v are left and right singular vectors corresponding to a repeated singular value of $M(0)$; this is discussed further in section 3.1. The smoothness properties described in Corollary 5 are not shared by the SVD itself. For negative t the elements of $D(t)$ will be distinct but not in decreasing order, and hence the columns of U and V will be permuted relative to the SVD of M . If a diagonal element of $D(t)$ changes sign at $t = 0$, then either the left or right singular vector that corresponds to it must change sign in the SVD. For paths $M(t)$ that are not transversal to the critical set, the situation requires a more delicate treatment. For example, $M(t)$ may lie entirely within the critical set; if $M(t)$ is analytic, some of the singular subspaces may have dimension greater than one and the basis vectors must be carefully chosen in order to preserve the analytic nature of the SVD [2].

Remark 2. Since by (2.2) $\dot{U}(0) = U(0)\bar{U}(0)$ and $\dot{V}(0) = V(0)\bar{V}(0)$, the equations for the barred quantities in Lemma 4 can be rewritten as differential equations which the solutions $U(t)$, $V(t)$ satisfy at $t = 0$. For paths not intersecting the critical set, these equations are well known [3, 13]. For analytic paths passing through the critical set in the real case, Wright [13] indicated how these derivatives could be found in the case of k coalescing singular values, $k = 2$ or 3 . The expressions on the right-hand sides of equations (L4e), (L4f) give these derivatives explicitly, without the limitation on k . For example, if $M(t)$ is Hermitian and $U(0) = V(0)$, and if we add a multiple of the identity to M if necessary so as to make all eigenvalues positive, the well-known relations $(U(0)^*\dot{U}(0))_{ij} = -T_{ij}/(d_i - d_j)$, where $d_i \neq d_j$, are supplemented by the relations

$$\left(U(0)^*\dot{U}(0) \right)_{ij} = (T_{ii} - T_{jj})^{-1} \left(-(1/2)U(0)^*\ddot{M}(0)U(0) + T(D_0 - dI)^+T \right)_{ij}$$

for those i, j where $d_i = d_j = d$. For linear paths with zero second derivative, the formulas of Lemma 4 may be thought of as providing differential-like information

about the evolution of the singular vectors along the trajectory, although obviously there is no differential at the critical point. The second derivatives of the eigenvalues can be read off directly from equation (L4a) of the lemma; this generalizes a second derivative formula for simple eigenvalues in [9].

In [2] it was shown that if $M(t)$ is C^k and has singular values that are nonzero and distinct for all t , so that the path does not intersect $\mathcal{S}$, then a C^k SVD of $M(t)$ exists. Theorem 3 allows us to extend this decomposition to paths that cross $\mathcal{S}$ transversely, with the proviso that negative singular values be allowed and that these values may appear along the diagonal in other than nonincreasing order. A similar result was proved in [5]; the hypotheses there differ in that zero cannot be a repeated singular value (i.e., the rank of M decreases by at most one at the critical point) and that the order of contact of the singular values be finite. The order-of-contact hypothesis is weaker than the statement below—for a regular path, transversal intersection of the path with $\mathcal{S}$ implies order of contact 1.

COROLLARY 6. *Let $M(t)$ be a C^k path, $a < t < b$, such that whenever $M(t) \in \mathcal{S}$, the matrix $\dot{M}(t)$ is transversal to $\mathcal{S}$. Then there exist C^{k-1} functions $U(t)$, $V(t)$ and a C^k function $D(t)$ such that for all t , U and V are unitary, D is real and diagonal, and $M = UDV^*$.*

Proof. It suffices to prove that if the decomposition exists for $t \in (a, t_1)$, with $M(t_1) \in \mathcal{S}$, then the decomposition can be extended to $t \in (a, t_2)$ with $t_2 > t_1$. Theorem 3 shows the existence of $\tilde{U}(t)\tilde{D}(t)\tilde{V}(t)^*$ equal to $M(t)$ for $|t - t_1| < \epsilon$. Let $t_* = t_1 - \epsilon/2$. Since $M(t_*)$ is regular, the columns of $U(t_*)$, $V(t_*)$ and $\tilde{U}(t_*)$, $\tilde{V}(t_*)$ and the diagonal elements of $D(t_*)$, $\tilde{D}(t_*)$ may be permuted to form proper SVDs of $M(t_*)$; i.e., the singular values are in strictly decreasing order. These SVDs are equivalent; thus $\tilde{U}(t_*)$, $\tilde{D}(t_*)$, $\tilde{V}(t_*)$ can be reordered so that the matrices $D_U = U(t_*)^*\tilde{U}(t_*)$ and $D_V = V(t_*)^*\tilde{V}(t_*)$ are diagonal. Then $\tilde{U}(t)D_U^{-1}$ and $\tilde{V}(t)D_V^{-1}$ can be used on $[t_*, t_1 + \epsilon)$ to continuously extend the factorization to $t < t_1 + \epsilon$. By using a C^∞ interpolation through diagonal unitary matrices from the identity to D_U and D_V , the extension will have the desired degree of differentiability. $\square$

Next we extend Theorem 3 to nonsquare matrices. The SVD of a matrix $M \in M_{m,n}$, $m < n$, consists of unitary matrices $U \in M_{m,m}$, $V \in M_{n,n}$ and reals $d_1 \geq \dots \geq d_m \geq 0$ such that $(U^*MV)_{ij} = d_i\delta_{ij}$. If k singular values are nonzero, then the last $n - k$ columns of V span the nullspace of M . Two SVDs UDV^* , $\tilde{U}\tilde{D}\tilde{V}^*$ of M will be called equivalent if $U^*\tilde{U}$ is diagonal and equal to the upper left $m \times m$ block of $V^*\tilde{V}$. A nonsquare matrix is a critical point of the SVD if any singular value has multiplicity greater than one (as in the case of square matrices) or if zero is a singular value. To see this, suppose that v is any vector in the nullspace of the matrix M and that $d_m = 0$. Then v is clearly a singular vector for the matrix $M + \epsilon u_m v^*$ and appears as a column in its SVD for $\epsilon > 0$; since the nullspace of M is at least two-dimensional, the SVD is discontinuous at M . The criterion for transversality must be modified to take this into account. The following definition adds the words “and positive” to Definition 1, and Lemma 8 below has a corresponding change in conditions (L8b). This lemma may be proved using the same argument as in the proof of Lemma 2.

DEFINITION 7. *An $m \times n$ matrix X , $m < n$, is transversal to $\mathcal{S}$ at M if for every d in $\{d_1, \dots, d_m\}$, either $\mathcal{P}_H(U_d^*XV_d)$ has distinct eigenvalues (if $d > 0$), or the singular values of the matrix $U_d^*XV_d$ are distinct and positive (if $d = 0$).*

LEMMA 8. *Suppose X is transversal to $\mathcal{S}$ at M . Then there are unitary matrices U and V such that $U^*MV = \text{diag}(d_1, \dots, d_n)$, and the components P_{ij} of the matrix $P = U^*XV$ satisfy the following alignment conditions: for all d and for every pair*

$i < j$ such that $d_i = d_j = d$,

(L8a) if $d \neq 0$, then $P_{ij} = -(P_{ji})^*$ and $\operatorname{Re}(P_{ii}) > \operatorname{Re}(P_{jj})$;

(L8b) if $d = 0$, then $P_{ij} = P_{ji} = 0$ and $P_{ii} > P_{jj} > 0$.

Furthermore, U and V are unique up to equivalence and depend continuously on the elements of X .

The next theorem is more general than Theorem 2.4 of [3], where the rank of M is assumed constant; here, an arbitrary number of singular values may go to zero at the critical point. However, the assumption of transversality is needed.

THEOREM 9. *Suppose $M(t)$ is a continuous complex $m \times n$ matrix-valued function of a real parameter t , twice differentiable at $t = 0$. If $\dot{M}(0)$ is transversal to $\mathcal{S}$ at $M(0)$, then there are continuous unitary matrix-valued functions $U(t)$, $V(t)$ and real-valued functions $d_1(t), \dots, d_m(t)$ defined for t in a neighborhood of 0, satisfying $d_k(0) = d_k$ for $1 \leq k \leq m$, $d_1(t) > \dots > d_m(t) > 0$ for $t > 0$, and*

$$U^*(t)M(t)V(t) = [\operatorname{diag}(d_1(t), \dots, d_m(t)), 0_{m \times (n-m)}].$$

Furthermore, U and V are differentiable at $t = 0$.

Proof. Let M_e be the $n \times n$ matrix obtained by adding $r \equiv n - m$ rows of zeros to M , and let U_e be the $n \times n$ block-diagonal unitary matrix with blocks U and $I_{r \times r}$. Using the notation of the preceding proof, one finds that $\bar{U}_e$ is the block-diagonal matrix with blocks $\bar{U}$ and $0_{r \times r}$, and $\bar{M}_e$ is the matrix $\bar{M}$ with r rows of zeros added. To prove the theorem, it will suffice to solve the system of equations $U_e^* \bar{M}_e V = D$ for unitary U_e , V and real diagonal D in a way that preserves the form of U_e . Thus we must find a solution that keeps the $r \times r$ block in the lower right corner of U_e constant; it suffices to satisfy the equations $(\bar{U}_e)_{ij}(t) = 0$ for $m < i, j \leq n$.

The algebraic manipulations yielding Lemma 4 can be repeated to find a regular system of equations for $\bar{U}_{ij}$, $i, j \leq m$, and $\bar{V}_{ij}$, $i \leq m$ and $j \leq n$, as long as the following conditions on the Hermitian and skew-Hermitian parts of $P_e(0)$ hold:

- (a) if $i < j \leq m$ and $d_i = d_j \neq 0$, then $T_{ij} = 0$ and $T_{ii} > T_{jj}$;
- (b) if $i < j \leq m$ and $d_i = d_j = 0$, then $T_{ij} = S_{ij} = 0$ and $T_{ii} > T_{jj} > 0$;
- (c) if $i \leq m < j \leq n$ and $d_i = 0$, then $T_{ij} = S_{ij} = 0$ and $T_{ii} > T_{jj} \geq 0$.

Because $\dot{M}(0)$ is transversal to $\mathcal{S}$ at $M(0)$, these conditions can be satisfied by proper choice of the matrices $U(0)$ and $V(0)$ (Lemma 8). Conditions (a) and (b) are verified in exactly the same way. Condition (c) can be satisfied because the singular values of $U(0)\dot{M}(0)V(0)$ are positive, and hence for $i \leq m < j \leq n$, $d_i = 0$, we have $T_{ii} > 0$ and $T_{jj} = 0$.

It is now clear that the system of equations taken from Lemma 4 for $(\bar{U}_e \pm \bar{V})_{ij}$ for $i, j \leq m$ and $i \leq m < j$, along with $(\bar{U}_e)_{ij}(t) = 0$, $m < i, j$, has a solution at $t = 0$ that is regular and can therefore be continued to a neighborhood of 0. $\square$

3. Applications.

3.1. Structure of the SVD near the critical set; relation to the $\sin(\theta)$ theorem. The set of matrix triples (U, V, M) in $M_{m,m} \times M_{n,n} \times M_{m,n}$ corresponding to the SVD of M , i.e., U, V unitary and U^*MV real diagonal, is the zero set of a system of polynomial equations in the real and imaginary parts of the elements of these matrices—equations (2.1)—and is thus an affine variety $\mathbf{V}$ over the real numbers. The lemma below is proved in the next section.

LEMMA 10. *The variety $\mathbf{V}$ is smooth; that is, the tangent space of $\mathbf{V}$ has the same dimension at every point.*

Let π be the projection taking the triple (U, V, M) onto M . The fiber in $\mathbf{V}$ over a given matrix M not in $\mathcal{S}$, $\pi^{-1}(M) \cap \mathbf{V}$, consists of a single equivalence class for each permutation of the columns of U and V , as discussed in section 2; one equivalence class will correspond to the SVD of M . Lemma 2 shows that at a critical point $M_s \in \mathcal{S}$ there is a smooth map p taking a direction X which is transversal to the singular set to a point $p(X)$ on the fiber over M_s , unique up to equivalence class, at which the matrices U, V satisfy the alignment conditions and the diagonal elements of $U^* M_s V$ appear in nonincreasing order. Theorem 3 shows that a path from M_s in the direction X can be lifted (locally) to a path in $\mathbf{V}$ from $p(X)$, and this lift corresponds to the SVD along the original path. Corollary 5 describes the smoothness of this SVD. In geometric terms, smoothness of $\mathbf{V}$ (Lemma 10) and the smooth dependence of the SVD on the direction X from the critical set mean that the singularity can be eliminated by blowing up at the critical set. The loss of smoothness noted in Corollary 5 therefore has a simple cause: if the path $M(t)$ is C^k and $\dot{M}(0)$ is nonzero, then the path that takes t to the direction of $M(t) - M(t_0)$ in projective space, i.e., the equivalence class of nonzero multiples of $M(t) - M(t_0)$, is only C^{k-1} in general. If the path in projective space is as smooth as the original path, then so will be the SVD. For example, consider the C^2 paths

$$M_1(t) = \begin{bmatrix} t & t^{7/3} \\ 0 & 0 \end{bmatrix}, \quad M_2(t) = \begin{bmatrix} t^{7/3} & t^{7/3} \\ 0 & 0 \end{bmatrix},$$

which pass through the critical set at $t = 0$. Note that the nonzero row of $M_1(t) - M_1(0)$ is a multiple of $(1, t^{4/3})$, so the corresponding path in projective space is only C^1 and there is a loss of smoothness in the SVD. There is no such loss for the SVD of M_2 because this path lies along a straight line; the path in projective space is constant.

The function p allows us to describe the SVD of most matrices near a given $M_s \in \mathcal{S}$. For matrices M within an ϵ -neighborhood of M_s , for which $M - M_s$ is transversal to the critical set, the singular vectors of M will be determined to order 1 by the mapping p ; this can be refined to order ϵ by using the perturbation equations of Lemma 4. The mapping p is described by the argument proving Lemma 2. For simplicity, suppose M is a square matrix with no repeated singular values that lies in an ϵ -neighborhood of M_s , d is a nonzero repeated singular value of M_s , U_0 and V_0 are unitary matrices such that (U_0, V_0, M_s) lies on the fiber over M_s , and U_d, V_d are the blocks corresponding to d . The unitary matrices U and V that are aligned with the direction $M - M_s$ will have blocks $U_d Q_d, V_d Q_d$ where unitary Q_d diagonalizes $\mathcal{P}_H(U_d^*(M - M_s)V_d)$ or, equivalently, $\mathcal{P}_H(U_d^* M V_d)$, since $U_d^* M_s V_d$ is a multiple of the identity. Then $p(M - M_s) = (U, V, M_s)$, and the singular vectors of M are approximated by the columns of U and V with error of order ϵ . The $O(\epsilon)$ term of the perturbation expansion can be obtained from $\bar{U}(0)$ and $\bar{V}(0)$ by using (L4e), (L4f) with $L_{ij} = K_{ij} = 0$ (since we may assume a linear path $M(t)$).

As a concrete example, consider matrices of real numbers and suppose d is a singular value of M_s of multiplicity two. Let a_{ij} be the components of the 2×2 matrix $U_d^T M V_d$. The direction from M_s to M is transversal to the critical set if this matrix is not a multiple of the identity. An elementary computation shows that the columns of Q are $(\cos(\theta), \sin(\theta))$ and $(-\sin(\theta), \cos(\theta))$, where θ satisfies the relation $\tan(2\theta) = (a_{12} + a_{21})/(a_{11} - a_{22})$.

At this point it is possible to describe the relationship between Theorem 1 and the $\sin(\theta)$ theorem [4, 12], which bounds the rotation of eigenvectors or singular vectors

of a matrix M by a perturbation ΔM , in terms of the magnitude of the perturbation and the reciprocal of the smallest gap between eigenvalues or singular values. There are two cases of interest. In the first case, the matrix M has a repeated singular value and the $\sin(\theta)$ theorem does not apply. Theorem 3 implies that the singular vectors of $M + \Delta M$ will differ from the columns of U and V of a SVD of M by $O(\|\Delta M\|)$ if each $U_d^*(\Delta M)V_d$ is skew-Hermitian off the diagonal. The second case involves a matrix M with a small gap between some singular values, such that SVD is sensitive to perturbations in M and the bound provided by the $\sin(\theta)$ theorem is large. Theorem 3 can be best applied by first constructing a matrix M_s near M that has repeated singular values and the same singular vectors as M . For example, if d_1 and d_2 are singular values of M , $d_1 - d_2 = \epsilon$, then $M_s = M + \epsilon u_2 v_2^*$ is a critical point of the SVD with the same singular vectors as M . The same reasoning used in the preceding paragraph allows one to estimate the singular vectors of $M + \Delta M$ using $p(\Delta M - M_s)$. In the case of real matrices the same simple formula applies: the singular vectors of $M + \Delta M$ corresponding to the first two singular values will differ from the singular vectors of M by a rotation θ that satisfies, to order $\|\Delta M\|$,

$$\tan(2\theta) = (u_1^* \Delta M v_2 + u_2^* \Delta M v_1) / (u_1^* \Delta M v_1 - u_2^* \Delta M v_2).$$

Clearly, the advantage in this approach is greatest if $M_s - M$ is small compared to ΔM .

3.2. Perturbation bounds from derivative estimates. A number of perturbation inequalities involving matrices A , $A + \Delta A$ have been obtained by estimating a derivative along a path between the matrices [1, 10]. Since Theorem 3 establishes the existence of smooth U , V , and D that very nearly constitute the SVD along such a path—even when passing through critical points of the SVD—this technique can be applied even to paths that pass transversely through the critical set.

As an example, consider the *polar decomposition* of a square matrix. Using the SVD, it is easy to express a square matrix M as the product of a unitary matrix and a positive semidefinite one:

$$M = (UV^*)(VDV^*) := \mathcal{U}(M)|M|.$$

If M is nonsingular, this factorization is independent of the particular choice of U , V even when there are repeated singular values; thus given a parametrization $M(t)$ that has a derivative transversal to the critical set, we may choose U , V that are aligned with $\dot{M}$ so that the derivatives $U^* \dot{U}$ and $V^* \dot{V}$ may be computed from the formulas in Lemma 4. For example, using the notation of Theorem 3,

$$(3.1) \quad U^* \frac{d}{dt}(UV^*)V = U^* \dot{U} - V^* \dot{V} = \bar{U} - \bar{V},$$

and the components of the right-hand side can be read off from equations (L4d). Mathias [10] and Bhatia [1] used bounds on this derivative to derive perturbation bounds for $\mathcal{U}(M)$, so long as the perturbation is small enough to avoid singular matrices. By supplementing equations (L4d) with equations (L4f), one can obtain analogous perturbation inequalities for perturbations along a path $M(t)$ that passes through singular points; unfortunately, the change of sign of a singular vector of the SVD that was mentioned in section 2 means that $\mathcal{U}(M(t)) \neq U(t)V^*(t)$ for t on one side or the other of the value for which $M(t)$ is singular. Specifically, $\mathcal{U}(M(t)) = U(t)\Delta(t)V(t)^*$,

where $\Delta(t)$ is a diagonal matrix with diagonal elements ± 1 chosen so that the diagonal elements of $\Delta(t)D(t)$ are nonnegative; $\Delta(t)$ may be discontinuous when $M(t)$ is singular.

The matrix absolute value $|M(t)|$ does not have a discontinuity as $M(t)$ passes through a singular point. It is known that the matrix absolute value is Lipschitz continuous in the Frobenius norm

$$(3.2) \quad \| |A| - |B| \|_F \leq c \|A - B\|_F$$

with constant $c = \sqrt{2}$ for the general case and $c = 1$ if A, B are normal. An intermediate form of this inequality can be deduced using derivative estimates based on Lemma 4 along the linear path from A to B .

COROLLARY 11. *Let A, B be square complex matrices and let $d_1(t), d_n(t)$ denote the largest and smallest singular values of $(1-t)A + tB$, respectively. If $\omega(t) = (d_1(t) - d_n(t))/(d_1(t) + d_n(t))$ and $\kappa = \max\{\omega(t), 0 \leq t \leq 1\}$, then $0 \leq \kappa \leq 1$ and*

$$(3.3) \quad \| |A| - |B| \|_F \leq (1 + \kappa^2)^{1/2} \|A - B\|_F.$$

Proof. Let $M(t) = A + t(B - A)$ be the linear path from A to B . Since M is analytic in t , an analytic SVD exists in the sense of [2]. If $M(t)$ is singular for any $t \in [0, 1]$, then $\kappa = 1$ and (3.3) reduces to (3.2); therefore, assume $M(t)$ is nonsingular.

To begin we compute the derivative of $|M|$. If differentiation by t is indicated by either a dot or a prime (for typographical convenience), we have

$$V^* (VDV^*)' V = (V^* \dot{V})D + \dot{D} - D(V^* \dot{V}) = \dot{D} + [(V^* \dot{V}), D],$$

the last term being a Lie product. For those i, j for which $d_i = d_j$, the (i, j) element of the Lie product is zero and $\dot{D}_{ij} = T_{ij}$ by (L4a) in the case $i = j$ and by the alignment condition otherwise. For other i, j , the (i, j) element is easily found using (L4c), (L4d) to be $T_{ij} + S_{ij}(d_i - d_j)/(d_i + d_j)$. Let W be the matrix with elements $W_{ij} = (d_i - d_j)/(d_i + d_j)$ (with the understanding that $W_{ij} = 0$ if $d_i = d_j = 0$) and let $(S \circ W)_{ij} = S_{ij}W_{ij}$; then we have shown

$$(3.4) \quad V^* \frac{d}{dt} |M| V = T + S \circ W.$$

We now consider the simple inequality

$$(3.5) \quad \| |A| - |B| \|_F \leq \int_0^1 \| |M|' \|_F dt = \int_0^1 \|T + S \circ W\|_F dt.$$

It is clear that ω is the largest element of W , such that $\|S \circ W\|_F \leq \omega \|S\|_F$. The integrand $\|T + S \circ W\|_F$ may be majorized as follows:

$$(3.6) \quad \|T + S \circ W\|_F \leq \|T\|_F + \|S \circ W\|_F \leq \|T\|_F + \omega \|S\|_F.$$

Since the norms of $\dot{M}$, T , and S satisfy the Pythagorean relation $\|\dot{M}\|_F^2 = \|T + S\|_F^2 = \|T\|_F^2 + \|S\|_F^2$, the right-hand side of (3.6) is at its maximum when $\|S\|_F = \omega \|T\|_F$. Therefore,

$$\|T\|_F + \omega \|S\|_F \leq \|\dot{M}\|_F (1 + \omega^2)^{1/2} \leq \|B - A\|_F (1 + \kappa^2)^{1/2},$$

and the proof is complete. $\square$

Note that if A, B are both Hermitian, then S is identically zero in (3.6) and we recover (3.2) with $c = 1$.

Inequality (3.3) may be useful when dealing with small perturbations of well-conditioned matrices that are non-Hermitian. For example, let $E = B - A$ and write $\sigma_k(A)$ for the k th singular value of A , etc. Using the fact that ω is an increasing function of d_1 and a decreasing function of d_n , and the estimates

$$\sigma_1(A + tE) \leq \sigma_1(A) + t\sigma_1(E),$$

$$\sigma_n(A + tE) \geq \sigma_n(A) - t\sigma_1(E),$$

it follows that

$$\omega(t) \leq \omega(0) + 2t \left(\frac{\sigma_1(E)}{\sigma_1(A) + \sigma_n(A)} \right).$$

One may substitute this into the derivative estimate (3.6) and integrate, or use the upper bound $\omega(0) + 2\sigma_1(E)/\sigma_1(A)$ as κ in (3.3).

The inequality (3.2) has a surprising extension in the generalized Araki–Yamagami inequality [1]:

$$(3.7) \quad \| |A| - |B| \|_F^2 + \| |A^*| - |B^*| \|_F^2 \leq 2\|A - B\|_F^2.$$

If A and B are nonsingular and $\mathcal{U}(A) = \mathcal{U}(B)$, then equality holds in (3.7). The converse can be proved by examining the derivatives of $|M|$ and $|M^*|$.

COROLLARY 12. *If A is nonsingular and equality holds in (3.7), then $B = \mathcal{U}(A)|B|$; if B is also nonsingular, then $\mathcal{U}(A) = \mathcal{U}(B)$.*

Proof. Let $M(t) = (1-t)A + tB$ so that $M(t) = U(t)D(t)V(t)^*$ with each factor analytic in t . Since $\det(M)$ is not identically zero, there is a finite set $J \subset [0, 1]$ of values of t for which $M(t)$ is singular. The polar factor $\mathcal{U}(M(t))$ may be discontinuous for $t \in J$; nonetheless, by changing the signs of columns of $U(t)$ as necessary, we may assert that $\mathcal{U}(M(t)) = U(t)V(t)^*$ on each interval in $[0, 1] - J$. By applying the Schwartz inequality to (3.5), we may write

$$(3.8) \quad \| |A| - |B| \|_F^2 \leq \int_{[0,1]-J} \| |M|' \|_F^2 dt = \int_{[0,1]-J} \| T + S \circ W \|_F^2 dt,$$

with equality only if $|M(t)|'$ is constant for $t \notin J$. Interchanging U and V in the derivation of (3.4) yields the formula $U^* |M^*|' U = T - S \circ W$, and hence

$$(3.9) \quad \| |A^*| - |B^*| \|_F^2 \leq \int_{[0,1]-J} \| T - S \circ W \|_F^2 dt.$$

Therefore the left-hand side of inequality (3.7) is majorized by

$$(3.10) \quad \begin{aligned} & \int_{[0,1]-J} (\| T + S \circ W \|_F^2 + \| T - S \circ W \|_F^2) dt \\ &= \int_{[0,1]-J} 2 (\| T \|_F^2 + \| S \circ W \|_F^2) dt. \end{aligned}$$

This integrand satisfies the inequality

$$(3.11) \quad \| T \|_F^2 + \| S \circ W \|_F^2 \leq \| T \|_F^2 + \| S \|_F^2 = \| \dot{M} \|_F^2 = \| A - B \|_F^2.$$

We now can see that equality is achieved in (3.7) only when it is achieved in (3.11) for all $t \notin J$. For M nonsingular, every element of W has absolute value less than 1 so that equality in (3.11) implies $S = 0$. It follows from (3.1) and (L4d) that $\mathcal{U}(M(t))$ is constant on each interval in $[0, 1] - J$. It also follows that $\|T \pm S \circ W\|_F = \|T\|_F = \|\dot{M}\|_F = \|B - A\|_F$ so that the right-hand sides of inequalities (3.8), (3.9) are equal to $\|A - B\|_F^2$. This being the case, equality in (3.7) implies that equality is also achieved in both (3.8) and (3.9), and hence $|M(t)|'$ is constant on $[0, 1] - J$. Since $\mathcal{U}(M)$ has zero derivative on this set, $|M|' = \mathcal{U}(M)^*(B - A)$. Finally, note that

$$\mathcal{U}(M(t))^* = \mathcal{U}(M(t))^* (M(t) - t(B - A)) A^{-1} = (|M(t)| - t|M(t)|') A^{-1}$$

has a continuous extension to $[0, 1]$. This extension has zero derivative on $[0, 1]$ and thus is constant; hence $B = \lim_{t \rightarrow 1} \mathcal{U}(M(t))|M(t)| = \mathcal{U}(A)|B|$. Clearly $\mathcal{U}(A) = \mathcal{U}(B)$ if B is nonsingular. $\square$

While the value $c = \sqrt{2}$ in inequality (3.2) cannot be reduced, neither can equality be attained except in the trivial case.

COROLLARY 13. *Suppose A is nonsingular and $\| |A| - |B| \|_F = \sqrt{2}\|A - B\|_F$. Then $A = B$.*

Proof. The hypothesis implies equality in (3.7). The preceding corollary shows that $|B| = \mathcal{U}(A)^*B$ and hence $\| |A| - |B| \|_F = \|A - B\|_F$. $\square$

4. Proofs of Lemmas 4 and 10. It will be convenient to use the following identities involving skew-Hermitian matrices S_1 and S_2 , the Lie product $[A, B] \equiv AB - BA$, and the Jordan product $\{A, B\} \equiv AB + BA$:

$$(I1) \quad \mathcal{P}_H(S_1^*X + XS_2) = \frac{1}{2}[\mathcal{P}_H(X), S_1 + S_2] + \frac{1}{2}\{\mathcal{P}_S(X), S_2 - S_1\},$$

$$(I2) \quad \mathcal{P}_S(S_1^*X + XS_2) = \frac{1}{2}[\mathcal{P}_S(X), S_1 + S_2] + \frac{1}{2}\{\mathcal{P}_H(X), S_2 - S_1\}.$$

We now proceed to find a system of equations equivalent to (2.1) for nonzero t . For the remainder of the proof, we take $t \neq 0$ unless otherwise noted. Substitution of relations (2.2) into the equations $U^*U = V^*V = I$ and division by t give

$$(4.1) \quad \bar{U} + \bar{U}^* + t\bar{U}^*\bar{U} = 0 = \bar{V} + \bar{V}^* + t\bar{V}^*\bar{V}.$$

Thus $\bar{U}$ differs from its skew-Hermitian part by a quantity that is $O(t)$; the same is true for $\bar{V}$. This shows that, to $O(t)$, the matrices $\bar{U}, \bar{V}$ may be used as S_1, S_2 in identities (I1) and (I2). The auxiliary condition $(U(0)^*U(t))_{ii} > 0$ is equivalent to the condition that $\bar{U}_{ii}$ is real for all i . Together with (4.1), we have (L4b), $\bar{U}(t)_{ii} = 0 + O(t)$.

The condition that D be real diagonal in (2.1) is equivalent to $\mathcal{P}_S(\bar{D}) = 0$, $\mathcal{P}_H(\bar{D})$ is diagonal. Substitution of (2.2) into the equation $D(t) = U^*(t)M(t)V(t)$ and division by t yield the equation

$$(4.2) \quad \bar{D}(t) = (\bar{U}^*D_0 + D_0\bar{V} + P) + t(\bar{U}^*P + P\bar{V} + \bar{U}^*D_0\bar{V}) + t^2\bar{U}^*P\bar{V}.$$

To compute $\mathcal{P}_H(\bar{D})$ and $\mathcal{P}_S(\bar{D})$, begin by applying (4.1) to find

$$\begin{aligned} 2\mathcal{P}_H(\bar{U}^*D_0 + D_0\bar{V}) &= [D_0, \bar{U} + \bar{V}] - t(\bar{U}^*\bar{U} + \bar{V}^*\bar{V})D_0, \\ 2\mathcal{P}_S(\bar{U}^*D_0 + D_0\bar{V}) &= \{D_0, \bar{V} - \bar{U}\} - t(\bar{U}^*\bar{U} - \bar{V}^*\bar{V})D_0. \end{aligned}$$

Identities (I1) and (I2) can be used to compute the projections of $\bar{U}^*P + P\bar{V}$ to order t . Thus we obtain

$$(4.3) \quad 2 \mathcal{P}_H(\bar{D}) = [D_0, \bar{U} + \bar{V}] + 2T \\ + t ([T, \bar{U} + \bar{V}] + \{S, \bar{V} - \bar{U}\} + (\bar{U}^2 + \bar{V}^2)D_0 - \bar{U}D_0\bar{V} - \bar{V}D_0\bar{U}) + O(t^2),$$

$$(4.4) \quad 2 \mathcal{P}_S(\bar{D}) = \{D_0, \bar{V} - \bar{U}\} + 2S \\ + t ([S, \bar{U} + \bar{V}] + \{T, \bar{V} - \bar{U}\} + (\bar{U}^2 - \bar{V}^2)D_0 - \bar{U}D_0\bar{V} + \bar{V}D_0\bar{U}) + O(t^2).$$

These expressions, together with the facts that $\mathcal{P}_H(\bar{D})$ is zero off the diagonal and $[D_0, \bar{U} + \bar{V}]$ is zero on the diagonal, imply

$$(4.5) \quad T = D(T) - \frac{1}{2}[D_0, \bar{U} + \bar{V}] + O(t),$$

$$(4.6) \quad S = -\frac{1}{2}\{D_0, \bar{V} - \bar{U}\} + O(t),$$

where $D(T) = \text{diag}(T_{11}, \dots, T_{nn})$. These relations will be helpful later.

The equations $\mathcal{P}_S(\bar{D})_{ij} = 0$ can now be written as

$$(4.7) \quad 0 = (d_i + d_j)(\bar{V} - \bar{U})_{ij} + 2S_{ij} + O(t),$$

yielding equations (L4d) for all indices i, j such that $d_i + d_j \neq 0$. For $i \neq j$ we have

$$(4.8) \quad 0 = \mathcal{P}_H(\bar{D})_{ij} = (d_i - d_j)(\bar{U} + \bar{V})_{ij} + 2T_{ij} + O(t),$$

yielding equations (L4c) if $d_i - d_j \neq 0$. (Note that $(\bar{U} + \bar{V})_{ii}$ can be found from $(\bar{V} - \bar{U})_{ii}$ since $\bar{U}_{ii} = 0 + O(t)$.)

If all the singular values d_i are distinct, (4.7) and (4.8) determine $\bar{U}$ and $\bar{V}$ at $t = 0$. However, if a singular value is repeated, then some of equations (4.8) fail to determine the corresponding elements of $\bar{U} + \bar{V}$; if the repeated singular value is zero, then a similar observation holds for equations (4.7). The remainder of the proof constructs replacements for these equations, leading to (L4e), (L4f).

First consider the case $d_i = d_j = d$. Recall the definition $L_{ij} \equiv T_{ij}/t$ for $t \neq 0$. Since $[D_0, \bar{U} + \bar{V}]_{ij} = 0$, the expression for $\mathcal{P}_H(\bar{D})_{ij}$ can be rewritten as

$$(4.9) \quad (2/t)\mathcal{P}_H(\bar{D})_{ij} = 2L_{ij} \\ + ([T, \bar{U} + \bar{V}] + \{S, \bar{V} - \bar{U}\} + (\bar{U}^2 + \bar{V}^2)D_0 - \bar{U}D_0\bar{V} - \bar{V}D_0\bar{U})_{ij} + O(t).$$

Since $d_i = d_j = d$, we may apply the identity

$$(I3) \quad d(\bar{U}^2 + \bar{V}^2) - \bar{U}D_0\bar{V} - \bar{V}D_0\bar{U} \\ = (1/2)((\bar{V} - \bar{U})(D_0 + dI)(\bar{V} - \bar{U}) - (\bar{U} + \bar{V})(D_0 - dI)(\bar{U} + \bar{V})).$$

Also, according to (4.5) and (4.6),

$$[T, \bar{U} + \bar{V}] = [D(T), \bar{U} + \bar{V}] - (1/2)[[D_0, \bar{U} + \bar{V}], \bar{U} + \bar{V}] + O(t), \\ \{S, \bar{V} - \bar{U}\} = -(1/2)\{\{D_0, \bar{V} - \bar{U}\}, \bar{V} - \bar{U}\} + O(t).$$

Since $[[A, B], B] = \{A, B^2\} - 2BAB$ and D_0 is diagonal, it is easily checked that

$$(-1/2)[[D_0, \bar{U} + \bar{V}], \bar{U} + \bar{V}]_{ij} = (-1/2)(d_i + d_j)((\bar{U} + \bar{V})^2)_{ij} + ((\bar{U} + \bar{V})D_0(\bar{U} + \bar{V}))_{ij} \\ = ((\bar{U} + \bar{V})(D_0 - dI)(\bar{U} + \bar{V}))_{ij}$$

and, similarly, that

$$(-1/2)\{\{D_0, \bar{V} - \bar{U}\}, \bar{V} - \bar{U}\}_{ij} = -((\bar{V} - \bar{U})(D_0 + dI)(\bar{V} - \bar{U}))_{ij},$$

which puts (4.9) into the form

$$\begin{aligned} (2/t)\mathcal{P}_H(\bar{D})_{ij} &= 2L_{ij} + (T_{ii} - T_{jj})(\bar{U} + \bar{V})_{ij} \\ &\quad + (1/2)((\bar{U} + \bar{V})(D_0 - dI)(\bar{U} + \bar{V}) \\ &\quad - (\bar{V} - \bar{U})(D_0 + dI)(\bar{V} - \bar{U}))_{ij} + O(t). \end{aligned}$$

At this point, (4.7), (4.8) may be brought to bear. Suppressing terms of order t ,

$$\begin{aligned} ((\bar{U} + \bar{V})(D_0 - dI)(\bar{U} + \bar{V}))_{ij} &= \sum_k (\bar{U} + \bar{V})_{ik}(d_k - d)(\bar{U} + \bar{V})_{kj} \\ &= 4 \sum_{d_k \neq d} \frac{T_{ik}(d_k - d)T_{kj}}{(d - d_k)(d_k - d)} \\ &= -4(T(D_0 - dI)^+T)_{ij}. \end{aligned}$$

Similarly,

$$((\bar{V} - \bar{U})(D_0 + dI)(\bar{V} - \bar{U}))_{ij} = 4(S(D_0 + dI)^+S)_{ij} + O(t).$$

This gives the final form for (4.9):

$$\begin{aligned} (2/t)\mathcal{P}_H(\bar{D})_{ij} &= 2L_{ij} + (\bar{U} + \bar{V})_{ij}(T_{ii} - T_{jj}) \\ &\quad - 2(T(D_0 - dI)^+T + S(D_0 + dI)^+S)_{ij} + O(t). \end{aligned}$$

In the case $i \neq j$, (L4e) follows immediately; when $i = j$, we obtain (L4a).

Next we examine the equation $\mathcal{P}_S(\bar{D})_{ij} = 0$, where $d_i = d_j = 0$ (including the case $i = j$). We know that $K_{ij}(t) \equiv S_{ij}/t$ has a continuous extension at $t = 0$. Since $\{D_0, \bar{U} + \bar{V}\}_{ij} = 0$, the equation $\mathcal{P}_S(\bar{D})_{ij} = 0$ can be written as

$$0 = 2K_{ij} + ([S, \bar{U} + \bar{V}] + \{T, \bar{V} - \bar{U}\} + (\bar{U}^2 - \bar{V}^2)D_0 - \bar{U}D_0\bar{V} + \bar{V}D_0\bar{U})_{ij} + O(t).$$

Next, $-\bar{U}D_0\bar{V} + \bar{V}D_0\bar{U}$ can be expressed in terms of $\bar{U} + \bar{V}$ and $\bar{V} - \bar{U}$ through a modification of identity (I3):

$$(I4) \quad -\bar{U}D_0\bar{V} + \bar{V}D_0\bar{U} = -\frac{1}{2}((\bar{U} + \bar{V})D_0(\bar{V} - \bar{U}) - (\bar{V} - \bar{U})D_0(\bar{U} + \bar{V})).$$

Since $d_i = d_j = 0$, both the quantities $\{[D_0, \bar{U} + \bar{V}], \bar{V} - \bar{U}\}_{ij}$ and $\{[D_0, \bar{V} - \bar{U}], \bar{U} + \bar{V}\}_{ij}$ are equal to

$$((\bar{V} - \bar{U})D_0(\bar{U} + \bar{V}) - (\bar{U} + \bar{V})D_0(\bar{V} - \bar{U}))_{ij}.$$

Thus the equation $\mathcal{P}_S(\bar{D})_{ij} = 0$ reaches the form

$$\begin{aligned} 0 &= 2K_{ij} + (T_{ii} + T_{jj})(\bar{V} - \bar{U})_{ij} \\ &\quad - \frac{1}{2}((\bar{V} - \bar{U})D_0(\bar{U} + \bar{V}) - (\bar{U} + \bar{V})D_0(\bar{V} - \bar{U}))_{ij} + O(t). \end{aligned}$$

Again using (L4c), (L4d), we obtain the expression

$$0 = 2K_{ij} + (T_{ii} + T_{jj})(\bar{V} - \bar{U})_{ij} - 2(SD_0^+T + TD_0^+S)_{ij} + O(t),$$

giving (L4f). This completes the proof of Lemma 4.

Now we prove Lemma 10. Given a point $(U, V, M) \in \mathbf{V}$ and arbitrary fixed $n \times n$ matrices $\bar{U}$, $\bar{V}$, and $\bar{M}$, the straight line L with parametrization $(U(I + t\bar{U}), V(I + t\bar{V}), M + t\bar{M})$ passes through $\mathbf{V}$ at $t = 0$. The polynomials defining $\mathbf{V}$, when evaluated along the line, are thus divisible by t . The line is tangent to $\mathbf{V}$ at (U, V, M) if the polynomials defining $\mathbf{V}$ evaluated on the line are divisible by t^2 . The tangent space of $\mathbf{V}$ at the given point is thus the zero set $(\bar{U}, \bar{V}, \bar{M})$ of a system of linear polynomials obtained by first evaluating the defining polynomials along the line L , then dividing by t , and finally setting t equal to zero. A short computation shows that the polynomials $U^*U - I$, $V^*V - I$ when subjected to this procedure yield the linear polynomials $\bar{U}^* + \bar{U}$, $\bar{V}^* + \bar{V}$, and hence L is tangent to $\mathbf{V}$ only if $\bar{U}$ and $\bar{V}$ are skew-Hermitian. Furthermore, since $\mathcal{P}_H(D) = D_0 + t\mathcal{P}_H(\bar{D})$ and $\mathcal{P}_S(D) = t\mathcal{P}_S(\bar{D})$, the remaining linear polynomials, $\mathcal{P}_H(\bar{D}(0))$ and $\mathcal{P}_S(\bar{D}(0))$, are found from (4.3) and (4.4):

$$\begin{aligned} T_{ij} + (1/2)(d_i - d_j)(\bar{U} + \bar{V})_{ij} & \quad (i < j), \\ S_{ij} + (1/2)(d_i + d_j)(\bar{V} - \bar{U})_{ij} & \quad (i \leq j). \end{aligned}$$

The dimension of the zero set of this system clearly is independent of the values of the d_i and therefore is the same at all points. $\square$

REFERENCES

- [1] R. BHATIA, *Matrix factorizations and their perturbations*, Linear Algebra Appl., 197/198 (1994), pp. 245–276.
- [2] A. BUNSE-GERSTNER, R. BYERS, V. MEHRMANN, AND N. K. NICHOLS, *Numerical computation of an analytic singular value decomposition of a matrix valued function*, Numer. Math., 60 (1991), pp. 1–39.
- [3] J.-L. CHERN AND L. DIECI, *Smoothness and periodicity of some matrix decompositions*, SIAM J. Matrix Anal. Appl., 22 (2000), pp. 772–792.
- [4] C. DAVIS AND W. M. KAHAN, *The rotation of eigenvectors by a perturbation. III*, SIAM J. Numer. Anal., 7 (1970), pp. 1–46.
- [5] L. DIECI AND T. EIROLA, *On smooth decompositions of matrices*, SIAM J. Matrix Anal. Appl., 20 (1999), pp. 800–819.
- [6] T. KOLLO AND H. NEUDECKER, *The derivative of an orthogonal matrix of eigenvectors of a symmetric matrix*, Linear Algebra Appl., 264 (1997), pp. 489–493.
- [7] P. LAX, *Linear Algebra*, Pure Appl. Math., John Wiley & Sons, New York, 1996.
- [8] A. MACIEJEWSKI AND CH. KLEIN, *The singular value decomposition: Computation and applications to robotics*, Internat. J. Robotics Res., 8 (1989), pp. 63–79.
- [9] J. MAGNUS AND H. NEUDECKER, *Matrix Differential Calculus with Applications in Statistics and Econometrics*, John Wiley & Sons, Chichester, UK, 1988.
- [10] R. MATHIAS, *Perturbation bounds for the polar decomposition*, SIAM J. Matrix Anal. Appl., 14 (1993), pp. 588–597.
- [11] F. RELICH, *Perturbation Theory of Eigenvalue Problems*, Gordon and Breach, NY, 1969.
- [12] G. W. STEWART AND J. SUN, *Matrix Perturbation Theory*, Academic Press, New York, 1990.
- [13] K. WRIGHT, *Differential equations for the analytic singular value decomposition*, Numer. Math., 63 (1992), pp. 72–88.

A RELATIONSHIP BETWEEN LINEAR DISCRIMINANT ANALYSIS AND THE GENERALIZED MINIMUM SQUARED ERROR SOLUTION*

CHEONG HEE PARK[†] AND HAESUN PARK[‡]

Abstract. In this paper, a relationship between linear discriminant analysis (LDA) and the generalized minimum squared error (MSE) solution is presented. The generalized MSE solution is shown to be equivalent to applying a certain classification rule in the space defined by LDA. The relationship between the MSE solution and Fisher discriminant analysis is extended to multiclass problems and also to undersampled problems for which the classical LDA is not applicable due to singularity of the scatter matrices. In addition, an efficient algorithm for LDA is proposed exploiting its relationship with the MSE procedure. Extensive experiments verify the theoretical results.

Key words. dimension reduction method, generalized singular value decomposition, linear discriminant analysis, minimum squared error solution, pseudoinverse, undersampled problems

AMS subject classifications. 15A09, 68T10, 62H30, 65F15, 15A18

DOI. 10.1137/040607599

1. Introduction. The extensive utilization of linear discriminant functions for pattern recognition is attributed to their simple concept and ease of computation. In linear discriminant function-based methods, the existence of a hyperplane which can optimally separate two classes is assumed. The optimality of the separating hyperplanes can be measured by various criteria, and numerous methods have been proposed originating from the work by Fisher [1]. In the Perceptron method, a linear discriminant function is obtained by iterative procedures to reduce the amount of misclassified training data [2]. Support vector machines (SVM) search for a linear function which maximizes the margin between classes either in the original data space or nonlinearly transformed feature spaces [3]. The minimum squared error (MSE) solution seeks a linear discriminant function that minimizes the squared error [4, 5]. Bayesian classifiers are based on estimation of distribution functions by which data is generated. From the estimated class density function, the class posterior probability is computed using Bayes theorem and a new data item is assigned to the class having the maximum posterior probability. In particular, assuming normal density functions for each class, where different means but a common covariance are to be estimated, Bayesian classification rules turn into a set of linear discriminant functions [6, 7].

While being similar in that linear functions are utilized, linear discriminant analysis (LDA) is different from the above-mentioned methods since LDA is a dimension reduction method rather than a discriminant classifier. LDA finds a linear transforma-

*Received by the editors April 29, 2004; accepted for publication (in revised form) by A. H. Sayed March 7, 2005; published electronically November 22, 2005. This material is based upon work supported in part by National Science Foundation grants CCR-0204109 and ACI-0305543. Any opinions, findings, and conclusions or recommendations expressed in this material are those of the authors and do not necessarily reflect the views of the National Science Foundation.

<http://www.siam.org/journals/simax/27-2/60759.html>

[†]Department of Computer Science and Engineering, Chungnam National University, 220 Gung-dong, Yuseong-gu, Daejeon 305-764, South Korea (cheonghee@cnu.ac.kr).

[‡]Department of Computer Science and Engineering, University of Minnesota, Minneapolis, MN 55455 (hpark@cs.umn.edu), and The National Science Foundation, 4201 Wilson Boulevard, Arlington, VA 22230. Research was performed while this author was at the National Science Foundation (NSF) and was partly supported by IR/D from the NSF.

tion that can maximize the class separability in the reduced dimensional space. The criterion used in LDA is to find a dimension reducing transformation that maximizes the between-class scatter and minimizes the within-class scatter [7]. When the data set has two classes, the relationship between the MSE solution and Fisher discriminant analysis (FDA) has been studied [8, 4], where FDA is a special case of LDA for two-class cases. Since both FDA and the MSE solution in two-class cases deal with *one linear function*, the relationship between them follows naturally. The MSE procedure is generalized for multiclass problems by setting up multiple two-class problems each of which is constructed by one class and the remaining data points forming the other class [4, 6]. In this paper, we develop the relationship between LDA and the generalized MSE procedure for multiclass problems and also for undersampled problems. Utilizing the developed relationships, it is shown that the MSE solution can be obtained by applying a certain classification rule in the reduced dimensional space obtained by LDA, and conversely LDA can be performed through the MSE procedure without solving the eigenvalue problem explicitly.

The term *LDA* has also been used to denote Bayesian linear classifiers resulting from the assumption of normal density functions with a common covariance. We note that in the rest of the paper LDA refers to a dimension reduction method. Many problems including generalization of Bayesian linear classifiers have been studied [9, 10], and under certain restrictions the relationships between Bayesian linear classifiers and LDA were investigated [11, 10]. While the relationships developed in this paper are applicable for both oversampled and undersampled problems, the results in [11, 10] are restricted to oversampled problems. For the singular or ill-conditioned covariance which occurs in undersampled problems, regularization methods can be applied for computation of the eigenvalue decomposition [12, 13]. However, the estimation of the regularization parameters can be expensive, and generalization errors by overfitting, especially in undersampled problems, should be taken care of.

The rest of the paper is organized as follows. In sections 2 and 3, LDA and the MSE procedures are reviewed. In section 4, we generalize the relation between the MSE solution and FDA for undersampled problems for which the classical FDA fails due to the singularity of the scatter matrices. We also derive the relationship between LDA and the generalized MSE solution for multiclass problems. In section 5, we propose an efficient algorithm for LDA which utilizes the relationship with the MSE solution and does not require the solution of eigenvalue problems. The experimental results in section 6 verify the theoretical results.

2. Linear discriminant analysis. LDA is a linear dimension reduction method which can be used as a preprocessing step for data analysis. Based on the information from the given data, LDA finds a linear transformation that maximizes the between-class distances and minimizes the within-class scatter so that the class separability can be optimized in the transformed space. Throughout the paper, we assume the vector space representation of a data set A ,

$$(2.1) \quad A = [a_1, \dots, a_n] = [A_1, A_2, \dots, A_r] \in \mathbb{R}^{m \times n},$$

where each data item in the m -dimensional space is represented as a column vector a_i and a collection of data items in the i th class as a block matrix $A_i \in \mathbb{R}^{m \times n_i}$. Each class i ($1 \leq i \leq r$) has n_i elements and the total number of data items is $n = \sum_{i=1}^r n_i$. Let N_i ($1 \leq i \leq r$) be the index set of data items in the class i . The data set A can be considered a training set on which the modeling of data analysis algorithms is

based—for example, searching for a linear transformation for LDA and discriminant functions for the MSE procedure.

Given a data set A , the between-class scatter matrix S_b , within-class scatter matrix S_w , and total scatter matrix S_t are defined as

$$(2.2) \quad S_b = \sum_{i=1}^r n_i (c_i - c)(c_i - c)^T, \quad S_w = \sum_{i=1}^r \sum_{j \in N_i} (a_j - c_i)(a_j - c_i)^T,$$

$$S_t = S_b + S_w = \sum_{j=1}^n (a_j - c)(a_j - c)^T,$$

where $c_i = \frac{1}{n_i} \sum_{j \in N_i} a_j$ and $c = \frac{1}{n} \sum_{j=1}^n a_j$ are class centroids and the global centroid, respectively. The *traces* of the scatter matrices can be used to measure the quality of the cluster structure in the data set as

$$\text{trace}(S_b) = \sum_{i=1}^r n_i \|c_i - c\|_2^2, \quad \text{trace}(S_w) = \sum_{i=1}^r \sum_{j \in N_i} \|a_j - c_i\|_2^2.$$

The distance between classes is quantified by $\text{trace}(S_b)$, and $\text{trace}(S_w)$ measures the scatter within classes. The optimal dimension reducing transformation $G^T \in \mathbb{R}^{l \times m}$ for LDA is the one that maximizes

$$(2.3) \quad J(G) = \text{trace}((G^T S_w G)^{-1} (G^T S_b G)),$$

where $G^T S_b G$ and $G^T S_w G$ are scatter matrices in the transformed space. It is well known [7] that the criterion in (2.3) is maximized when the columns of $G \in \mathbb{R}^{m \times (r-1)}$ are the eigenvectors x corresponding to the $r-1$ largest eigenvalue λ of

$$(2.4) \quad S_b x = \lambda S_w x.$$

When S_w is nonsingular, one can solve the eigenvalue problem

$$(2.5) \quad S_w^{-1} S_b x = \lambda x,$$

referred to as the classical LDA.

In order to overcome some limitations in the classical LDA, several generalization methods have been proposed. The problems caused by the singularity of the scatter matrices on undersampled problems are circumvented by two-stage decompositions of the scatter matrices [14, 15, 16], and the criterion itself of LDA is criticized in [17]. Howland et al. [18, 19] applied the generalized singular value decomposition (GSVD) due to Paige and Saunders [20] which is applicable for undersampled problems. We briefly review the method used in [18] and give a new approach to it, which will be used in deriving the relationship between LDA and the generalized MSE solution.

When the GSVD [20] is applied to two matrices Z_b and Z_w with the same number of columns, p , we have

$$(2.6) \quad U_b^T Z_b X = \begin{bmatrix} \Gamma_b & 0 \end{bmatrix} \quad \text{and} \quad U_w^T Z_w X = \begin{bmatrix} \Gamma_w & 0 \end{bmatrix} \quad \text{for } s = \text{rank} \left(\begin{bmatrix} Z_b \\ Z_w \end{bmatrix} \right),$$

$\underbrace{\hspace{1.5cm}}_s \quad \underbrace{\hspace{1.5cm}}_{p-s} \qquad \underbrace{\hspace{1.5cm}}_s \quad \underbrace{\hspace{1.5cm}}_{p-s}$

where U_b and U_w are orthogonal and X is nonsingular,

$$\Gamma_b^T \Gamma_b + \Gamma_w^T \Gamma_w = I_s,$$

and $\Gamma_b^T \Gamma_b$ and $\Gamma_w^T \Gamma_w$ are diagonal matrices with nonincreasing and nondecreasing diagonal components, respectively. The method due to Howland et al. [18] utilizes the fact that the scatter matrices can be expressed as

$$\begin{aligned} S_b &= H_b H_b^T, & H_b &= [\sqrt{n_1}(c_1 - c), \dots, \sqrt{n_r}(c_r - c)] \in \mathbb{R}^{m \times r}, \\ S_w &= H_w H_w^T, & H_w &= [A_1 - c_1 e_1^T, \dots, A_r - c_r e_r^T] \in \mathbb{R}^{m \times n}, \\ S_t &= H_t H_t^T, & H_t &= [a_1 - c, \dots, a_n - c] \in \mathbb{R}^{m \times n}, \end{aligned}$$

where $e_i = [1, \dots, 1]^T \in \mathbb{R}^{n_i \times 1}$. Suppose the GSVD is applied to the matrix pair (H_b^T, H_w^T) , and we obtain

$$(2.7) \quad U_b^T H_b^T X = [\Gamma_b \quad 0] \quad \text{and} \quad U_w^T H_w^T X = [\Gamma_w \quad 0].$$

From (2.7) and $\Gamma_b^T \Gamma_b + \Gamma_w^T \Gamma_w = I_s$,

$$(2.8) \quad X^T S_b X = \begin{bmatrix} \Gamma_b^T \Gamma_b & \\ & 0_{m-s} \end{bmatrix} \equiv \begin{bmatrix} I_\mu & & & \\ & D_\tau & & \\ & & 0_{s-\mu-\tau} & \\ & & & 0_{m-s} \end{bmatrix}$$

and

$$(2.9) \quad X^T S_w X = \begin{bmatrix} \Gamma_w^T \Gamma_w & \\ & 0_{m-s} \end{bmatrix} \equiv \begin{bmatrix} 0_\mu & & & \\ & E_\tau & & \\ & & I_{s-\mu-\tau} & \\ & & & 0_{m-s} \end{bmatrix},$$

where the subscripts on I and 0 denote the order of square identity and zero matrices. Denoting the diagonal elements in (2.8) as η_i and the diagonal elements in (2.9) as ζ_i , we have

$$(2.10) \quad \zeta_i S_b x_i = \eta_i S_w x_i, \quad i = 1, \dots, m,$$

where x_i are the column vectors of X . Note that x_i , $i = s+1, \dots, m$, belong to $\text{null}(S_b) \cap \text{null}(S_w)$ and therefore do not convey any discriminant information. Since

$$\eta_1 \geq \dots \geq \eta_s \quad \text{and} \quad \zeta_1 \leq \dots \leq \zeta_s,$$

the $r-1$ leftmost columns of X give an optimal transformation for LDA. This method is called LDA/GSVD [18, 19].

The algorithm to compute the GSVD for the pair (H_b^T, H_w^T) was presented in [18] as follows:

1. Compute the SVD of $Z = \begin{bmatrix} H_b^T \\ H_w^T \end{bmatrix} \in \mathbb{R}^{(r+n) \times m}$: $Z = P \begin{bmatrix} \Sigma & 0 \\ 0 & 0 \end{bmatrix} U^T$, where $s = \text{rank}(Z)$ and $P \in \mathbb{R}^{(r+n) \times (r+n)}$ and $U \in \mathbb{R}^{m \times m}$ are orthogonal and the diagonal components of $\Sigma \in \mathbb{R}^{s \times s}$ are nonincreasing.
2. Compute V from the SVD of $P(1:r, 1:s)$, which is $P(1:r, 1:s) = W \Gamma_b V^T$.
3. Compute the first $r-1$ columns of $X = U \begin{bmatrix} \Sigma^{-1} V & 0 \\ 0 & I \end{bmatrix}$, and assign them to the transformation matrix G .

Since $\Gamma_b^T \Gamma_b + \Gamma_w^T \Gamma_w = I_s$, from (2.8) and (2.9), we have

$$(2.11) \quad X^T S_t X = X^T S_b X + X^T S_w X = \begin{bmatrix} I_s & 0 \\ 0 & 0 \end{bmatrix},$$

where $s = \text{rank}(Z)$. Equation (2.11) implies $s = \text{rank}(S_t)$ and from step 3 in the LDA/GSVD algorithm

$$(2.12) \quad S_t = X^{-T} \begin{bmatrix} I_s & 0 \\ 0 & 0 \end{bmatrix} X^{-1} = U \begin{bmatrix} \Sigma^2 & 0 \\ 0 & 0 \end{bmatrix} U^T,$$

which results in the eigenvalue decomposition (EVD) of S_t . Partitioning U as

$$U = [\underbrace{U_1}_s \underbrace{U_2}_{m-s}],$$

we have

$$(2.13) \quad X = U \begin{bmatrix} \Sigma^{-1}V & 0 \\ 0 & I \end{bmatrix} = [U_1 \Sigma^{-1}V \quad U_2].$$

By substituting X in (2.8) with (2.13),

$$(2.14) \quad \Sigma^{-1}U_1^T S_b U_1 \Sigma^{-1} = V \Gamma_b^T \Gamma_b V^T.$$

Note that the optimal transformation matrix G by LDA/GSVD is obtained by the leftmost $r-1$ columns of X , which are the leftmost $r-1$ columns of $U_1 \Sigma^{-1}V$. Hence (2.12) and (2.14) show that the solution to LDA/GSVD can be obtained as follows:

1. Compute the EVD of S_t :

$$(2.15) \quad S_t = [U_1 \quad U_2] \begin{bmatrix} \Sigma^2 & 0 \\ 0 & 0 \end{bmatrix} \begin{bmatrix} U_1^T \\ U_2^T \end{bmatrix}.$$

2. Compute V from the EVD of $\tilde{S}_b \equiv \Sigma^{-1}U_1^T S_b U_1 \Sigma^{-1} : \tilde{S}_b = V \Gamma_b^T \Gamma_b V^T$.

3. Assign the first $r-1$ columns of $U_1 \Sigma^{-1}V$ to G .

In step 2 of the new approach, denoting

$$(2.16) \quad F = U_1 \Sigma^{-1} \in \mathbb{R}^{m \times s},$$

the EVD of $\tilde{S}_b$ can be computed from the SVD of $F^T H_b$ as

$$(2.17) \quad F^T H_b = V \Gamma_b^T S^T = [\underbrace{V_1}_{r-1} \underbrace{V_2}_{s-r+1}] \begin{bmatrix} \Gamma_{b1} & 0 \\ 0 & 0 \end{bmatrix} \begin{bmatrix} S_1^T \\ S_2^T \end{bmatrix},$$

where $V \in \mathbb{R}^{s \times s}$, $S \in \mathbb{R}^{r \times r}$ are orthogonal and $\Gamma_{b1} \in \mathbb{R}^{(r-1) \times (r-1)}$ is a diagonal matrix with nonincreasing diagonal elements. Hence

$$(2.18) \quad X = [U_1 \Sigma^{-1}V \quad U_2] = [FV_1 \quad FV_2 \quad U_2]$$

and the transformation matrix G is given as

$$(2.19) \quad G = FV_1 = U_1 \Sigma^{-1}V_1.$$

For any $x \in \text{null}(S_b) \cap \text{null}(S_w)$,

$$0 = x^T S_b x = (x^T H_b)(H_b^T x) = \|x^T H_b\|^2 = \sum_{i=1}^r n_i |x^T c_i - x^T c|^2$$

and

$$0 = x^T S_w x = \sum_{j=1}^n |x^T a_j - x^T c_i|^2, \quad \text{where } a_j \text{ belongs to the } i\text{th class.}$$

Hence

$$\begin{cases} x^T c_i = x^T c & \text{for } i = 1, \dots, r, \\ x^T a_j = x^T c_i & \text{for all } j \text{ in } N_i \text{ and } i = 1, \dots, r, \end{cases}$$

therefore,

$$(2.20) \quad U_2^T z = U_2^T c$$

for any given data item z . This implies that the vectors $x_i, i = s+1, \dots, m$, belonging to $\text{null}(S_b) \cap \text{null}(S_w)$ do not convey discriminative information among the classes, even though the corresponding eigenvalues are not necessarily zero. Since $\text{rank}(S_b) \leq r-1$, from (2.8) and (2.9)

$$x_i^T S_b x_i = 0 \quad \text{and} \quad x_i^T S_w x_i = 1 \quad \text{for } r \leq i \leq s,$$

and the between-class scatter becomes zero by the projection onto the vector x_i . Hence it is justifiable that the linear transformation G^T for LDA can be formed by taking the first $r-1$ columns from $[FV \ U_2] = [FV_1 \ FV_2 \ U_2]$.

3. Minimum squared error solution. The MSE solution in a two-class problem (i.e., $r = 2$) seeks a linear discriminant function

$$g(z) = w_0 + w^T z$$

for which

$$(3.1) \quad g(z) = w_0 + w^T z = \begin{cases} \beta_1 & \text{if } z \in \text{class 1,} \\ \beta_2 & \text{if } z \in \text{class 2,} \end{cases}$$

where β_i is the prespecified number for each class. For the data set A given in (2.1), the problem (3.1) can be reformulated to minimize the squared error

$$(3.2) \quad \left\| \begin{bmatrix} 1 & a_1^T \\ \vdots & \vdots \\ 1 & a_n^T \end{bmatrix} \begin{bmatrix} w_0 \\ w \end{bmatrix} - \begin{bmatrix} y_1 \\ \vdots \\ y_n \end{bmatrix} \right\|_2^2,$$

where $y_i = \beta_1$ if a_i is in class 1 and $y_i = \beta_2$ if a_i is in class 2. Denoting

$$(3.3) \quad \mathcal{P} = \begin{bmatrix} 1 & a_1^T \\ \vdots & \vdots \\ 1 & a_n^T \end{bmatrix},$$

a solution which minimizes the squared error (3.2) can be computed using the pseudo-inverse $\mathcal{P}^+$ of $\mathcal{P}$ as

$$(3.4) \quad \begin{bmatrix} w_0 \\ w \end{bmatrix} = \mathcal{P}^+ \begin{bmatrix} y_1 \\ \vdots \\ y_n \end{bmatrix}.$$

When the number of columns of P is greater than the number of rows, i.e., $m+1 > n$, the least squares problem of (3.2) is underdetermined and there may exist infinitely many solutions. The one given in (3.4) is one of many possible solutions.

Different choices of $\beta = [\beta_1, \beta_2]^T$ would give different discriminant functions. In particular, when $\beta_1 = n/n_1$ and $\beta_2 = -n/n_2$, the MSE solution is related to the FDA [4]. The vector w in (3.4) is the same as the solution x of FDA except for some scaling factor α as

$$(3.5) \quad w = \alpha S_w^{-1}(c_1 - c_2) \equiv \alpha x \quad \text{and} \quad w_0 = -w^T c,$$

where c and c_i are the global and class centroids, respectively. A new data item is assigned to class 1 if

$$(3.6) \quad w^T z + w_0 = w^T(z - c) = \alpha x^T(z - c) > 0;$$

otherwise it is assigned to class 2.

The MSE procedure is generalized to multiclass cases as a set of multiple two-class problems [4]. For each class i ($1 \leq i \leq r$), the MSE solution to the problem

$$(3.7) \quad g_i(z) = w_{0i} + w_i^T z = \begin{cases} \beta_i & \text{if } z \in \text{class } i, \\ 0 & \text{otherwise} \end{cases}$$

is to be found. The solution of the multiclass problem (3.7) in contrast to the problem (3.1) will be referred to as the *generalized* MSE solution whenever the distinction is needed. As in [4], one choice for β_i would be assigning $\beta_i = 1$ for $i = 1, \dots, r$. The squared error function in the multiclass problem is expressed using the Frobenius norm as

$$(3.8) \quad \left\| \begin{bmatrix} 1 & a_1^T \\ \vdots & \vdots \\ 1 & a_n^T \end{bmatrix} \begin{bmatrix} w_{01} & \cdots & w_{0r} \\ w_1 & \cdots & w_r \end{bmatrix} - \begin{bmatrix} y_{11} & \cdots & y_{1r} \\ \vdots & & \vdots \\ y_{n1} & \cdots & y_{nr} \end{bmatrix} \right\|_F^2,$$

where $y_{ji} = \beta_i$ if a_j belongs to the class i , and 0 otherwise. Denoting

$$(3.9) \quad \mathcal{W} = \begin{bmatrix} w_{01} & \cdots & w_{0r} \\ w_1 & \cdots & w_r \end{bmatrix} \quad \text{and} \quad \mathcal{Y} = \begin{bmatrix} y_{11} & \cdots & y_{1r} \\ \vdots & & \vdots \\ y_{n1} & \cdots & y_{nr} \end{bmatrix}$$

and with $\mathcal{P}$ defined as in (3.3), the MSE solution of the problem (3.8) can be obtained by

$$(3.10) \quad \mathcal{W} = \mathcal{P}^+ \mathcal{Y},$$

and a new data item z is assigned to the class i if, for all $j \neq i$,

$$(3.11) \quad g_i(z) > g_j(z).$$

Let us consider the mapping defined by the discriminant functions of the MSE solution (3.10) as

$$(3.12) \quad x \longrightarrow [g_1(x), \dots, g_r(x)]^T \in \mathbb{R}^{r \times 1}.$$

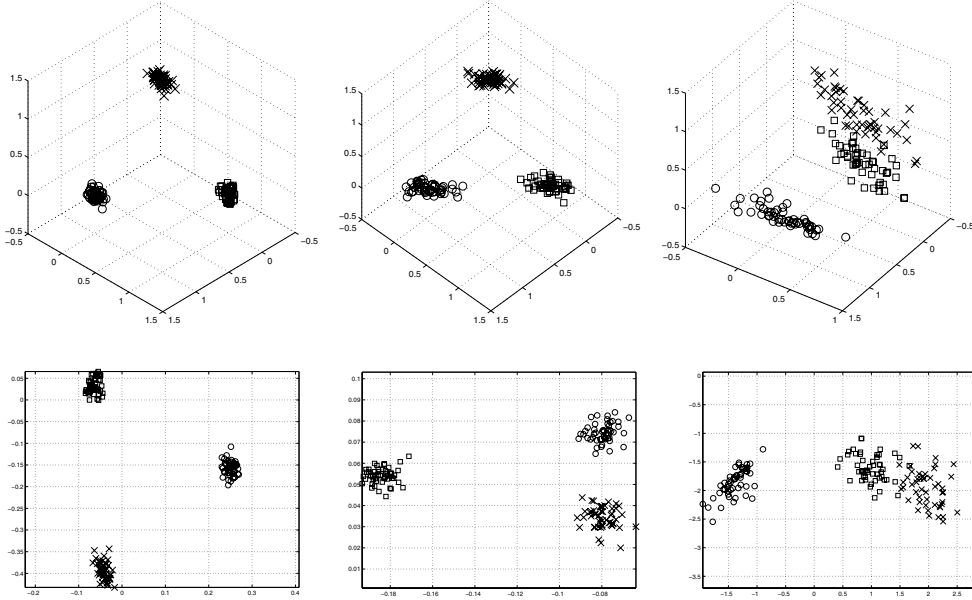


FIG. 1. The spaces transformed by the mapping (3.12) induced by the generalized MSE solution (first row) and LDA (second row).

Then (3.8) can be represented as

$$\|\mathcal{P}\mathcal{W} - \mathcal{Y}\|_F^2 = \sum_{1 \leq i \leq r} \sum_{j \in N_i} \left\| \begin{bmatrix} g_1(a_j) \\ \vdots \\ g_r(a_j) \end{bmatrix} - \beta_i b_i \right\|_2^2,$$

where $b_i \in \mathbb{R}^{r \times 1}$ ($1 \leq i \leq r$) is the column vector with 1 in the i th position and 0 elsewhere. Hence in the space transformed by the mapping (3.12), the i th class centroid will be mapped close to the point $\beta_i b_i$. Figure 1 illustrates the transformed spaces by LDA and the mapping (3.12), where β_i was set to 1, $1 \leq i \leq 3$, for a problem with three classes. The figures in the first row were obtained by the mapping (3.12) resulting in the dimension which is the same as the number of classes, while the figures in the second row show the reduced dimensional space by LDA for which the dimension is one less than the number of classes. The first two figures on the top were obtained by randomly taking three subclasses in the Isolet data set from the UCI Machine Learning Repository.¹ The Isolet data set has 26 classes, and a detailed explanation of the data set will be given in section 6. The third figure on the top, which was obtained by the Iris data set, illustrates that two classes among three classes are not well separable. The figures in the second row show the transformed space by LDA which corresponds to the figures on the top. The corresponding figures look quite similar.

What is the mathematical relationship between the two methods? If there is any relationship, is it possible to take advantage of the merits from each method and combine them? In the next section, we answer these questions by studying the relationship between LDA and the generalized MSE solution for multiclass problems.

¹<http://www.ics.uci.edu/~mllearn/MLRepository.html>

4. Relationships between LDA and the generalized MSE solution. The relationship of the MSE solution and FDA given in (3.5) holds when the within-class scatter matrix S_w is nonsingular. Now we show that the relationship (3.5) can be generalized for multiclass and undersampled problems by using the algorithm discussed in section 2.

4.1. FDA and the MSE solution on undersampled problems. Let

$$g(z) = w_0 + w^T z$$

be the MSE solution to the problem

$$(4.1) \quad g(z) = w_0 + w^T z = \begin{cases} n/n_1 & \text{if } z \in \text{class 1,} \\ -n/n_2 & \text{if } z \in \text{class 2.} \end{cases}$$

The normal equations for the problem in (3.2) are

$$(4.2) \quad \begin{bmatrix} 1 & \cdots & 1 \\ a_1 & \cdots & a_n \end{bmatrix} \begin{bmatrix} 1 \\ a_1^T \\ \vdots \\ a_n^T \\ 1 \end{bmatrix} \begin{bmatrix} w_0 \\ w \end{bmatrix} = \begin{bmatrix} 1 & \cdots & 1 \\ a_1 & \cdots & a_n \end{bmatrix} \begin{bmatrix} \frac{n}{n_1} e_{n_1} \\ -\frac{n}{n_2} e_{n_2} \end{bmatrix},$$

where e_{n_i} is the $n_i \times 1$ column vector with elements 1. From (4.2), we obtain

$$(4.3) \quad \begin{cases} nw_0 + nc^T w = 0, \\ ncw_0 + \left(\sum_{1 \leq j \leq n} a_j a_j^T \right) w = \frac{n}{n_1} \sum_{j \in N_1} a_j - \frac{n}{n_2} \sum_{j \in N_2} a_j. \end{cases}$$

From the first equation in (4.3) we have

$$(4.4) \quad w_0 = -c^T w.$$

By substituting (4.4) in the second equation of (4.3) and using the expressions of S_b and S_w for two-class problems

$$S_b = \sum_{1 \leq i \leq 2} n_i c_i c_i^T - n c c^T \quad \text{and} \quad S_w = \sum_{1 \leq j \leq n} a_j a_j^T - \sum_{1 \leq i \leq 2} n_i c_i c_i^T,$$

we obtain

$$(4.5) \quad (S_b + S_w)w = n(c_1 - c_2).$$

Let x_1 be the first column vector of $[FV_1 \ FV_2 \ U_2]$ in (2.18). From the discussion given in section 2 and the fact that $\text{rank}(S_b) = 1$, we have

$$\zeta_i S_b x_i = \eta_i S_w x_i \quad \text{for} \quad \eta_1 > \eta_2 = \cdots = \eta_m = 0.$$

Since $\eta_1 + \zeta_1 = 1$,

$$(4.6) \quad \eta_1 (S_b + S_w) x_1 = (\eta_1 + \zeta_1) S_b x_1 = S_b x_1 = \frac{n_1 n_2}{n} (c_1 - c_2)(c_1 - c_2)^T x_1.$$

Denoting

$$\mu = \eta_1 \frac{n^2}{n_1 n_2 (c_1 - c_2)^T x_1},$$

(4.6) becomes

$$(4.7) \quad (S_b + S_w)\mu x_1 = n(c_1 - c_2).$$

Then by (4.5) and (4.7), we have

$$(4.8) \quad S_t w = (S_b + S_w)w = (S_b + S_w)\mu x_1 = S_t \mu x_1.$$

From (4.8) and the EVD of S_t in (2.15),

$$U_1 \Sigma^2 U_1^T w = U_1 \Sigma^2 U_1^T \mu x_1 \quad \text{and} \quad U_1^T w = U_1^T \mu x_1,$$

and from (2.20) and (4.4) we obtain

$$(4.9) \quad \begin{aligned} w^T z + w_0 &= w^T(z - c) = w^T(U_1 U_1^T + U_2 U_2^T)(z - c) \\ &= w^T U_1 U_1^T(z - c) = \mu x_1^T U_1 U_1^T(z - c) \\ &= \mu x_1^T (U_1 U_1^T + U_2 U_2^T)(z - c) = \mu x_1^T(z - c). \end{aligned}$$

Equation (4.9) gives the relation between the MSE solution and the generalized solution of FDA, which holds regardless of the singularity of the scatter matrices.

While FDA gives a one-dimensional reduced representation and the MSE solution produces one discriminant function, the generalized MSE solution works with r linear discriminant functions and LDA gives an $(r - 1)$ -dimensional representation of the original data space. Now we show the relationship between LDA and the generalized MSE solution.

4.2. LDA and the generalized MSE solution. The generalized MSE solution to the problem

$$g_i(z) = w_{0i} + w_i^T z = \begin{cases} \beta_i & \text{if } z \in \text{class } i \\ 0 & \text{otherwise} \end{cases} \quad \text{for } i = 1, \dots, r$$

can be solved by the normal equation

$$(4.10) \quad \mathcal{P}^T \mathcal{P} \mathcal{W} = \mathcal{P}^T \mathcal{Y},$$

where $\mathcal{P}$, $\mathcal{Y}$, and $\mathcal{W}$ are defined in (3.3) and (3.9). From (4.10), we obtain

$$\begin{aligned} &\begin{bmatrix} n & \sum_{j=1}^n a_j^T \\ \sum_{j=1}^n a_j & \sum_{j=1}^n a_j a_j^T \end{bmatrix} \begin{bmatrix} w_{01} & \cdots & w_{0r} \\ w_1 & \cdots & w_r \end{bmatrix} \\ &= \begin{bmatrix} n_1 \beta_1 & \cdots & n_r \beta_r \\ (\sum_{j \in N_1} a_j) \beta_1 & \cdots & (\sum_{j \in N_r} a_j) \beta_r \end{bmatrix}, \end{aligned}$$

resulting in a linear system

$$(4.11) \quad \begin{cases} n w_{0i} + n c^T w_i = n_i \beta_i \\ n c w_{0i} + \left(\sum_{j=1}^n a_j a_j^T \right) w_i = n_i \beta_i c_i \end{cases} \quad \text{for } i = 1, \dots, r.$$

By substituting w_{0i} of the second equation with w_{0i} of the first equation in (4.11), (4.11) becomes

$$(4.12) \quad (n_i \beta_i - n c^T w_i) c + \left(\sum_{j=1}^n a_j a_j^T \right) w_i = n_i \beta_i c_i, \quad i = 1, \dots, r.$$

From (4.12) and

$$S_b = \sum_{1 \leq i \leq r} n_i c_i c_i^T - n c c^T \quad \text{and} \quad S_w = \sum_{1 \leq j \leq n} a_j a_j^T - \sum_{1 \leq i \leq r} n_i c_i c_i^T,$$

we have

$$(4.13) \quad S_t w_i = (S_b + S_w) w_i = n_i \beta_i (c_i - c), \quad i = 1, \dots, r.$$

Recall that according to (2.16), (2.17), and (2.19), the transformation matrix G for LDA was obtained by

$$(4.14) \quad G = F V_1 = U_1 \Sigma^{-1} V_1,$$

where

$$(4.15) \quad S_t = U_1 \Sigma^2 U_1^T \quad \text{and} \quad F^T H_b = \Sigma^{-1} U_1^T H_b = V_1 \Gamma_{b1} S_1^T.$$

The following theorem gives the relation between the MSE solution and the matrix G for LDA.

THEOREM 4.1. *Let G be the dimension reducing transformation matrix from LDA given in (4.14) and let*

$$\{g_i(z) = w_{0i} + w_i^T z\}_{1 \leq i \leq r}$$

be the discriminant functions for the MSE problem (3.7). Then

$$(4.16) \quad \begin{bmatrix} w_1^T \\ \vdots \\ w_r^T \end{bmatrix} U_1 U_1^T = \begin{bmatrix} n_1 \beta_1 (c_1 - c)^T \\ \vdots \\ n_r \beta_r (c_r - c)^T \end{bmatrix} G G^T.$$

Proof. From (4.13), we have

$$(4.17) \quad \begin{aligned} S_t w_i &= n_i \beta_i (c_i - c) \rightarrow U_1 \Sigma^2 U_1^T w_i = n_i \beta_i (c_i - c) \\ &\rightarrow w_i^T U_1 = n_i \beta_i (c_i - c)^T U_1 \Sigma^{-2}. \end{aligned}$$

Then by (4.17),

$$(4.18) \quad \begin{aligned} \begin{bmatrix} w_1^T \\ \vdots \\ w_r^T \end{bmatrix} U_1 U_1^T &= \begin{bmatrix} n_1 \beta_1 (c_1 - c)^T \\ \vdots \\ n_r \beta_r (c_r - c)^T \end{bmatrix} U_1 \Sigma^{-2} U_1^T \\ &= \text{diag}(\sqrt{n_1} \beta_1, \dots, \sqrt{n_r} \beta_r) H_b^T U_1 \Sigma^{-1} (V_1 V_1^T + V_2 V_2^T) \Sigma^{-1} U_1^T \\ &= \text{diag}(\sqrt{n_1} \beta_1, \dots, \sqrt{n_r} \beta_r) H_b^T F V_1 V_1^T F^T \\ &= \begin{bmatrix} n_1 \beta_1 (c_1 - c)^T \\ \vdots \\ n_r \beta_r (c_r - c)^T \end{bmatrix} G G^T. \end{aligned}$$

The third equality in (4.18) holds, since

$$\text{span}(V_2) \subset \text{null}(F^T S_b F) = \text{null}(H_b^T F)$$

from (2.14) and (2.17). $\square$

Let us denote the reduced dimensional representation obtained by the linear transformation G^T from LDA as

$$\tilde{z} = G^T z \quad \text{for any data item } z.$$

First we consider the case that S_w is nonsingular and therefore S_t is nonsingular. In this case, $U = U_1$ is orthogonal and U_2 does not appear in the EVD of S_t in (2.15). Then by Theorem 4.1 and in (4.11) for any data item z ,

$$\begin{aligned} (4.19) \quad \begin{bmatrix} g_1(z) \\ \vdots \\ g_r(z) \end{bmatrix} &= \begin{bmatrix} w_{01} \\ \vdots \\ w_{0r} \end{bmatrix} + \begin{bmatrix} w_1^T \\ \vdots \\ w_r^T \end{bmatrix} z = \begin{bmatrix} n_1\beta_1/n \\ \vdots \\ n_r\beta_r/n \end{bmatrix} + \begin{bmatrix} w_1^T \\ \vdots \\ w_r^T \end{bmatrix} (z - c) \\ &= \begin{bmatrix} n_1\beta_1/n \\ \vdots \\ n_r\beta_r/n \end{bmatrix} + \begin{bmatrix} w_1^T \\ \vdots \\ w_r^T \end{bmatrix} U_1 U_1^T (z - c) \\ &= \begin{bmatrix} n_1\beta_1/n \\ \vdots \\ n_r\beta_r/n \end{bmatrix} + \begin{bmatrix} n_1\beta_1(\tilde{c}_1 - \tilde{c})^T \\ \vdots \\ n_r\beta_r(\tilde{c}_r - \tilde{c})^T \end{bmatrix} (\tilde{z} - \tilde{c}). \end{aligned}$$

Equation (4.19) shows that the decision rule in the generalized MSE solution

$$(4.20) \quad \arg \max_{1 \leq i \leq r} \{g_i(z)\}$$

is equivalent to

$$(4.21) \quad \arg \max_{1 \leq i \leq r} \{n_i\beta_i/n + n_i\beta_i(\tilde{c}_i - \tilde{c})^T(\tilde{z} - \tilde{c})\}$$

in the reduced dimensional space obtained by LDA. This implies that the MSE procedure is equivalent to applying centroid-based classification with an inner product similarity measure in the reduced dimensional space obtained by LDA. If $\beta_i = 1$ ($1 \leq i \leq r$), then (4.21) becomes

$$(4.22) \quad \arg \max_{1 \leq i \leq r} \{n_i/n + n_i(\tilde{c}_i - \tilde{c})^T(\tilde{z} - \tilde{c})\}.$$

On the other hand, with $\beta_i = n/n_i$, i.e.,

$$(4.23) \quad g_i(z) = w_{0i} + w_i^T z = \begin{cases} n/n_i & \text{if } z \in \text{class } i, \\ 0 & \text{otherwise,} \end{cases}$$

(4.21) becomes

$$(4.24) \quad \arg \max_{1 \leq i \leq r} \{(\tilde{c}_i - \tilde{c})^T(\tilde{z} - \tilde{c})\}.$$

The difference between (4.22) and (4.24) is whether weighting by the number of elements in each class is considered or not.

The problem formulation (4.23) also gives a natural generalization of the relationship between the generalized MSE solution for the two-class case and FDA. Let

$\tilde{z}$ be the one-dimensional representation obtained by FDA. Then the equivalence of (4.20) and (4.24) gives

$$\begin{aligned} g_1(z) > g_2(z) &\leftrightarrow (\tilde{c}_1 - \tilde{c})(\tilde{z} - \tilde{c}) > (\tilde{c}_2 - \tilde{c})(\tilde{z} - \tilde{c}) \\ &\leftrightarrow (\tilde{c}_1 - \tilde{c}_2)(\tilde{z} - \tilde{c}) > 0, \end{aligned}$$

indicating the decision rule (3.6) in FDA.

Let us consider undersampled problems where all the scatter matrices are singular, and therefore we have the term U_2 in the EVD of S_t . For a given data item $z = a_i$, by (2.20)

$$U_2^T z = U_2^T c$$

and by Theorem 4.1, we have

$$\begin{aligned} (4.25) \quad \begin{bmatrix} g_1(z) \\ \vdots \\ g_r(z) \end{bmatrix} &= \begin{bmatrix} n_1\beta_1/n \\ \vdots \\ n_r\beta_r/n \end{bmatrix} + \begin{bmatrix} w_1^T \\ \vdots \\ w_r^T \end{bmatrix} (z - c) \\ &= \begin{bmatrix} n_1\beta_1/n \\ \vdots \\ n_r\beta_r/n \end{bmatrix} + \begin{bmatrix} w_1^T \\ \vdots \\ w_r^T \end{bmatrix} (U_1 U_1^T + U_2 U_2^T)(z - c) \\ &= \begin{bmatrix} n_1\beta_1/n \\ \vdots \\ n_r\beta_r/n \end{bmatrix} + \begin{bmatrix} n_1\beta_1(\tilde{c}_1 - \tilde{c})^T \\ \vdots \\ n_r\beta_r(\tilde{c}_r - \tilde{c})^T \end{bmatrix} (\tilde{z} - \tilde{c}). \end{aligned}$$

Equation (4.25) is exactly the same as (4.19), which was obtained for the case when S_w is nonsingular, implying that the above discussion regarding the nonsingular case still holds for undersampled problems. However, when a new unseen data item z is presented, the third equality in (4.25) becomes an approximation since (2.20) is based on the given training data set. If new data items come from the same distribution as the training data, (4.25) should hold almost exactly as the experiments in section 6 show.

5. Performing LDA through the generalized MSE procedure. Now we show how to obtain the reduced dimensional space of LDA through the MSE procedure without computing the transformation matrix G of the LDA procedure. From the relation (4.19) (and also (4.25)) of LDA and MSE, we have

$$(5.1) \quad \begin{bmatrix} w_1^T \\ \vdots \\ w_r^T \end{bmatrix} [c_1 - c, \dots, c_r - c] = \begin{bmatrix} n_1\beta_1(\tilde{c}_1 - \tilde{c})^T \\ \vdots \\ n_r\beta_r(\tilde{c}_r - \tilde{c})^T \end{bmatrix} [\tilde{c}_1 - \tilde{c}, \dots, \tilde{c}_r - \tilde{c}],$$

where $\tilde{c}_i, i = 1, \dots, r$, are the class centroids in the reduced dimensional space obtained by LDA, i.e.,

$$\tilde{c}_i = G^T c_i.$$

Denoting

$$(5.2) \quad L \equiv \begin{bmatrix} \sqrt{n_1}(\tilde{c}_1 - \tilde{c})^T \\ \vdots \\ \sqrt{n_r}(\tilde{c}_r - \tilde{c})^T \end{bmatrix} = H_b^T G,$$

(5.1) becomes

$$(5.3) \quad \begin{bmatrix} \frac{1}{\sqrt{n_1}\beta_1} & & \\ & \ddots & \\ & & \frac{1}{\sqrt{n_r}\beta_r} \end{bmatrix} \begin{bmatrix} w_1^T \\ \vdots \\ w_r^T \end{bmatrix} [c_1 - c, \dots, c_r - c] \begin{bmatrix} \sqrt{n_1} & & \\ & \ddots & \\ & & \sqrt{n_r} \end{bmatrix} = LL^T.$$

Let the EVD of the left side in (5.3) be $Q\Lambda Q^T$, where Q is orthogonal and Λ has nonincreasing diagonal components. Then

$$(5.4) \quad LL^T = Q\Lambda Q^T \equiv \underbrace{[Q_1]}_{r-1} \underbrace{[Q_2]}_1 \begin{bmatrix} \Lambda_1 & 0 \\ 0 & 0 \end{bmatrix} \begin{bmatrix} Q_1^T \\ Q_2^T \end{bmatrix} = Q_1 \Lambda_1 Q_1^T.$$

On the other hand, from (5.2), (4.14), and (4.15)

$$(5.5) \quad L^T L = G^T H_b H_b^T G = (V_1^T F^T H_b) (H_b^T F V_1) = (\Gamma_{b1} S_1^T) (S_1 \Gamma_{b1}^T) = \Gamma_{b1} \Gamma_{b1}^T.$$

Hence from (5.4) and (5.5),

$$\Lambda_1 = \Gamma_{b1} \Gamma_{b1}^T$$

and we can obtain the SVD of L as

$$L = Q_1 \Lambda_1^{1/2}, \quad \text{i.e.,} \quad L^+ = \left(\Lambda_1^{1/2} \right)^+ Q_1^T.$$

When

$$\text{rank}(H_b) = \text{rank}([c_1 - c, \dots, c_r - c]) = r - 1,$$

L has full column rank and $L^+ L = I$. When S_w is nonsingular, from (4.19) we have

$$(5.6) \quad \begin{bmatrix} g_1(z) \\ \vdots \\ g_r(z) \end{bmatrix} = \begin{bmatrix} w_{01} \\ \vdots \\ w_{0r} \end{bmatrix} + \begin{bmatrix} n_1 \beta_1 (\tilde{c}_1 - \tilde{c})^T \\ \vdots \\ n_r \beta_r (\tilde{c}_r - \tilde{c})^T \end{bmatrix} \tilde{z} \\ = \begin{bmatrix} w_{01} \\ \vdots \\ w_{0r} \end{bmatrix} + \begin{bmatrix} \sqrt{n_1} \beta_1 & & \\ & \ddots & \\ & & \sqrt{n_r} \beta_r \end{bmatrix} L \tilde{z},$$

and therefore

$$(5.7) \quad \tilde{z} = L^+ \begin{bmatrix} \frac{1}{\sqrt{n_1}\beta_1} & & \\ & \ddots & \\ & & \frac{1}{\sqrt{n_r}\beta_r} \end{bmatrix} \left(\begin{bmatrix} g_1(z) \\ \vdots \\ g_r(z) \end{bmatrix} - \begin{bmatrix} w_{01} \\ \vdots \\ w_{0r} \end{bmatrix} \right) \\ = (\Lambda_1^{1/2})^{-1} Q_1^T \begin{bmatrix} \frac{1}{\sqrt{n_1}\beta_1} w_1^T \\ \vdots \\ \frac{1}{\sqrt{n_r}\beta_r} w_r^T \end{bmatrix} z.$$

Equation (5.7) shows that the reduced dimensional representation by LDA can be obtained from the discriminant functions of the MSE solution

$$\{g_i(z) = w_{0i} + w_i^T z\}_{(1 \leq i \leq r)}$$

ALGORITHM 1. An efficient algorithm for LDA.

Given a data matrix $A \in \mathbb{R}^{m \times n}$ with r classes, this computes a $(r - 1)$ -dimensional representation of any data point $z \in \mathbb{R}^{m \times 1}$.

1. Compute $\begin{bmatrix} w_{01} & \cdots & w_{0r} \end{bmatrix} = \mathcal{P}^+ \mathcal{Y}$, where $\mathcal{P}$ and $\mathcal{Y}$ are defined in (3.3) and (3.9), respectively.
2. Compute the EVD of the left-hand side of (5.3):

$$\begin{aligned} & \begin{bmatrix} \frac{1}{\sqrt{n_1}\beta_1} w_1^T \\ \vdots \\ \frac{1}{\sqrt{n_r}\beta_r} w_r^T \end{bmatrix} \begin{bmatrix} \sqrt{n_1}(c_1 - c), \dots, \sqrt{n_r}(c_r - c) \end{bmatrix} \\ &= \underbrace{\begin{bmatrix} Q_1 & Q_2 \end{bmatrix}}_{\substack{r-1 \\ 1}} \begin{bmatrix} \Lambda_1 & 0 \\ 0 & 0 \end{bmatrix} \begin{bmatrix} Q_1^T \\ Q_2^T \end{bmatrix}. \end{aligned}$$

3. For any data item z , the $(r - 1)$ -dimensional representation is given by

$$\Lambda_{11}^{-1/2} Q_1^T \begin{bmatrix} \frac{1}{\sqrt{n_1}\beta_1} w_1^T \\ \vdots \\ \frac{1}{\sqrt{n_r}\beta_r} w_r^T \end{bmatrix} z.$$

and the EVD of the $r \times r$ matrix, instead of solving the generalized eigenvalue problem for LDA.

For undersampled problems, (5.7) must change accordingly:

$$(5.8) \quad \begin{bmatrix} g_1(z) \\ \vdots \\ g_r(z) \end{bmatrix} = \begin{bmatrix} w_{01} \\ \vdots \\ w_{0r} \end{bmatrix} + \begin{bmatrix} w_1^T \\ \vdots \\ w_r^T \end{bmatrix} U_2 U_2^T z + \begin{bmatrix} n_1 \beta_1 (\tilde{c}_1 - \tilde{c})^T \\ \vdots \\ n_r \beta_r (\tilde{c}_r - \tilde{c})^T \end{bmatrix} \tilde{z}.$$

The second term of the right-hand side in (5.8) is invariant for any given training data item since they are transformed to the constant point by U_2^T . Hence we can obtain the reduced dimensional representation except for a translation factor as

$$(5.9) \quad \tilde{z} \approx \left(\Lambda_1^{1/2} \right)^{-1} Q_1^T \begin{bmatrix} \frac{1}{\sqrt{n_1}\beta_1} w_1^T \\ \vdots \\ \frac{1}{\sqrt{n_r}\beta_r} w_r^T \end{bmatrix} z.$$

The new algorithm to compute the LDA solution is summarized in Algorithm 1.

This approach to LDA utilizing the relation with the MSE solution has the following properties. First, the scatter matrices S_b and S_w need not be computed explicitly. It reduces computational complexities and saves memory requirements. Second, in addition to the SVD of the matrix $\mathcal{P}$ defined in (3.3), the EVD is needed only for an $r \times r$ matrix, where the number of classes r is usually much smaller than the data dimension m or the total number of data n . Table 1 compares computational complexities among the classical LDA, LDA/GSVD [18, 19], and the newly proposed Algorithm 1. The cost for computing the SVD of an $m \times n$ ($m \geq n$) matrix is estimated as $O(mn^2)$ [21, p. 254]. When $\mathcal{P}$ has full rank, QR decomposition can be used

TABLE 1

The comparison of computational complexities. m : data dimension; n : number of data items; r : number of classes; s : rank of Z .

	When S_w is nonsingular	Undersampled problems
Classical LDA $S_w^{-1}S_b x = \lambda x$	S_w^{-1} : $O(m^3)$ EVD of $S_w^{-1}S_b$: $O(m^3)$	Not applicable
LDA/GSVD in [18]	SVD of $Z = \begin{bmatrix} H_b^T \\ H_w^T \end{bmatrix}$: $O(\min\{m(n+r)^2, m^2(n+r)\})$ SVD of $P(1:r, 1:s)$: $O(sr^2)$	
Algorithm 1 via relation with MSE	Pseudoinverse of $\mathcal{P}$: $O(\min\{mn^2, m^2n\})$ Step 2: $O(r^3)$	

TABLE 2

The description of data sets.

	Data set	No. of classes	Dimension	No. of data
UCI Machine Learning Repository	Musk	2	166	6598
	Isolet	26	617	7797
	M-feature	10	649	2000
	B-scale	3	4	625
	B-cancer	2	9	699
	Wdbc	2	30	569
	Car	4	6	1728
	Glass	2	9	214
Text documents	Cacmcisi	2	14409	4663
	Cranmed	2	9038	2431
	Hitech	6	13170	2301
	La1	6	17273	3204
	La2	6	15211	3075
	Tr23	6	5832	204
	Tr41	10	7454	878
	Tr45	10	8261	690

to compute the pseudoinverse of $\mathcal{P}$, which is cheaper than the SVD [21]. This is due to the fact that when $m+1 \geq n$, the reduced QR decomposition of $\mathcal{P}^T = Q_1 R$ gives the pseudoinverse of $\mathcal{P}^T$ as $R^{-1}Q_1^T$; therefore, $\mathcal{P}^+ = Q_1(R^T)^{-1}$.

6. Experimental results. In order to verify the theoretical results for the relationship between LDA and the MSE procedure, we conducted extensive experiments. The experiments use two types of data sets: the first has a nonsingular within-class scatter matrix S_w , and therefore the classical LDA can be performed for these data sets; the other is from undersampled problems which have singular scatter matrices. Data sets were collected from the UCI Machine Learning Repository² and text documents.³ A collection of text documents is represented as a term-document matrix, where each document is expressed as a column vector. The term-document matrix is obtained after preprocessing with common words and rare term removal, stemming, and term frequency and inverse term frequency weighting and normalization [22]. The term-document matrix representation often makes the high dimensionality inevitable. Each data set is split randomly into training data and test data of equal size, and this is repeated 10 times in order to prevent any possible bias from random splitting. The detailed description of the data sets are shown in Table 2.

²<http://www.ics.uci.edu/~mllearn/MLRepository.html>

³<http://www-users.cs.umn.edu/~karypis/cluto/download.html>

TABLE 3

The comparison of classification performances for the verification of the relation (6.1). The mean prediction accuracies (%) from 10 times random splittings of training and test sets are shown.

	$\beta_i = 1$		$\beta_i = n/n_i$	
	MSE	LDA	MSE	LDA
Musk	93.7	93.7	79.7	79.7
M-feature	98.0	98.0	98.0	98.0
B-scale	87.2	87.2	83.1	83.1
B-cancer	95.8	95.8	96.9	96.9
Wdbc	95.1	95.1	96.1	96.1
Car	76.8	76.8	45.9	45.9
Glass	91.5	91.5	91.4	91.4
Isolet	91.3	91.3	91.3	91.3
Undersampled problems				
Cacmcisi	95.3	95.3	96.3	96.3
Cranmed	99.8	99.8	99.7	99.7
Hitech	70.5	70.5	62.7	62.7
La1	87.8	87.8	82.2	82.2
La2	89.2	89.2	84.4	84.4
Tr23	89.5	89.5	80.9	80.7
Tr41	95.7	95.7	84.7	84.7
Tr45	92.8	92.8	87.7	87.6

TABLE 4

Verification of the new efficient algorithm for LDA. The mean prediction accuracies (%) from 10 times random splittings are shown.

	LDA			Algorithm 1		
	1-NN	15-NN	29-NN	1-NN	15-NN	29-NN
Musk	91.4	93.8	93.9	91.4	93.8	93.9
M-feature	98.1	98.1	98.1	98.1	98.1	98.1
B-scale	87.3	88.1	88.5	87.2	88.1	88.5
B-cancer	95.5	96.8	96.4	95.5	96.8	96.4
Wdbc	95.2	96.2	95.9	95.2	96.2	95.9
Car	88.0	87.1	86.6	88.0	87.1	86.6
Glass	90.8	91.3	90.9	90.8	91.3	90.9
Isolet	92.0	92.5	92.2	92.0	92.5	92.2
Undersampled problems						
Cacmcisi	95.3	95.3	95.3	95.3	95.3	95.3
Cranmed	99.8	99.8	99.8	99.8	99.8	99.8
Hitech	69.9	69.9	69.9	69.9	69.9	69.9
La1	86.4	86.4	86.4	86.4	86.4	86.4
La2	87.7	87.7	87.7	87.7	87.7	87.7
Tr23	84.6	75.9	75.9	84.6	75.9	75.9
Tr41	93.9	93.4	90.0	93.9	93.4	90.0
Tr45	88.6	88.4	86.3	88.6	88.4	86.3

For all data sets in Table 2 the relationship between the MSE procedure and LDA

$$\begin{aligned}
 (6.1) \quad & \arg \max_{1 \leq i \leq r} \{g_i(z) = w_{0i} + w^T z\} \\
 & = \arg \max_{1 \leq i \leq r} \left\{ \frac{n_i \beta_i}{n} + n_i \beta_i (G^T c_i - G^T c)^T (G^T z - G^T c) \right\}
 \end{aligned}$$

was demonstrated by comparing the prediction accuracies. Table 3 reports the mean prediction accuracies (%) from 10 random splittings of training and test sets. The

relation (6.1) was verified for all the data sets, subject to minor differences in Tr23 and Tr45.

Algorithm 1 was tested for all the data sets in order to verify our derivation by comparing the prediction accuracies by Algorithm 1 with those by LDA using k -NN classifier. Table 4 shows the mean prediction accuracies (%) from 10 runs. Exactly the same results were obtained by both algorithms in all the data sets used except in the B-scale data set with a 1-NN classifier, which resulted in a 0.1% difference.

7. Conclusion. In this paper, we have shown a relationship between LDA and the generalized MSE solution for multiclass problems. It generalizes the relation between the MSE solution and FDA to multiclass cases and on undersampled problems. We also proposed an efficient algorithm for LDA which utilizes the relationship with the generalized MSE solution. In Algorithm 1, the generalized eigenvalue problem is solved by the SVDs of the matrix $\mathcal{P}$ and the small $r \times r$ matrix. In addition, the proposed algorithm does not need to explicitly compute the scatter matrices, thus saving computational costs as well as memory requirements.

REFERENCES

- [1] R. A. FISHER, *The use of multiple measurements in taxonomic problems*, Annu. Eugenics, 7 (Part II) (1936), pp. 179–188.
- [2] F. ROSENBLATT, *Principles of Neurodynamics: Perceptrons and the Theory of Brain Mechanisms*, Spartan Books, Washington, D.C., 1962.
- [3] B. E. BOSER, I. M. GUYON, AND V. VAPNIK, *A training algorithm for optimal margin classifiers*, in Proceedings of the Fifth Annual Workshop on Computational Learning Theory (Pittsburgh, PA), ACM, New York, 1992.
- [4] R. O. DUDA AND P. E. HART, *Pattern Classification and Scene Analysis*, Wiley-Interscience, New York, 1973.
- [5] Y-C. HO AND R. L. KASHYAP, *An algorithm for linear inequalities and its applications*, IEEE Trans. Elec. Comp., 14 (1965), pp. 683–688.
- [6] R. O. DUDA, P. E. HART, AND D. G. STORK, *Pattern Classification*, Wiley-Interscience, New York, 2001.
- [7] K. FUKUNAGA, *Introduction to Statistical Pattern Recognition*, 2nd ed., Academic Press, New York, 1990.
- [8] J. S. KOFORD AND G. F. GRONER, *The use of an adaptive threshold element to design a linear optimal pattern classifier*, IEEE Trans. Inf. Theory, 12 (1966), pp. 42–50.
- [9] T. HASTIE, R. TIBSHIRANI, AND A. BUJA, *Flexible discriminant analysis by optimal scoring*, J. Amer. Statist. Assoc., 89 (1994), pp. 1255–1270.
- [10] T. HASTIE AND R. TIBSHIRANI, *Discriminant analysis by Gaussian mixture*, J. Roy. Statist. Soc. Ser. B, 58 (1996), pp. 155–176.
- [11] N. CAMPBELL, *Canonical variate analysis—A general formulation*, Austral. J. Statist., 26 (1984), pp. 86–96.
- [12] J. H. FRIEDMAN, *Regularized discriminant analysis*, J. Amer. Statist. Assoc., 84 (1989), pp. 165–175.
- [13] T. HASTIE, A. BUJA, AND R. TIBSHIRANI, *Penalized discriminant analysis*, Ann. Statist., 23 (1995), pp. 73–102.
- [14] L. CHEN, H. M. LIAO, M. KO, J. LIN, AND G. YU, *A new LDA-based face recognition system which can solve the small sample size problem*, Pattern Recogn., 33 (2000), pp. 1713–1726.
- [15] H. YU AND J. YANG, *A direct LDA algorithm for high-dimensional data with application to face recognition*, Pattern Recogn., 34 (2001), pp. 2067–2070.
- [16] X. WANG AND X. TANG, *Dual-space linear discriminant analysis for face recognition*, in Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Washington, D.C., 2004.
- [17] J. YANG, J.-Y. YANG, AND D. ZHANG, *What's wrong with Fisher criterion?*, Pattern Recogn., 35 (2002), pp. 2665–2668.
- [18] P. HOWLAND, M. JEON, AND H. PARK, *Structure preserving dimension reduction for clustered text data based on the generalized singular value decomposition*, SIAM J. Matrix Anal. Appl., 25 (2003), pp. 165–179.

- [19] P. HOWLAND AND H. PARK, *Generalizing discriminant analysis using the generalized singular value decomposition*, IEEE Trans. Pattern Anal. Machine Intel., 26 (2004), pp. 995–1006.
- [20] C. C. PAIGE AND M. A. SAUNDERS, *Towards a generalized singular value decomposition*, SIAM J. Numer. Anal., 18 (1981), pp. 398–405.
- [21] G. H. GOLUB AND C. F. VAN LOAN, *Matrix Computations*, 3rd ed., Johns Hopkins University Press, Baltimore, MD, 1996.
- [22] T. G. KOLDA AND D. P. O’LEARY, *A semidiscrete matrix decomposition for latent semantic indexing in information retrieval*, ACM Trans. Inf. Syst., 16 (1998), pp. 322–346.

ERROR ANALYSIS OF THE QUASI-GRAM–SCHMIDT ALGORITHM*

G. W. STEWART†

Abstract. Let the $n \times p$ ($n \geq p$) matrix X have the QR factorization $X = QR$, where R is an upper triangular matrix of order p and Q is orthonormal. This widely used decomposition has the drawback that Q is not generally sparse even when X is. One cure is to discard Q , retaining only X and R . Products like $a = Q^T y = R^{-T} X^T y$ can then be formed by computing $b = X^T y$ and solving the system $R^T a = b$. This approach can be used to modify the Gram–Schmidt algorithm for computing Q and R to compute R without forming Q or altering X . Unfortunately, this quasi-Gram–Schmidt algorithm can produce inaccurate results. In this paper it is shown that with reorthogonalization the inaccuracies are bounded under certain natural conditions.

Key words. Gram–Schmidt algorithm, sparse matrix, QR factorization, orthogonalization, rounding-error analysis

AMS subject classifications. 65F25, 65F50

DOI. 10.1137/040607794

1. Introduction. In many applications (see, e.g., [1, 2, 8, 11]) it is necessary to find low-rank approximations to a large sparse matrix A . One way of doing this is to compute a partial QR factorization with column pivoting. The k th stage of this process produces a factorization of the form

$$A \cong Q_k R_k,$$

where Q_k is orthonormal with k columns and R_k is upper triangular with k rows. It also gives the norm of the error in the approximation, so that k can be determined adaptively. Unfortunately, fill-in can cause Q_k and R_k to be denser than A .¹ This problem is exacerbated by the pivoting strategy, which ignores sparsity considerations as it tries to produce good low-rank approximations.

In [10] the author has shown how to use a Q-less QR factorization that requires only the first k columns of R_k to compute low-rank approximations that use rows and columns of A itself as bases for the approximation.² The heart of this procedure is a method called the quasi-Gram–Schmidt method—or for short, the QGS method. The purpose of this paper is to give an informal rounding-error analysis that exhibits the properties of the method.

To set the background let X be an $n \times p$ matrix with $n \geq p$.³ Then X has the

*Received by the editors May 5, 2004; accepted for publication (in revised form) by M. Hanke March 9, 2005; published electronically November 22, 2005.

<http://www.siam.org/journals/simax/27-2/60779.html>

†Department of Computer Science and Institute for Advanced Computer Studies, University of Maryland, College Park, MD 20742 (stewart@cs.umd.edu). This work was supported in part by the National Science Foundation under grant CCR0204084. This report is available by anonymous ftp from thales.cs.umd.edu in the directory pub/reports or on the Web at <http://www.cs.umd.edu/~stewart/>.

¹For example, in linear regression the matrix A often has the form $(\mathbf{e} \ X)$, where $\mathbf{e}$ is a vector whose components are one that cannot be left out of the model. Its presence causes Q to fill in completely after one step.

²For a MATLAB implementation see [3].

³In the problem of low-rank approximation, X represents a selection of columns of A chosen by the pivoting process.

QR factorization

$$X = QR,$$

where $Q^T Q = I$ and R is upper triangular with positive diagonal elements. If X has full column rank—as we will assume from now on—then this QR factorization is unique.

An important way of computing a QR factorization is the Gram–Schmidt algorithm with reorthogonalization. The heart of the method is an algorithm for updating a QR factorization. Suppose we know the QR factorization of X and wish to compute the QR factorization of $(X \ x)$, where x is independent of the columns of X . This can be done as follows.

$$(1.1) \quad \begin{array}{ll} 1. & r = Q^T x \\ 2. & u = x - Qr \\ 3. & \rho = \|u\| \\ 4. & q = u/\rho \end{array}$$

Here the norm $\|\cdot\|$ is the vector 2-norm. We call this process a Gram–Schmidt step. It is easy to show that if q , r , and ρ are computed exactly then

$$(X \ x) = (Q \ q) \begin{pmatrix} R & r \\ 0 & \rho \end{pmatrix}$$

is the QR factorization of $(X \ x)$. The Gram–Schmidt algorithm consists of applying Gram–Schmidt steps successively to the columns of a matrix to obtain its QR factorization.

When the Gram–Schmidt step is executed in finite-precision floating-point arithmetic, the vector q may not be orthogonal to the columns of Q . Specifically, when there is cancellation in step 2 of (1.1), the trailing digits in the computed value of u will be inaccurate, and u will deviate from orthogonalization in proportion to the degree of cancellation.

The cure for this problem is to repeat the orthogonalization as follows.

$$(1.2) \quad \begin{array}{ll} 1. & r_1 = Q^T x \\ 2. & u_1 = x - Qr_1 \\ 3. & r_2 = Q^T u_1 \\ 4. & u_2 = u_1 - Qr_2 \\ 5. & r = r_1 + r_2 \\ 6. & \rho = \|u_2\| \\ 7. & q = u_2/\rho \end{array}$$

Typically this algorithm exhibits the following behavior. The norm of u_1 is less than the norm of x by a factor proportional to the degree of dependence of x on X . The norm of u_2 , on the other hand, is not much smaller than the norm of u , which guarantees the orthogonality of q to the column space of X . For more on the properties of Gram–Schmidt with reorthogonalization, see [5, section 2.2.4] or [9, section 3.1.4]).⁴

When X is large and sparse, the Gram–Schmidt algorithm is unsatisfactory. The reason is that as the algorithm progresses, the new columns of Q become less and less

⁴Typically, u_2 can fail to be orthogonal to the column space of x , in which case the reorthogonalization process must be repeated.

sparse. A cure for this problem [10] is to recognize that, since $Q = XR^{-1}$, it is only necessary to retain the matrix R . Products such as $r = Q^T x$ can be computed by first computing $a = X^T x$ and then solving the system $R^T r = a$. These considerations lead to the following QGS step with reorthogonalization.

$$(1.3) \quad \begin{aligned} 1. & \quad a_1 = X^T x \\ 2. & \quad \text{Solve the system } R^T r_1 = a_1 \\ 3. & \quad \text{Solve the system } R b_1 = r_1 \\ 4. & \quad u_1 = x - X b_1 \\ 5. & \quad a_2 = X^T u_1 \\ 6. & \quad \text{Solve the system } R^T r_2 = a_2 \\ 7. & \quad \text{Solve the system } R b_2 = r_2 \\ 8. & \quad u_2 = u_1 - X b_2 \\ 9. & \quad r = r_1 + r_2 \\ 10. & \quad \rho = \|u_2\| \end{aligned}$$

It is important to note that we do not compute $q = u_2/\rho$. Instead we *define* q to be the last column of

$$(1.4) \quad (X \ x) \begin{pmatrix} R & r \\ 0 & \rho \end{pmatrix}^{-1} = (X \ x) \begin{pmatrix} R^{-1} & -\rho^{-1} R^{-1} r \\ 0 & \rho^{-1} \end{pmatrix};$$

that is,

$$(1.5) \quad q = \rho^{-1} x - \rho^{-1} X R^{-1} r = \rho^{-1} (x - Q r).$$

There is good reason to be wary of this algorithm. If R is ill conditioned, the systems in statements 1, 2, 3, and 4 may be solved inaccurately, and these inaccuracies could compromise the orthogonality of $Q = XR^{-1}$. In fact, something like this must happen. For suppose $\tilde{R}$ is the correctly rounded R-factor from the QR factorization of X . Then $\tilde{R} = R + E$, where

$$(1.6) \quad \|E\| \leq \gamma \|R\| \epsilon_M.$$

Here ϵ_M is the rounding unit (about $2.2 \cdot 10^{-16}$ in IEEE double-precision floating-point arithmetic), $\|\cdot\|$ is the spectral norm, and γ is a constant depending on the dimensions of X . Suppose we compute an approximation $\tilde{Q}$ to Q using $R + E$ but with no further error. Noting that up to terms of order ϵ_M^2

$$(R + E)^{-1} \cong R^{-1} - R^{-1} E R^{-1},$$

we find that

$$\tilde{Q} = X(R + E)^{-1} \cong Q - Q E R^{-1}.$$

Hence

$$\tilde{Q}^T \tilde{Q} \cong Q^T Q - Q^T Q E R^{-1} - R^{-T} E Q^T Q.$$

Hence if we define

$$W = Q^T Q - I$$

and take

$$\omega = \|W\|$$

as a measure of the nonorthonormality of Q , then

$$(1.7) \quad \tilde{\omega} \lesssim \omega + 2\|Q^T Q\| \|R^{-1}\| \|E\| \leq \omega + 2\gamma\kappa(R)\epsilon_M,$$

where $\kappa(R) = \|R\| \|R^{-1}\|$. Thus, however small the value of ω , the rounding of R is likely to increase it to something on the order of $\gamma\kappa(R)\epsilon_M$.

The above analysis shows that there is a lower limit on the orthogonality the QGS algorithm can be expected to attain. In [10], the author gave numerical examples that suggested that the algorithm would come close to this limiting accuracy, provided a reorthogonalization step is included (as it is in (1.3)). The purpose of this note is to show by an informal rounding-error analysis of the QGS step that this is likely to be the case.

This paper is organized as follows. In section 2 we discuss the effects of scaling on the condition of R and on sizes of the quantities in the QGS step. In section 3 we give some numerical examples of the behavior of the QGS method to provide the reader with an advanced view of where our informal analysis is headed. In section 4 we show how the QGS step behaves in the absence of rounding error. This analysis is related to material in a paper by Hoffmann [7]. Rounding error is introduced in section 5. The analysis itself is given in section 6, followed by a discussion.

As above, ϵ_M will denote the rounding unit and $\|\cdot\|$ the spectral norm. We will ignore second and higher order terms in ϵ_M and indicate their omission by the substitution of “ $\cong$ ” for “ $=$ ” and “ $\lesssim$ ” for “ $\leq$ ”. We will use a generic constant $\langle\gamma\rangle$ to make minor adjustments in our bounds or to restore “ $=$ ” or “ $\leq$ ”. Note that two appearances of $\langle\gamma\rangle$ in an expression need not represent the same number.

We will denote the column space of X by $\mathcal{R}(X)$ and its orthogonal complement by $\mathcal{R}(X)^\perp$. For any vector u , we will write

$$u = u_X + u_\perp,$$

where u_X is the orthogonal projection of u onto $\mathcal{R}(X)$ and $u_\perp$ is the orthogonal projection of u onto $\mathcal{R}(X)^\perp$. We will measure the orthogonality of u to $\mathcal{R}(X)$ by the ratio

$$(1.8) \quad \tau(u) = \frac{\|u_X\|}{\|u_\perp\|}.$$

Note that $\tau(u)$ is the tangent of the angle between $u_\perp$ and $\mathcal{R}(X)^\perp$. Thus a small value indicates that u has only a small component along $\mathcal{R}(X)$. Conversely, a large value indicates that u has only a small component in $\mathcal{R}(X)^\perp$.

The QGS algorithm without reorthogonalization is related to the method of seminormal equations for solving the least squares problem

$$\|x - Xr\|_2^2 = \min.$$

With reorthogonalization it is related to Björck’s method [4] of corrected seminormal equations. However, both methods assume that the matrix R is the R-factor from a nearby matrix $X + E$, where $\|E\|$ is of the order $\|X\|\epsilon_M$. Björck gives an extensive error analysis; however, it does not seem adaptable to our problem of tracking the orthogonality of the implicit Q-factor as the QGS method progresses.

2. Scaling and magnitude. Before beginning the analysis proper, we must dispose of some questions of scaling and magnitude. Let $D = \text{diag}(d_1, \dots, d_p)$ be any diagonal matrix with positive diagonal elements. Then

$$Q = XR^{-1} = (XD)(RD)^{-1}.$$

Thus, without affecting Q we can scale the columns of X and R by arbitrary factors. In particular, by taking $d_1, \dots, d_{p-1} = 1$ and d_p sufficiently small, we can make $\kappa(R)$ as large as we want. This means that the bound (1.7) on $\tilde{\omega}$ can be made arbitrarily large. Thus for this bound to be informative, we must find a scaling that minimizes $\kappa(R)$. Although this problem is unsolved, a remarkable theorem of van der Sluis [12] states that the condition number of R is approximately minimized when the columns of R all have the same norm.

It is also important to have a sense of the sizes of the quantities involved in our bounds. Since $\|Q\|^2 = \|Q^T Q\| = \|I + W\|$, we have

$$\sqrt{1 - \omega} \leq \|Q\| \leq \sqrt{1 + \omega}.$$

Moreover, $X^T X = R^T (Q^T Q) R = R^T (I + W) R$. Hence,

$$\sqrt{1 - \omega} \|R\| \leq \|X\| \leq \sqrt{1 + \omega} \|R\|.$$

Thus if ω is less than, say, 0.1, then $\|Q\|$ is approximately 1, and $\|X\|$ and $\|R\|$ are approximately equal. Similar results hold for the individual columns of Q , X , and R .

In what follows we will assume that

$$\|X\| = 1 \quad \text{and} \quad \|x\| = 1.$$

This means that in expressions like $\|R\|_{\epsilon_M}$ we can absorb the $\|R\|$ in our generic constant $\langle \gamma \rangle$, with a resulting simplification in the bounds. We will also assume that the columns x_j of X all have the same norm. This means that the columns of R all have approximately the same norm, which, as we have seen above, is required to make our results meaningful.

3. Three examples. To help the reader follow the analysis of the QGS method we present three numerical examples that illustrate the behavior of algorithm (1.3) (see Table 3.1). In all three examples, four steps of algorithm (1.3) are applied to a 50×5 matrix $X = (x_1 \cdots x_5)$ to orthogonalize its columns. Thus after the k th step, QR is the quasi-QR factorization of $X = (x_1 \cdots x_{k+1})$. The matrices X were generated in the form

$$X = USV^T,$$

where U is a 50×5 random orthonormal matrix, V is a 5×5 random orthogonal matrix, and S is a diagonal matrix containing the singular values of X . All computations were performed in IEEE standard arithmetic with a rounding unit of about $2.2 \cdot 10^{-16}$.

In the first example, there is a sharp relative gap of about 10^{-6} between the second and third singular values. The last two columns in the table contain the values of ω and

$$(3.1) \quad \hat{\alpha} = \|R^{-1}\|_{\epsilon_M}$$

for the Q and R after the QGS step. They track each other nicely, as we have conjectured.

TABLE 3.1
Three examples.

Example 1: Singular values

	1.0e+00	7.2e-01	3.6e-07	1.0e-07	6.1e-08	
$\hat{\alpha}$	τ_x	τ_1	τ_2	τ_q	ω	$\hat{\alpha}_{\text{new}}$
2.2e-16	6.7e+00	6.4e-16	2.1e-17	5.7e-16	8.6e-16	3.8e-15
3.8e-15	2.1e+05	1.2e-10	4.8e-16	3.2e-11	5.3e-11	7.5e-10
7.5e-10	3.8e+06	6.4e-06	5.3e-11	4.4e-10	4.2e-10	2.4e-09
2.4e-09	6.0e+06	2.4e-03	5.7e-11	4.2e-10	6.2e-10	3.7e-09

Example 2: Singular values

	1.0e+00	1.4e-01	1.6e-03	4.6e-06	1.8e-07	
$\hat{\alpha}$	τ_x	τ_1	τ_2	τ_q	ω	$\hat{\alpha}_{\text{new}}$
2.2e-16	6.8e+00	8.0e-16	5.3e-17	1.2e-16	4.6e-16	4.8e-15
4.8e-15	1.1e+04	3.0e-12	6.8e-17	3.4e-13	4.7e-13	7.6e-12
7.6e-12	6.9e+02	3.6e-09	1.9e-13	1.2e-11	1.5e-11	5.0e-11
5.0e-11	2.1e+06	1.3e-05	1.4e-12	4.4e-11	1.3e-10	1.3e-09

Example 3: Singular values

	1.0e+00	4.6e-04	2.3e-07	1.2e-11	7.3e-16	
$\hat{\alpha}$	τ_x	τ_1	τ_2	τ_q	ω	$\hat{\alpha}_{\text{new}}$
2.2e-16	2.9e+03	1.8e-13	0.0e+00	1.4e-13	1.7e-13	2.0e-12
2.0e-12	4.3e+05	1.7e-07	2.7e-13	9.1e-11	1.3e-10	1.1e-09
1.1e-09	4.3e+10	9.0e+00	2.2e-09	2.6e-06	3.6e-06	2.6e-05
2.6e-05	3.8e+14	2.2e+05	1.1e+00	2.7e+03	1.0e+00	7.2e-05

$$\hat{\alpha} = \|R^{-1}\|_{\epsilon_M}, \quad \tau_x = \tau(x), \quad \tau_1 = \tau(u_1), \quad \tau_2 = \tau(u_2), \quad \tau_q = \tau(q), \quad \omega = \|I - Q^T Q\|$$

Moving across the first row, we find that x_2 has a reasonable component along the orthogonal complement of the first column as measured by τ_x . One QGS step produces that component to almost full accuracy, and the second orthogonalization is redundant. The new 50×2 matrix Q is almost fully orthonormal, and the corresponding value of $\hat{\alpha}$ is near the rounding unit.

The second row tells a more interesting tale. The columns x_1 and x_2 approximately span the subspace spanned by the singular vectors corresponding to the two large singular values. This means that when x_3 is orthogonalized it must lie approximately in the space spanned by the remaining singular vectors. Since the corresponding singular values are small, $x_{\perp}^{(3)}$ must be small and hence τ_{x_3} is large. In fact, it is so large that it takes two QGS steps to make the result fully orthogonal. It is significant that $\tau_1 \cong \hat{\alpha}\tau_x$. It is also significant that although u_2 is fully orthogonal, some orthogonality is lost in the passage to q . This loss is proportional to the increase in $\hat{\alpha}$.

The third row illustrates yet another point. As in the first column we have $\tau_1 \cong \hat{\alpha}\tau_x$. But in the reorthogonalization step the reduction stagnates: τ_2 is approximately equal to $\hat{\alpha}$. (The reduction also stagnates in the second row. But since $\hat{\alpha}$ is near the rounding unit, it cannot be seen whether the stagnation is due to the size of $\hat{\alpha}$ or to the fact that we cannot hope to orthogonalize beyond the rounding unit.)

The second example shows the same phenomenon in a gentler setting. The singular values grade smoothly without gaps from 1 to about 10^{-7} . The values of τ_x reflect this grading. In all cases $\tau_1 \cong \hat{\alpha}\tau_x$, but τ_2 is never much less than $\hat{\alpha}$. At every step, some orthogonality is lost in the passage from u_2 to q and the loss is approximately proportional to the increase in $\hat{\alpha}$.

The third example shows that the QGS method can fail. The singular values are

steeply graded from 1 to about 10^{-15} . The first three rows behave in the manner we have come to expect. But in the fourth row the two orthogonalizations are not sufficient to make τ_2 of the same order of magnitude as $\hat{\alpha}$, and orthogonality in Q is lost. We could overcome this problem by including additional reorthogonalizations. But as we shall see, there are good reasons for not doing so.

To summarize, we must explain three things about the algorithm.

1. Why do the orthogonalization steps enhance orthogonality by a factor of about $\hat{\alpha}$?
2. Why is $\hat{\alpha}$ a lower bound on the attainable orthogonality in u_1 and u_2 ?
3. Why is orthogonality lost in passing from u_2 to q , and why is this loss proportional to the increase in $\hat{\alpha}$?

4. The exact QGS step. The idea for our analysis is to show first that, absent rounding errors, repeated QGS steps produce increasingly orthogonal vectors q . Then, in the next section, we show that rounding errors perturb that q by quantities of order $\|R^{-1}\|_{\epsilon_M}$. There we also analyze the formation of q from u_2 .

Without rounding error the QGS step without reorthogonalization can be written in the form

$$u = (I - QQ^T)x \equiv (I - P)x,$$

where $Q = XR^{-1}$. Here we skip the last normalization step. As usual, decompose

$$x = x_X + x_\perp \quad \text{and} \quad u = u_X + u_\perp.$$

The problem then is to find expressions for u_X and $u_\perp$.

The vector $u_\perp$ is easy to find. Let $U_\perp$ be any orthonormal basis for $\mathcal{R}(X)^\perp$. Then

$$\begin{aligned} u_\perp &= U_\perp U_\perp^T u \\ &= U_\perp U_\perp^T (I - QQ^T)x \\ &= U_\perp U_\perp^T x \quad (\text{since } U_\perp^T X = 0) \\ &= x_\perp. \end{aligned} \tag{4.1}$$

In other words, multiplication by $I - P$ does not change the component of x along $\mathcal{R}(X)^\perp$.

To determine u_X we will construct a specific orthonormal basis for $\mathcal{R}(X)$. As above, let $W = Q^T Q - I$, and assume that $\omega = \|W\| < 1$. Then $I + W$ is positive definite and has a positive definite square root $(I + W)^{\frac{1}{2}}$. It follows that

$$U_X = Q(I + W)^{-\frac{1}{2}}$$

is orthonormal. Since the column spaces of Q and X are the same, U_X forms an orthonormal basis for the column space of X .

Now

$$\begin{aligned} u_X &= U_X U_X^T (I - QQ^T)x \\ &= U_X U_X^T [I - U_X (I + W)^{\frac{1}{2}} (I + W)^{\frac{1}{2}} U_X^T]x \quad (\text{since } (I + W)^{\frac{1}{2}} \text{ is symmetric}) \\ &= U_X [I - (I + W)] U_X^T x \\ &= U_X W U_X^T x \\ &= U_X W U_X^T x_X \quad (\text{since } U_X^T x_\perp = 0). \end{aligned}$$

It follows that

$$(4.2) \quad \|u_X\| \leq \omega \|x_X\|.$$

In other words, multiplication by $I - P$ reduces the component along $\mathcal{R}(X)$ by a factor of at least ω .

The consequence of all this is that the repeated application of the exact QGS step produces a sequence of vectors that converges to $x_\perp$ — or with normalization to $x_\perp/\|x_\perp\|$. The rate of convergence is that of the approach of ω^k to zero with increasing k . It remains to determine to what extent rounding error limits this convergence.

5. The effects of rounding error. We now assume that the QGS step (1.3) without reorthogonalization is computed in floating-point arithmetic with rounding unit ϵ_M . We assume that the arithmetic is standard in the sense that

$$\text{fl}(a \circ b) = (a \circ b)(1 + \epsilon), \quad |\epsilon| \leq |\epsilon_M|, \quad \circ = +, -, \times, \div,$$

where $\text{fl}(a \circ b)$ denotes the computed value of $a \circ b$.

Using standard techniques of rounding error analysis, we have the following relations among the computed values:

$$(5.1) \quad \begin{aligned} a &= (X + E)^T x, & \|E\| &\leq \langle \gamma \rangle \|X\| \epsilon_M, \\ r &= (R + F)^{-T} a, & \|F\| &\leq \langle \gamma \rangle \|R\| \epsilon_M, \\ b &= (R + G)^{-1} r, & \|G\| &\leq \langle \gamma \rangle \|R\| \epsilon_M, \\ \tilde{u} &= x - (X + H)b + h, & \|H\| &\leq \langle \gamma \rangle \|X\| \epsilon_M, \\ & & \|h\| &\leq \langle \gamma \rangle \|x\| \epsilon_M. \end{aligned}$$

Here we have assumed that $\|\tilde{u}\| \leq \|x\|$, which is reasonable in this context. Since by our scaling we have insured that the norms of X , R , and x are near one, we may absorb their norms into $\langle \gamma \rangle$ and use the simpler bounds

$$(5.2) \quad \|E\|, \|F\|, \|G\|, \|H\|, \|h\| \leq \langle \gamma \rangle \epsilon_M.$$

Let

$$(5.3) \quad \tilde{P} = (X + H)^T (R + G)^{-1} (R + F)^{-1} (X + E)^T,$$

so that

$$\tilde{u} = (I - \tilde{P})x + h.$$

Now up to second order terms in ϵ_M , we have

$$\begin{aligned} \tilde{P} &\cong XR^{-1}R^{-T}X^T \\ &\quad + HR^{-1}R^{-T}X^T \\ &\quad - XR^{-1}GR^{-1}R^{-T}X^T \\ &\quad - XR^{-1}R^{-T}FR^{-T}X^T \\ &\quad + XR^{-1}R^{-T}E^T. \end{aligned}$$

Setting

$$(5.4) \quad P = XR^{-1}R^{-T}X^T = QQ^T,$$

we may write

$$\tilde{P} \cong P + HR^{-1}Q^T - QGR^{-1}Q^T - QFR^{-T}Q + QR^{-T}E^T.$$

It follows that with $u = (I - P)x$ and $\tilde{u} = (I - \tilde{P})x + h$ we have

$$(5.5) \quad \tilde{u} = u + e,$$

where

$$(5.6) \quad \|e\| \leq \langle \gamma \rangle \|R^{-1}\| \|x\|_{\epsilon_M} = \alpha \|x\|,$$

with

$$(5.7) \quad \alpha = \langle \gamma \rangle \|R^{-1}\|_{\epsilon_M}.$$

(Note that α differs from $\hat{\alpha}$ in (3.1) only by the generic factor $\langle \gamma \rangle$.) Thus the effect of rounding error is to replace $u = (I - P)x$ by $u + e$, where $\|e\|$ is bounded by $\alpha \|x\|$.

6. Analysis. The two inequalities (4.2) and (5.5) lie at the heart of our analysis. The first says that ω attempts to orthogonalize by reducing x_X . The second says that α tries to spoil orthogonality by adding a small error. We are especially interested in the case $\omega \cong \alpha$. This condition says that the deviation from orthogonality is approximately equal to the condition of R times the rounding unit, and as such is an informal induction hypothesis for the beginning of a QGS step. For the nonce, however, we will treat ω and α separately. Keep in mind that ω and α are quite small—less than 10^{-8} in our examples (save for the breakdown in Example 3).

Our analysis is divided into two parts. The first treats the orthogonalization proper corresponding to the first four columns in our examples. The second treats the deviation in orthogonality as we update R —the last three columns in our example.

For the first part of our analysis set

$$u^0 = \tilde{u}^0 = x$$

and consider the vectors

$$\tilde{u}^{k+1} = (I - \tilde{P}^{(k)})\tilde{u}^k + h^k, \quad k = 0, 1, 2, \dots,$$

where $\tilde{P}^{(k)}$ is defined by (5.3). (The superscripts on P and h are reminders that the rounding errors are different for each step of the recurrence.) We are going to develop a recursion for $\tau(\tilde{u}^k)$. The passage from $\tau(\tilde{u}^0)$ to $\tau(\tilde{u}^1)$ is typical. Let

$$u^1 = (I - P)u^0,$$

where P is defined by (5.4). From (5.5) it follows that

$$\tilde{u}_X^1 = u_X^1 + e_X \quad \text{and} \quad \tilde{u}_\perp^1 = u_\perp^1 + e_\perp,$$

where by (5.6),

$$\|e_X\|, \|e_\perp\| \leq \alpha \|\tilde{u}^0\|.$$

Hence

$$\|\tilde{u}_X^1\| \leq \|u_X^1\| + \|e\| \leq \omega \|\tilde{u}_X^0\| + \alpha (\|\tilde{u}_X^0\| + \|\tilde{u}_\perp^0\|),$$

where the second inequality follows from (4.2). Similarly,

$$\|\tilde{u}_\perp^1\| \geq \|u_\perp^1\| - \|e\| \geq \|\tilde{u}_\perp^0\| - \alpha(\|\tilde{u}_X^0\| + \|\tilde{u}_\perp^0\|),$$

where the second inequality follows from (4.1). Hence

$$\tau(\tilde{u}^1) \leq \frac{\omega\|\tilde{u}_X^0\| + \alpha(\|\tilde{u}_X^0\| + \|\tilde{u}_\perp^0\|)}{\|\tilde{u}_\perp^0\| - \alpha(\|\tilde{u}_X^0\| + \|\tilde{u}_\perp^0\|)}.$$

On dividing the numerator and denominator by $\|u_\perp^0\|$, we find that

$$\tau(\tilde{u}^1) \leq \frac{\alpha + (\omega + \alpha)\tau(\tilde{u}^0)}{1 - \alpha - \alpha\tau(\tilde{u}^0)}.$$

Thus if we define $\tilde{\tau}^0 = \tau(x)$, then the recursion

$$(6.1) \quad \tilde{\tau}^{k+1} = \frac{\alpha + (\omega + \alpha)\tilde{\tau}^k}{1 - \alpha - \alpha\tilde{\tau}^k}, \quad k = 0, 1, \dots,$$

generates a sequence of upper bounds for the $\tau(\tilde{u}^k)$.

To analyze the behavior of these bounds, assume that

$$(6.2) \quad \alpha\tilde{\tau}^0 = \alpha\tau(x) < 0.1$$

and $\alpha + \omega < 1$. Then the sequence $\tilde{\tau}^k$ is strictly decreasing and converges to a fixed point of the function

$$\frac{\alpha + (\omega + \alpha)\tilde{\tau}}{1 - \alpha - \alpha\tilde{\tau}}$$

or, equivalently, to the smallest root of the quadratic equation

$$\tau^2 - \frac{1 - \omega - 2\alpha}{\alpha}\tau + 1 = 0.$$

If α is small, it is easily seen that this root must be approximately equal to α . The asymptotic rate of convergence is as the convergence of $(\omega + \alpha)^k$ to zero. The corresponding values of $\tau(\tilde{u}^k)$ must decrease at least as fast. The bounds suggest that the $\tau(\tilde{u}^k)$ will stagnate around α , although they cannot prove it. For example, a fortuitous cancellation might make some τ_k exactly zero, as is τ_2 in the first line of Example 3.

This part of the analysis answers the first two questions raised above. If we take $\omega = \alpha$, then the decrease of τ is as $(2\alpha)^k$. And the bounds suggest that orthogonalization will stop when $\tau \cong \alpha$.

The condition (6.2) explains why one reorthogonalization is all that is necessary. For it implies that $\tau(x) \leq 0.1\alpha^{-1}$. Our theory says that after two orthogonalizations the value of τ will be $\alpha^2\tau(x) = 0.1\alpha$, which is below the point of stagnation.

We now turn to the loss of orthogonality in passing from u_2 to q in the QGS step. With two steps of orthogonalization, we have computed the vector u_2 at step 8 in (1.3), which satisfies $\tau(u_2) \cong \alpha$. However, we do not use u_2/ρ as our q . Instead we define it as the last column of (1.4). Mathematically, this amounts to setting

$$(6.3) \quad u = x - XR^{-1}r = x - XR^{-1}(r_1 + r_2) \quad \text{and} \quad q = u/\rho,$$

where ρ is the computed value of $\|u_2\|$. Thus we must evaluate the degree of orthonormality of the matrix $(Q \ q)$. Let

$$(Q \ q)^T(Q \ q) = \begin{pmatrix} I + W & w \\ w^T & \nu \end{pmatrix}.$$

Since we know $\|W\| = \omega$, we must bound $\|w\|$ and $|1 - \nu|$.

Before we begin the analysis proper, let us introduce some notation and an assumption. Specifically, let

$$R_{\text{new}} = \begin{pmatrix} R & r_1 + r_2 \\ 0 & \rho \end{pmatrix} = \begin{pmatrix} R & r \\ 0 & \rho \end{pmatrix}.$$

Then

$$(6.4) \quad R_{\text{new}}^{-1} = \begin{pmatrix} R^{-1} & -\rho^{-1}R^{-1}r \\ 0 & \rho^{-1} \end{pmatrix}.$$

In what follows we will assume that

$$(6.5) \quad \|R^{-1}r_1\| + \|R^{-1}r_2\| \leq \langle \gamma \rangle \|R^{-1}r\|.$$

This says that there is no significant cancellation in the sum $R^{-1}r_1 + R^{-1}r_2$. We will discuss this assumption later.

From (5.1) and (5.2) we know that

$$u_2 = x - (X + H_1)(R + E_1)^{-1}r_1 + (X + H_2)(R + E_2)^{-1}r_2,$$

where $\|E_i\|, \|H_i\| \leq \langle \gamma \rangle \epsilon_M$. (We are ignoring the insignificant effects of h in the rounding error bounds.) It follows from (6.3) that the difference

$$\begin{aligned} g &= u - u_2 \cong XR^{-1}E_1R^{-1}r_1 + XR^{-1}E_2R^{-1}r_2 - H_1R^{-1}r_1 + H_2R^{-1}r_2 \\ &= Q(E_1R^{-1}r_1 + E_2R^{-1}r_2) - H_1R^{-1}r_1 + H_2R^{-1}r_2. \end{aligned}$$

Here we have used first order expansions of the form $(R + E^{-1}) \cong R^{-1} - R^{-1}ER^{-1}$ and dropped terms of $O(\epsilon_M^2)$. Remembering that $\|Q\|$ is near one, we have

$$\|g\| \leq \langle \gamma \rangle (\|R^{-1}r_1\| + \|R^{-1}r_2\|) \epsilon_M = \langle \gamma \rangle \|R^{-1}r\| \epsilon_M,$$

the last equality following from (6.5).

To bound $\|w\| = \|\rho^{-1}Q^T u\|$, observe that $\tau(u_2) = \|u_X^{(2)}\|/\|u_\perp^{(2)}\| \cong \alpha$, which we assume is small. This implies that

$$\|u_X^{(2)}\| \cong \alpha \|u_\perp^{(2)}\| \cong \alpha \|u_2\| = \alpha \rho.$$

On the other hand,

$$Q^T u = Q^T(u_2 + g) = Q^T(u_X^{(2)} + u_\perp^{(2)} + g) = Q^T(u_X^{(2)} + g).$$

Hence

$$\|Q^T u\| \leq \langle \gamma \rangle (\alpha \rho + \|R^{-1}r\| \epsilon_M),$$

and

$$(6.6) \quad \|w\| = \|Q^T u\|/\rho \leq \langle \gamma \rangle (\alpha + \rho^{-1} \|R^{-1} r\|_{\epsilon_M}) \leq \langle \gamma \rangle \|R_{\text{new}}^{-1}\|_{\epsilon_M}.$$

To bound $1 - \nu$, we first observe that since ρ is the computed value of $\|u_2\|$, we have

$$\rho^{-1} \|u_2\| = 1 + \langle \gamma \rangle \epsilon_M.$$

Hence ignoring terms of order ϵ_M^2 , we have

$$\begin{aligned} \nu &= \rho^{-2} (u_2 + g)^T (u_2 + g) \\ &= \rho^{-2} (u_2^T u_2 + 2u_2^T g + g^T g) \\ &= 1 + \langle \gamma \rangle \epsilon_M + 2\rho^{-2} u_2^T g + \rho^{-2} g^T g \\ &\cong 1 + \langle \gamma \rangle \epsilon_M + 2\rho^{-2} u_2^T g. \end{aligned}$$

Since $\rho^{-1} \|u\| \cong 1$, we have

$$(6.7) \quad |1 - \nu| \leq \langle \gamma \rangle (1 + \rho^{-1} \|R^{-1} r\|) \epsilon_M \leq \langle \gamma \rangle \|R_{\text{new}}^{-1}\|_{\epsilon_M}.$$

It now follows that if we define

$$\alpha_{\text{new}} = \langle \gamma \rangle \|R_{\text{new}}^{-1}\|_{\epsilon_M},$$

then the loss of orthogonality is bounded by α_{new} , which, since $\|R_{\text{new}}\| \cong 1$, is proportional to the condition number of R_{new} .

Turning now to the noncancellation assumption (6.5), we must first derive an expression for the size of r_1 . From (5.1) and (5.2),

$$\begin{aligned} r_1 &= (R + F)^{-T} (X + E)^T x \\ &\cong (R^{-T} - R^{-T} F^{-T} R^{-T}) (X + E)^T x \\ &\cong Q^T x_X + R^{-T} E^T x - R^{-T} F^T Q^T x. \end{aligned}$$

Since $\|R^{-T} E\|, \|R^{-T} F\| \leq \alpha$, and $\|Q^T x_X\| \cong \|x_X\|$, we have

$$\|r_1\| = \langle \gamma \rangle \|x_X\|,$$

whenever $\tau(x) > \alpha$.

We must now justify the assumption that $\|R^{-1} r\| \cong \|R^{-1} r_1\| + \|R^{-1} r_2\|$. Actually, we shall show that the assumption is satisfied if $\tau(x) > 1$ and that it is unnecessary if $\tau(x) \leq 1$.

First assume that $\tau(x) > 1$. Then one step of the QGS orthogonalization reduces x_X by a factor of α . Hence

$$(6.8) \quad \|r_2\| \leq \alpha \|r_1\|.$$

We will now show that (6.8) implies that there cannot be any cancellation in the sum

$$R^{-1} r_1 + R^{-1} r_2 \equiv s_1 + s_2.$$

We begin with a lower bound on s_1 . Since $\|R\| \cong 1$, we have

$$\|s_1\| \geq \langle \gamma \rangle \|r_1\|.$$

On the other hand,

$$\begin{aligned}\|s_2\| &\leq \|R^{-1}\| \|r_2\| \\ &\leq \alpha \|R^{-1}\| \|r_1\| && \text{by (6.8)} \\ &= \langle \gamma \rangle \epsilon_M \|R^{-1}\|^2 \|r_1\| && \text{by (5.7)}.\end{aligned}$$

Hence if

$$(6.9) \quad \langle \gamma \rangle \|R^{-1}\|^2 \epsilon_M \leq 0.1,$$

then the norm of s_2 is less than one-tenth of the norm of s_1 , and no cancellation can occur in the sum $s_1 + s_2$. We will discuss the implications of (6.9) in section 7.

Turning now to the case where $\tau(x) \leq 1$, we first note that in this case

$$\|x_X\| \leq \|x_\perp\| < 1 \quad \text{and} \quad \|x_\perp\| \geq \frac{1}{\sqrt{2}}.$$

The first of these inequalities implies that $\|r_1\|, \|r_2\| < 1$. The second implies that $\rho \geq \langle \gamma \rangle / \sqrt{2}$, since $\rho \cong \|x_\perp\|$. Hence the last column of R_{new}^{-1} is bounded by $\langle \gamma \rangle \|R^{-1}\|$ (see (6.4)). But $\|R^{-1}\|$ is a block of R_{new}^{-1} . Hence (6.6) and (6.7) continue to hold.

7. Discussion. The answer to the first two of the three questions in section 3 has already been given in the form of the recursion for the bounds $\tilde{\tau}^k$: namely, for the case $\omega = \alpha$,

$$\tilde{\tau}^{k+1} = \frac{\alpha + 2\alpha\tilde{\tau}^k}{1 - \alpha - \alpha\tilde{\tau}^k}, \quad k = 0, 1, \dots$$

If $\tilde{\tau}^k$ is larger than $(2\alpha)^{-1}$, then $\tilde{\tau}^{k+1}$ is essentially $(2\alpha)\tilde{\tau}^k$. On the other hand, if $\tilde{\tau}^k$ is less than $(2\alpha)^{-1}$, the additive term α in the numerator makes it impossible for the $\tilde{\tau}^k$ to ever become less than α . Thus the sequence $\tilde{\tau}^k$ decreases as $(2\alpha)^k$, as seen in our examples, until it nears α , after which it stagnates. As we have pointed out, the tangents $\tau(u^k)$ of the computed vectors may be smaller than the $\tilde{\tau}^k$; but this is a chance event.

The condition $\alpha\tau(x) < 0.1$ serves to start off recurrence (6.1) for the $\tilde{\tau}^k$ and to insure that τ^2 is near α , which implies that only one reorthogonalization is required to obtain the limiting orthogonality in the QGS step. However, the condition has additional significance. If $\alpha\tau(x) > 1$, then x_X is essentially one and $\|x_\perp\|$ is less than α . Consequently, $x_\perp$ will be swamped by the perturbation of size α generated by rounding error, making orthogonalization meaningless.

The loss of orthogonality in q is a consequence of the fact that the norm of the $(1, 2)$ -block of R_{new} (see (6.4)) appears in the bounds (6.6) and (6.7). In fact, there need be no loss of orthogonality. For if $\tau(x)$ is small, then ρ is near one, and $\|r\| \cong \sqrt{1 - \rho^2}$ will be small. Hence $\rho^{-1}\|R^{-1}r\|$ will be bounded by a multiple of $\|R\|$, and the condition of R_{new} will be comparable to the condition of R .

If we rewrite the condition (6.9) in the form

$$\langle \gamma \rangle \kappa(R) \leq \epsilon_M^{-\frac{1}{2}},$$

we see that the QGS method can be expected to work in IEEE double precision only when $\kappa(R)$ is less than roughly 10^8 . An analogous breakdown occurs when one tries to compute the Cholesky decomposition of $X^T X$. In applications where well-conditioned

reduced-rank approximations are needed, the restriction is not onerous, since we will stop the QGS process long before $\kappa(R)$ is near 10^8 . It is worth noting that when QGS is applied with column pivoting for size, the algorithm for choosing a pivot column breaks down at the same point.

Finally, it is important to keep in mind that we have analyzed only a single step of the QGS algorithm. It is possible that when we combine the bounds from step to step, there will be an undue growth in the combined bounds.⁵ I think this unlikely, given the self-correcting nature of the reorthogonalization and the small size of α . But to verify this rigorously would require a refined (and much longer) analysis that would take into consideration the actual values of the generic constants represented by $\langle\gamma\rangle$.

REFERENCES

- [1] M. W. BERRY, Z. DRMAČ, AND E. R. JESSUP, *Matrices, vector spaces, and information retrieval*, SIAM Rev. 41 (1999), pp. 335–362.
- [2] M. W. BERRY AND D. I. MARTIN, *Principal component analysis for information retrieval*, in Handbook of Parallel Computing and Statistics, Marcel Dekker, New York, 2004.
- [3] M. W. BERRY, S. A. PULATOVA, AND G. W. STEWART, *Computing sparse reduced-rank approximations to sparse matrices*, ACM Trans. Math. Software, 31 (2005), pp. 252–269.
- [4] A. BJÖRCK, *Stability analysis of the method of seminormal equations for linear squares problems*, Linear Algebra Appl., 88/89 (1987), pp. 31–48.
- [5] A. BJÖRCK, *Numerical Methods for Least Squares Problems*, SIAM, Philadelphia, 1996.
- [6] T. F. CHAN, *Rank revealing QR factorizations*, Linear Algebra Appl., 88/89 (1987), pp. 67–82.
- [7] W. HOFFMANN, *Iterative algorithms for Gram–Schmidt orthogonalization*, Computing, 41 (1989), pp. 335–348.
- [8] P. JIANG AND M. W. BERRY, *Solving total least squares problems in information retrieval*, Linear Algebra Appl., 316 (2000), pp. 137–156.
- [9] G. W. STEWART, *Matrix Algorithms: Basic Decompositions*, SIAM, Philadelphia, 1998.
- [10] G. W. STEWART, *Four algorithms for the efficient computation of truncated pivoted QR approximations to a sparse matrix*, Numer. Math., 83 (1999), pp. 313–323.
- [11] G. W. STUART AND M. W. BERRY, *A comprehensive whole genome bacterial phylogeny using correlated peptide motifs defined in a high dimensional vector space*, J. Bioinfo. Comput. Biol., 1 (2003), pp. 475–493.
- [12] A. VAN DER SLUIS, *Condition numbers and equilibration of matrices*, Numer. Math., 14 (1969), pp. 14–23.

⁵Such a growth occurs in Tony Chan’s groundbreaking paper on rank-revealing QR factorizations [6].

UNITARILY INVARIANT METRICS ON THE GRASSMANN SPACE*

LI QIU[†], YANXIA ZHANG[‡], AND CHI-KWONG LI[§]

Abstract. Let $\mathcal{G}_{m,n}$ be the Grassmann space of m -dimensional subspaces of $\mathbb{F}^n$. Denote by $\theta_1(\mathcal{X}, \mathcal{Y}), \dots, \theta_m(\mathcal{X}, \mathcal{Y})$ the canonical angles between subspaces $\mathcal{X}, \mathcal{Y} \in \mathcal{G}_{m,n}$. It is shown that $\Phi(\theta_1(\mathcal{X}, \mathcal{Y}), \dots, \theta_m(\mathcal{X}, \mathcal{Y}))$ defines a unitarily invariant metric on $\mathcal{G}_{m,n}$ for every symmetric gauge function Φ . This provides a wide class of new metrics on $\mathcal{G}_{m,n}$. Some related results on perturbation and approximation of subspaces in $\mathcal{G}_{m,n}$, as well as the canonical angles between them, are also discussed. Furthermore, the equality cases of the triangle inequalities for several unitarily invariant metrics are analyzed.

Key words. subspace, unitarily invariant metric, canonical angles, perturbation, singular values

AMS subject classifications. 15A18, 15A03

DOI. 10.1137/040607605

1. Introduction. Let $\mathbb{F}$ be the real field $\mathbb{R}$ or the complex field $\mathbb{C}$. Denote by $\mathbb{F}^n$ the linear space of $n \times 1$ vectors with entries in $\mathbb{F}$. Suppose $\mathcal{X}$ and $\mathcal{Y}$ are one-dimensional subspaces of $\mathbb{F}^n$ spanned by the unit vectors x and y , respectively. There are natural metrics to measure the distance between $\mathcal{X}$ and $\mathcal{Y}$.

- (a) The acute angle between $\mathcal{X}$ and $\mathcal{Y}$, i.e., $\theta(\mathcal{X}, \mathcal{Y}) = \arccos |x^H y| \in [0, \pi/2]$.
- (b) The gap metric between $\mathcal{X}$ and $\mathcal{Y}$ defined as $\sin \theta(\mathcal{X}, \mathcal{Y}) = \inf \{\|x - yq\| : q \in \mathbb{F}\}$.
- (c) The Hausdorff metric between $\mathcal{X}$ and $\mathcal{Y}$ defined as

$$2 \sin(\theta(\mathcal{X}, \mathcal{Y})/2) = \inf \{\|x - yu\| : u \in \mathbb{F}, |u| = 1\}.$$

To gain more geometrical insight, let us temporarily specialize these measures to real Euclidean space $\mathbb{R}^n$. It is clear that the quantity in (a) is in fact the acute angle between the lines $\mathcal{X}$ and $\mathcal{Y}$. The quantity in (b) represents the distance of any unit vector in one line, either x or $-x$, to the other line $\mathcal{Y}$; it also turns out to be the difference between the orthogonal projections onto $\mathcal{X}$ and $\mathcal{Y}$, respectively, i.e., $\|xx^H - yy^H\|$, measured by the operator norm. The quantity in (c) represents the Hausdorff distance between the set of unit vectors in one line, $\{x, -x\}$, and that in the other line, $\{y, -y\}$.

It is well known and not hard to check that each of the measures in (a), (b), and (c) indeed defines a metric ρ on the set $\mathcal{G}_{1,n}$ of one-dimensional subspaces of $\mathbb{F}^n$, i.e., for any one-dimensional subspaces $\mathcal{X}, \mathcal{Y}, \mathcal{Z}$ of $\mathbb{F}^n$, we have

- (i) $\rho(\mathcal{X}, \mathcal{Y}) \geq 0$; the equality holds if and only if $\mathcal{X} = \mathcal{Y}$;

*Received by the editors April 29, 2004; accepted for publication (in revised form) by R. Mathias March 30, 2005; published electronically November 22, 2005.

<http://www.siam.org/journals/simax/27-2/60760.html>

[†]Department of Electrical and Electronic Engineering, Hong Kong University of Science and Technology, Clear Water Bay, Kowloon, Hong Kong, China (eeqiu@ust.hk). The research of this author was supported by the Hong Kong Research Grants Council through grant HKUST6032/01E.

[‡]Department of Mathematics, Beijing Institute of Technology, Beijing 100081, China (yxzhang@bit.edu.cn). The research of this author was partially supported by the Hong Kong Research Grants Council through grant HKUST6032/01E.

[§]Department of Mathematics, College of William and Mary, Williamsburg, VA 23187-8795 (ckli@math.wm.edu). The research of this author was supported by the National Science Foundation through grant DMS-0071994 and the Hong Kong Research Grants Council through grant 10207475.

- (ii) $\rho(\mathcal{X}, \mathcal{Y}) = \rho(\mathcal{Y}, \mathcal{X})$;
- (iii) $\rho(\mathcal{X}, \mathcal{Z}) \leq \rho(\mathcal{X}, \mathcal{Y}) + \rho(\mathcal{Y}, \mathcal{Z})$.

Furthermore, each of these metrics is unitarily invariant in the sense that

- (iv) $\rho(U\mathcal{X}, U\mathcal{Y}) = \rho(\mathcal{X}, \mathcal{Y})$ for any unitary transformation U on $\mathbb{F}^n$.

In a certain sense, the metric defined in (a) is superior to those in (b) or (c) for the following reasons.

First, for any positive subadditive nondecreasing function $f : [0, \pi/2] \rightarrow [0, \infty)$, i.e., any function f satisfying $f(\alpha) > 0$ for $\alpha \in (0, \pi/2]$, $f(\alpha + \beta) \leq f(\alpha) + f(\beta)$ for any $\alpha, \beta \in [0, \pi/2]$ with $\alpha + \beta \leq \pi/2$, and $f(\alpha) \leq f(\beta)$ for $\alpha \leq \beta$, the function $(\mathcal{X}, \mathcal{Y}) \rightarrow f(\theta(\mathcal{X}, \mathcal{Y}))$ is again a metric on $\mathcal{G}_{1,n}$. Proper choice of such an f generates a wide class of metrics. Taking $f(\alpha) = \sin \alpha$ and $f(\alpha) = 2 \sin(\alpha/2)$, one obtains the metrics in (b) and (c) from that of (a). One might also take, for example, $f(\alpha) = \ln(1 + \alpha)$ or $f(\alpha) = \tan(\alpha/2)$.

Second, for $\rho(\mathcal{X}, \mathcal{Y}) = \theta(\mathcal{X}, \mathcal{Y})$, the equality in (iii) holds for $\mathcal{X}, \mathcal{Y}, \mathcal{Z} \in \mathcal{G}_{1,n}$ if and only if there are unit vectors x, y, z in $\mathcal{X}, \mathcal{Y}, \mathcal{Z}$ such that $y = [tx + (1-t)z] / \|tx + (1-t)z\|$ for some $t \in [0, 1]$ and $x^*z \geq 0$. Geometrically, it means that $V = \text{span}\{x, y, z\}$ is two-dimensional and the vector y lies on the circular arc with length not larger than $\pi/2$ joining the two vectors x and z in V . This is most useful when we consider problems involving small perturbation of subspaces. In contrast, for the metrics defined in (b) and (c), the equality in (iii) holds only in the trivial case when $\mathcal{X} = \mathcal{Y}$ or $\mathcal{Y} = \mathcal{Z}$.

The main purpose of this paper is to extend the nice features of the metric $\theta(\mathcal{X}, \mathcal{Y})$ on $\mathcal{G}_{1,n}$ to the Grassmann space $\mathcal{G}_{m,n}$ of m -dimensional subspaces of $\mathbb{F}^n$ for $1 \leq m \leq n$. It is well known that between two members of $\mathcal{G}_{m,n}$, there are m angles, called canonical angles or principal angles. We will define a family of metrics using directly the canonical angles, instead of some trigonometrical functions of the canonical angles as in the current common practice. This strengthens the fundamental role played by the canonical angles in studying the relationship between subspaces. We will then establish some results in the perturbation analysis of the canonical angles and in the robustness of certain geometric relations, namely the nullity (dimension of the intersection) and deficiency (codimension of the sum), between a pair of subspaces using the new metrics defined. We will justify that the new family of metrics do have advantages over the existing ones.

The demand for a distance measure between linear subspaces arises in countless occasions, ranging from pure mathematics to engineering applications. Examples include perturbation theory of unbounded linear operators [16], perturbation analysis of invariant subspaces, deflating subspaces in generalized eigenvalue problems, and singular subspaces in singular value decompositions [3], [27], [28], optimization [6], robust control [9], [24], [25], [33], geometric approach to linear multivariable control [17], system identification [32], signal processing [4], [21], and so on. Our study was mainly motivated by robust control of linear feedback systems. Roughly speaking a linear time-invariant system can be described by a subspace valued frequency function, and the description of an uncertain system calls for a suitable distance measure between subspaces. The stability of a feedback system can be restated as the complement property of two frequency dependent subspaces. Hence the robust stability deals with the robustness of subspace complement, motivating the study of robustness of subspace nullity and deficiency, which is also done in this paper. Not all the distance measures used in the aforementioned works are metrics, i.e., some of them do not satisfy the triangle inequality, but metrics are certainly preferred from both theoretical and practical view points. Indeed most of the measures used are metrics

and most of them are defined from certain trigonometrical functions of the canonical angles between the subspaces. Among the possible metrics, the ones with tight triangle inequalities are most desirable, since they give the tightest perturbation bounds, best estimation of errors, etc. The main ramification of this paper is that the newly defined family of metrics does offer the desired advantages.

In addition to being used to measure the distance between subspaces, the canonical angles also have other utilities. Notably, they can give all the singular values of an oblique projection (Gohberg and Krein [10] related the largest singular value with the smallest canonical angle, but the relation can be extended to all singular values); they are used to study the volumes of matrices [23]; they are used in statistics to define the so-called canonical correlations [13], which have attracted further mathematical treatment [11] and have wide applications (see those mentioned in [11]); and they are used in characterizing other geometric relations between subspaces such as nullity, deficiency, complementarity, and so on [16], [37]. Therefore, the perturbation analysis of the canonical angles themselves is of importance and has been conducted in the works [11], [18], [29]. We will carry out the analysis using the newly defined family of metrics.

In the next section, we present the theory of canonical angles on a pair of subspaces in $\mathcal{G}_{m,n}$ and we develop some relationships among the canonical angles between three different subspaces, which constitute the key technical vehicle for the proof of our main results. In section 3, we prove that any symmetric gauge function of the canonical angles defines a unitarily invariant metric; we also prove several results on approximation and perturbation of subspaces as well as canonical angles using the newly defined metrics. In section 4, we analyze and compare the equality cases of the triangular inequalities for several unitarily invariant metrics. Further results and extensions will be discussed in section 5.

We will use $\mathcal{M}_{m,n}(\mathbb{F})$ (respectively, $\mathcal{M}_n(\mathbb{F})$) to denote the linear space of $m \times n$ (respectively, $n \times n$) matrices over $\mathbb{F}$. The group of complex unitary or real orthogonal matrices is denoted by $\mathcal{U}_n(\mathbb{F})$. We will identify a matrix in $\mathcal{M}_n(\mathbb{F})$ and a linear transformation on $\mathbb{F}^n$. We often abbreviate the notation to $\mathcal{M}_{m,n}$, $\mathcal{M}_n$, and $\mathcal{U}_n$ if the statements are valid for both $\mathbb{F} = \mathbb{C}$ or $\mathbb{R}$.

2. Canonical angles between subspaces. The *singular values* of $A \in \mathcal{M}_{m,n}$ are the nonnegative square roots of the l largest eigenvalues of positive semidefinite matrix $A^H A$, where $l = \min\{m, n\}$, and are denoted by $\sigma_1(A) \geq \cdots \geq \sigma_l(A)$. We also use the more compact notation $\sigma(A)$ to denote the decreasingly ordered l -tuple of singular values of A .

Let $\mathcal{X}, \mathcal{Y} \in \mathcal{G}_{m,n}$. Suppose the columns of $X_1, Y_1 \in \mathcal{M}_{n,m}$ form orthonormal bases for $\mathcal{X}$ and $\mathcal{Y}$, respectively. Then the singular values of $X_1^H Y_1$ lie in $[0, 1]$ and are independent of the particular choices of the orthonormal bases. We define the *canonical angles* between $\mathcal{X}$ and $\mathcal{Y}$ to be

$$\theta_i(\mathcal{X}, \mathcal{Y}) = \arccos \sigma_{m-i+1}(X_1^H Y_1) \in [0, \pi/2], \quad i = 1, 2, \dots, m.$$

In the following, we denote the decreasingly ordered m -tuple of canonical angles between $\mathcal{X}$ and $\mathcal{Y}$ by

$$\theta(\mathcal{X}, \mathcal{Y}) := (\theta_1(\mathcal{X}, \mathcal{Y}), \dots, \theta_m(\mathcal{X}, \mathcal{Y})).$$

Let $\mathcal{X}_\perp$ and $\mathcal{Y}_\perp$ be the orthogonal complements of $\mathcal{X}$ and $\mathcal{Y}$, respectively. Suppose $X_2, Y_2 \in \mathcal{M}_{n,n-m}$ have columns forming orthonormal bases for $\mathcal{X}_\perp$ and $\mathcal{Y}_\perp$,

respectively. Then $X = [X_1 \ X_2]$ and $Y = [Y_1 \ Y_2]$ are unitary matrices. By the CS decomposition of matrices in $\mathcal{U}_n$ (e.g., see Theorem 5.2 in [27]), there are $E = \text{diag}(E_1, E_2), F = \text{diag}(F_1, F_2) \in \mathcal{U}_n$ with $E_1, F_1 \in \mathcal{U}_m$ such that

$$E^H X^H Y F = \text{diag} \left[I_{m-l}, \begin{bmatrix} \Gamma & -\Sigma \\ \Sigma & \Gamma \end{bmatrix}, I_{n-m-l} \right],$$

where $l = \min\{m, n-m\}$, Γ and Σ are diagonal matrices with diagonal entries in $[0, 1]$ satisfying $\Gamma^2 + \Sigma^2 = I_l$. Hence the diagonal entries of Γ are the cosines of the first l canonical angles between $\mathcal{X}$ and $\mathcal{Y}$; the diagonal entries of Σ are the sines of the first l canonical angles. The rest $m-l$ canonical angles between $\mathcal{X}$ and $\mathcal{Y}$ are always zero. This leads to several observations:

- (1) The nonzero canonical angles between $\mathcal{X}$ and $\mathcal{Y}$ are the same as those of $\mathcal{X}_\perp$ and $\mathcal{Y}_\perp$.
- (2) The singular values of $X_2^H Y_1$ are $\sin \theta_1(\mathcal{X}, \mathcal{Y}) \geq \cdots \geq \sin \theta_l(\mathcal{X}, \mathcal{Y})$, and so are those of $X_1^H Y_2$.
- (3) The canonical angles are unitarily invariant, i.e., $\theta(U\mathcal{X}, U\mathcal{Y}) = \theta(\mathcal{X}, \mathcal{Y})$ for all $U \in \mathcal{U}_n$, since $(UX_1)^H(UY_1) = X_1^H Y_1$.

In view of the above observations, we can always assume $m \leq n/2$ in our discussion; otherwise, we can consider the canonical angles between $\mathcal{X}_\perp$ and $\mathcal{Y}_\perp$ instead.

Jordan [15] discovered the canonical angles shortly after he discovered the matrix singular values [14]. In statistics, Hotelling [13] used canonical angles to define the canonical correlations shortly after he used the singular values to define the principal components [12]. Also see, for example, [1], [3], [35].

The CS decomposition also yields a special unitary transformation mapping $\mathcal{X}$ to $\mathcal{Y}$, called a direct rotation from $\mathcal{X}$ to $\mathcal{Y}$ [3]. Set $\hat{X} = XE$ and $\hat{Y} = YF$. We have

$$(2.1) \quad \hat{X}^H \hat{Y} = \begin{bmatrix} \Gamma & -\Sigma & 0 \\ \Sigma & \Gamma & 0 \\ 0 & 0 & I_{n-2m} \end{bmatrix}.$$

A direct rotation from $\mathcal{X}$ to $\mathcal{Y}$ is given by

$$(2.2) \quad R = \hat{X} \begin{bmatrix} \Gamma & -\Sigma & 0 \\ \Sigma & \Gamma & 0 \\ 0 & 0 & I_{n-2m} \end{bmatrix} \hat{X}^H = \hat{X} \exp \begin{bmatrix} 0 & -A \\ A^H & 0 \end{bmatrix} \hat{X}^H,$$

where $A = [\text{diag} \theta(\mathcal{X}, \mathcal{Y}) \ 0_{m, n-2m}]$. It is shown in [3] that among all unitary transformations mapping $\mathcal{X}$ to $\mathcal{Y}$, the direct rotation is the “most economic” in some sense. We will revisit this point in section 5.

Similarly, if we have another subspace $\mathcal{Z} \in \mathcal{G}_{m, n}$, then a direct rotation from $\mathcal{Y}$ to $\mathcal{Z}$ is given by

$$S = \bar{Y} \exp \begin{bmatrix} 0 & -B \\ B^H & 0 \end{bmatrix} \bar{Y}^H,$$

where $B = [\text{diag} \theta(\mathcal{Y}, \mathcal{Z}) \ 0_{m, n-2m}]$, $\bar{Y} = [\bar{Y}_1 \ \bar{Y}_2] \in \mathcal{U}_n$, and the columns of $\bar{Y}_1$ and $\bar{Y}_2$ form special orthonormal bases of $\mathcal{Y}$ and $\mathcal{Y}_\perp$, respectively. Also a direct rotation mapping $\mathcal{X}$ to $\mathcal{Z}$ is given by

$$T = \tilde{X} \exp \begin{bmatrix} 0 & -C \\ C^H & 0 \end{bmatrix} \tilde{X}^H,$$

where $C = [\text{diag } \theta(\mathcal{X}, \mathcal{Z}) \quad 0_{m, n-2m}]$, $\tilde{X} = [\tilde{X}_1 \quad \tilde{X}_2] \in \mathcal{U}_n$, and the columns of $\tilde{X}_1$ and $\tilde{X}_2$ form special orthonormal bases of $\mathcal{X}$ and $\mathcal{X}_\perp$, respectively.

Since SR also takes $\mathcal{X}$ onto $\mathcal{Z}$, it can only differ from T locally in $\mathcal{X}$ and $\mathcal{X}_\perp$ [3], i.e., there exists unitary matrix $Q = \text{diag}(Q_1, Q_2)$ with $Q_1 \in \mathcal{U}_m$ and $Q_2 \in \mathcal{U}_{n-m}$ such that

$$\tilde{X} \exp \begin{bmatrix} 0 & -C \\ C^H & 0 \end{bmatrix} Q \tilde{X}^H = SR,$$

which is equivalent to

$$\exp \begin{bmatrix} 0 & -C \\ C^H & 0 \end{bmatrix} Q = \tilde{X}^H \bar{Y} \exp \begin{bmatrix} 0 & -B \\ B^H & 0 \end{bmatrix} \bar{Y}^H \tilde{X} \tilde{X}^H \hat{X} \exp \begin{bmatrix} 0 & -A \\ A^H & 0 \end{bmatrix} \hat{X}^H \tilde{X}.$$

Denote $\tilde{X}^H \bar{Y}$ by M and $\tilde{X}^H \hat{X}$ by N . Notice that $N = \text{diag}(N_1, N_2)$ for some $N_1 \in \mathcal{U}_m$ and $N_2 \in \mathcal{U}_{n-m}$. Hence

$$(2.3) \quad \exp \begin{bmatrix} 0 & -C \\ C^H & 0 \end{bmatrix} Q = M \exp \begin{bmatrix} 0 & -B \\ B^H & 0 \end{bmatrix} M^H N \exp \begin{bmatrix} 0 & -A \\ A^H & 0 \end{bmatrix} N^H.$$

Taking inverse, we get

$$Q^H \exp \begin{bmatrix} 0 & C \\ -C^H & 0 \end{bmatrix} = N \exp \begin{bmatrix} 0 & A \\ -A^H & 0 \end{bmatrix} N^H M \exp \begin{bmatrix} 0 & B \\ -B^H & 0 \end{bmatrix} M^H.$$

Multiplying $J = \text{diag}(I_m, -I_{n-m})$ from both sides and noticing that J commutes with both Q and N , we get

$$(2.4) \quad Q^H \exp \begin{bmatrix} 0 & -C \\ C^H & 0 \end{bmatrix} = N \exp \begin{bmatrix} 0 & -A \\ A^H & 0 \end{bmatrix} N^H J M \exp \begin{bmatrix} 0 & B \\ -B^H & 0 \end{bmatrix} M^H J.$$

Multiplying (2.3) and (2.4), we obtain

$$\begin{aligned} & \exp \begin{bmatrix} 0 & -2C \\ 2C^H & 0 \end{bmatrix} \\ &= M \exp \begin{bmatrix} 0 & -B \\ B^H & 0 \end{bmatrix} M^H N \exp \begin{bmatrix} 0 & -2A \\ 2A^H & 0 \end{bmatrix} N^H J M \exp \begin{bmatrix} 0 & B \\ -B^H & 0 \end{bmatrix} M^H J \\ &= \exp \left[M \begin{bmatrix} 0 & -B \\ B^H & 0 \end{bmatrix} M^H \right] \exp \left[N \begin{bmatrix} 0 & -2A \\ 2A^H & 0 \end{bmatrix} N^H \right] \\ & \quad \times \exp \left[J M \begin{bmatrix} 0 & B \\ -B^H & 0 \end{bmatrix} M^H J \right]. \end{aligned}$$

The product of matrix exponentials on the right-hand side can be transformed to the exponential of a sum by using a matrix exponential formula of Thompson [31] (also see [2]), which asserts that for skew-Hermitian matrices $S, T \in \mathcal{M}_n$, there exist unitary matrices $U, V \in \mathcal{U}_n(\mathbb{C})$ such that

$$\exp(S) \exp(T) = \exp(USU^H + VTV^H).$$

This leads to the following relation of the canonical angles.

PROPOSITION 2.1. *There exist $U_1, U_2, V \in \mathcal{U}_n(\mathbb{C})$ such that*

$$(2.5) \quad \exp \begin{bmatrix} 0 & -2C \\ 2C^H & 0 \end{bmatrix} \\ = \exp \left[V \begin{bmatrix} 0 & -2A \\ 2A^H & 0 \end{bmatrix} V^H + U_1 \begin{bmatrix} 0 & -B \\ B^H & 0 \end{bmatrix} U_1^H + U_2 \begin{bmatrix} 0 & B \\ -B^H & 0 \end{bmatrix} U_2^H \right].$$

Note that even when $\mathbb{F} = \mathbb{R}$, we have the above relations of canonical angles among three subspaces in terms of complex skew-Hermitian and unitary matrices.

3. Unitarily invariant metrics. A norm $\|\cdot\|$ on $\mathcal{M}_{m,n}$ is *unitarily invariant* if $\|UAV\| = \|A\|$ for any $U \in \mathcal{U}_m$, $V \in \mathcal{U}_n$, and $A \in \mathcal{M}_{m,n}$. It is immediately seen from the singular value decomposition that a unitarily invariant norm on $\mathcal{M}_{m,n}$ depends only on the singular values of matrices.

Unitarily invariant norms are intimately related to symmetric gauge functions. A *symmetric gauge function* $\Phi : \mathbb{R}^l \rightarrow \mathbb{R}$ is a norm function satisfying the additional properties that it is symmetric, i.e.,

$$\Phi(P\xi) = \Phi(\xi)$$

for any $\xi \in \mathbb{R}^l$ and permutation matrix P , and that it is absolute, i.e., $\Phi(|\xi|) = \Phi(\xi)$. The absolute property can be replaced by the monotone property, i.e., $\Phi(|\xi|) \leq \Phi(|\zeta|)$ if $|\xi| \leq |\zeta|$ in an elementwise sense. This follows from the well-known fact that a norm function is absolute if and only if it is monotone. A particular class of symmetric gauge functions, called Ky Fan functions [7], plays an important role. Let $1 \leq k \leq l$. The Ky Fan k -function is defined as

$$\Phi_k(\xi_1, \dots, \xi_l) = \max_{1 \leq i_1 < \dots < i_k \leq l} \{|\xi_{i_1}| + |\xi_{i_2}| + \dots + |\xi_{i_k}|\}.$$

The reason why the class of Ky Fan k -functions is special lies in the fact that for a pair of vectors $\xi, \zeta \in \mathbb{R}^l$, $\Phi_k(\xi) \leq \Phi_k(\zeta)$ for all k implies that $\Phi(\xi) \leq \Phi(\zeta)$ for all symmetric gauge functions Φ . When $\Phi_k(\xi) \leq \Phi_k(\zeta)$ for all $1 \leq k \leq l$, we say that ζ weakly majorizes ξ .

Unitarily invariant norms as well as their interplay with symmetric gauge functions have been very well understood. We list a few results that serve as models for our development of unitarily invariant metrics. For a more complete coverage, see [27].

- (I) (See [34].) There is a one-to-one correspondence between unitarily invariant norms $\|\cdot\|$ on $\mathcal{M}_{m,n}$ and symmetric gauge functions $\Phi : \mathbb{R}^l \rightarrow \mathbb{R}$, where $l = \min\{m, n\}$, given by

$$\|A\| = \Phi(\sigma(A)).$$

- (II) (See [22].) For the unitarily invariant norm $\|\cdot\|$ on $\mathcal{M}_{m,n}$ corresponding to the symmetric gauge function Φ ,

$$\Phi(\sigma(A) - \sigma(B)) \leq \|A - B\|$$

for all $A, B \in \mathcal{M}_{m,n}$.

- (III) (See [5], [22], [26].) Let $\|\cdot\|$ be a unitarily invariant norm on $\mathcal{M}_{m,n}$, and $1 \leq k \leq l = \min\{m, n\}$. Suppose $A \in \mathcal{M}_{m,n}$. Then

$$\min\{\|A - B\| : B \in \mathcal{M}_{m,n}, \text{rank}(B) \leq k\} = \Phi(0, \dots, 0, \sigma_{k+1}(A), \dots, \sigma_l(A)).$$

Specializing the triangle inequality of the norms in (I) to Ky Fan k -functions yields that for any $A, B, C \in \mathcal{M}_{m,n}$ with $C = A + B$,

$$(3.1) \quad \sum_{j=1}^k \sigma_j(C) \leq \sum_{j=1}^k \sigma_j(A) + \sum_{j=1}^k \sigma_j(B).$$

Specializing the Mirsky result (II) to Ky Fan k -functions yields that for any $A, B, C \in \mathcal{M}_{m,n}$ with $C = A + B$,

$$(3.2) \quad \sum_{j=1}^k \sigma_{i_j}(C) \leq \sum_{j=1}^k \sigma_{i_j}(A) + \sum_{j=1}^k \sigma_j(B)$$

for any $1 \leq i_1 < \dots < i_k \leq l = \min\{m, n\}$. In this sense, the inequalities in (II) are more general than the triangle inequalities of the norms in (I). Recently, the complete set of inequalities relating the singular values of matrices $A, B, C \in \mathcal{M}_{m,n}$ with $C = A + B$ has been determined; e.g., see [8].

One can use symmetric gauge functions to define unitarily invariant metrics on $\mathcal{G}_{m,n}$. Suppose $\Phi : \mathbb{R}^m \rightarrow \mathbb{R}$ is a symmetric gauge function and $\|\cdot\|$ is the corresponding unitarily invariant norm on $\mathcal{M}_{n,m}$. For $\mathcal{X}, \mathcal{Y} \in \mathcal{G}_{m,n}$, let $X_1, Y_1 \in \mathcal{M}_{n,m}$ have columns forming orthonormal bases for $\mathcal{X}$ and $\mathcal{Y}$. Two families of frequently used unitarily invariant metrics between $\mathcal{X}$ and $\mathcal{Y}$ are the *gap metrics*

$$(3.3) \quad \Phi(\sin \theta_1(\mathcal{X}, \mathcal{Y}), \dots, \sin \theta_m(\mathcal{X}, \mathcal{Y})) = \inf\{\|X_1 - Y_1 Q\| : Q \in \mathcal{M}_m\}$$

and the Hausdorff metrics

$$(3.4) \quad \Phi(2 \sin(\theta_1(\mathcal{X}, \mathcal{Y})/2), \dots, 2 \sin(\theta_m(\mathcal{X}, \mathcal{Y})/2)) = \inf\{\|X_1 - Y_1 V\| : V \in \mathcal{U}_m\},$$

which extend the metrics (b) and (c) for the case when $m = 1$ introduced in section 1. As we have shown in the case when $m = 1$, it is desirable to have an extension of metric (a). Potentially, the metrics defined by the canonical angles, instead of their sines or other functions, can be used to generate other metrics and have tighter triangle inequalities. It appears to be a long-established fact in the differential geometry literature [6], [38] that

$$\sqrt{\theta_1^2(\mathcal{X}, \mathcal{Y}) + \dots + \theta_m^2(\mathcal{X}, \mathcal{Y})}$$

is a metric. In [35], Wedin showed that the largest canonical angle $\theta_1(\mathcal{X}, \mathcal{Y})$ is a metric. Qiu and Davison [25] and Vinnicombe [33] also gave alternative proofs for this result in the engineering literature and demonstrated that the use of $\theta_1(\mathcal{X}, \mathcal{Y})$ as a metric is crucial in engineering applications. Recently, this fact was connected to a Finsler geometry of the Grassmann manifold [36]. It has been conjectured that any symmetric gauge function of the canonical angles is a metric. This conjecture was explicitly announced by the first author at the 10th ILAS Conference in 2002. Using the results in the last section, we are now able to prove the conjecture.

THEOREM 3.1. *Let $\Phi : \mathbb{R}^m \rightarrow \mathbb{R}$ be a symmetric gauge function. Define $\rho : \mathcal{G}_{m,n} \times \mathcal{G}_{m,n} \rightarrow \mathbb{R}$ by*

$$\rho(\mathcal{X}, \mathcal{Y}) = \Phi(\theta(\mathcal{X}, \mathcal{Y})).$$

Then ρ is a unitarily invariant metric.

Proof. Evidently, $\rho(\mathcal{X}, \mathcal{Y})$ is unitarily invariant and satisfies the positivity condition (i) and the symmetry condition (ii) in section 1. It remains to show that it satisfies the triangle inequality (iii). As shown in section 2, the canonical angles between subspaces $\mathcal{X}, \mathcal{Y}, \mathcal{Z}$ satisfy the relationship (2.5). Hence the eigenvalues of

$$(3.5) \quad \begin{bmatrix} 0 & -2C \\ 2C^H & 0 \end{bmatrix}$$

and those of

$$(3.6) \quad V \begin{bmatrix} 0 & -2A \\ 2A^H & 0 \end{bmatrix} V^H + U_1 \begin{bmatrix} 0 & -B \\ B^H & 0 \end{bmatrix} U_1^H + U_2 \begin{bmatrix} 0 & B \\ -B^H & 0 \end{bmatrix} U_2^H$$

are equal modulo $\pm i2\pi$, where $i = \sqrt{-1}$. Since the eigenvalues of (3.5) belong to $i[-\pi, \pi]$ and those of (3.6) are known to belong to $i[-2\pi, 2\pi]$, each eigenvalue of (3.6) either is equal to its corresponding eigenvalue of (3.5) or belongs to $i[-2\pi, -\pi] \cup i[\pi, 2\pi]$, hence having no less absolute value than its corresponding eigenvalue of (3.5). Since both matrices (3.5) and (3.6) are skew-Hermitian, their singular values are the absolute values of their eigenvalues. Therefore, by the monotone property of the symmetric gauge function,

$$\begin{aligned} \left\| \begin{bmatrix} 0 & -2C \\ 2C^H & 0 \end{bmatrix} \right\| &\leq \left\| V \begin{bmatrix} 0 & -2A \\ 2A^H & 0 \end{bmatrix} V^H + U_1 \begin{bmatrix} 0 & -B \\ B^H & 0 \end{bmatrix} U_1^H \right. \\ &\quad \left. + U_2 \begin{bmatrix} 0 & B \\ -B^H & 0 \end{bmatrix} U_2^H \right\| \\ &\leq \left\| \begin{bmatrix} 0 & -2A \\ 2A^H & 0 \end{bmatrix} \right\| + 2 \left\| \begin{bmatrix} 0 & -B \\ B^H & 0 \end{bmatrix} \right\| \end{aligned}$$

for all unitarily invariant norms $\|\cdot\|$ on $\mathcal{M}_n$. Hence

$$\|C\| \leq \|A\| + \|B\|$$

for all unitarily invariant norms $\|\cdot\|$ on $\mathcal{M}_{m,n-m}$. This is equivalent to

$$\rho(\mathcal{X}, \mathcal{Z}) \leq \rho(\mathcal{X}, \mathcal{Y}) + \rho(\mathcal{Y}, \mathcal{Z})$$

for all symmetric gauge functions Φ on $\mathbb{R}^m$. $\square$

The metrics defined by the symmetric gauge functions of the canonical angles as in Theorem 3.1 will be called the angular metrics. We designate the notation ρ for such a metric.

By continuity, slight perturbations in subspaces will result in slight perturbations in the canonical angles in between. How are the perturbations in the canonical angles bounded by the perturbations in the subspaces involved? The same question was asked in [29], and the following ‘‘Mirsky-type result’’ gives an answer.

THEOREM 3.2. *Let ρ be the angular metric corresponding to symmetric gauge function Φ . Then for $\mathcal{X}, \mathcal{Y}, \mathcal{Z} \in \mathcal{G}_{m,n}$,*

$$\Phi(\theta(\mathcal{X}, \mathcal{Z}) - \theta(\mathcal{X}, \mathcal{Y})) \leq \rho(\mathcal{Y}, \mathcal{Z}) = \Phi(\theta(\mathcal{Y}, \mathcal{Z})).$$

Proof. Let $\sigma_1 \geq \dots \geq \sigma_n$ be singular values of

$$\Sigma := V \begin{bmatrix} 0 & -2A \\ 2A^H & 0 \end{bmatrix} V^H + U_1 \begin{bmatrix} 0 & -B \\ B^H & 0 \end{bmatrix} U_1^H + U_2 \begin{bmatrix} 0 & B \\ -B^H & 0 \end{bmatrix} U_2^H.$$

Then it follows from the argument in the proof of Theorem 3.1 that there are singular values $c_i, i = 1, 2, \dots, n$, ordered properly, of

$$\begin{bmatrix} 0 & -2C \\ 2C^H & 0 \end{bmatrix}$$

such that either $\sigma_i = c_i$ or $\sigma_i = 2\pi - c_i$. Rewrite $\alpha_i = \theta_i(\mathcal{X}, \mathcal{Y})$, $\beta_i = \theta_i(\mathcal{Y}, \mathcal{Z})$, and $\gamma_i = \theta_i(\mathcal{X}, \mathcal{Z})$ for $i = 1, 2, \dots, m$. Since c_i is a reordering of $2\gamma_1, 2\gamma_1, \dots, 2\gamma_m, 2\gamma_m, 0, \dots, 0$, we have for any symmetric gauge function $\Psi : \mathbb{R}^n \rightarrow \mathbb{R}$,

$$\begin{aligned} & \Psi(2\gamma_1 - 2\alpha_1, 2\gamma_1 - 2\alpha_1, \dots, 2\gamma_m - 2\alpha_m, 2\gamma_m - 2\alpha_m, 0, \dots, 0) \\ & \leq \Psi(c_1 - 2\alpha_1, c_2 - 2\alpha_1, \dots, c_{2m-1} - 2\alpha_m, c_{2m} - 2\alpha_m, c_{2m+1}, \dots, c_n) \\ & \leq \Psi(\sigma_1 - 2\alpha_1, \sigma_2 - 2\alpha_1, \dots, \sigma_{2m-1} - 2\alpha_m, \sigma_{2m} - 2\alpha_m, \sigma_{2m+1}, \dots, \sigma_n). \end{aligned}$$

Here we used the fact that $\sigma_i - \eta$ either is equal to $c_i - \eta$ or has no less absolute value than $c_i - \eta$ for any $\eta \in [-\pi, \pi]$. By (II), we obtain

$$\begin{aligned} & \Psi(2\gamma_1 - 2\alpha_1, 2\gamma_1 - 2\alpha_1, \dots, 2\gamma_m - 2\alpha_m, 2\gamma_m - 2\alpha_m, 0, \dots, 0) \\ & \leq \left\| \Sigma - V \begin{bmatrix} 0 & -2A \\ 2A^H & 0 \end{bmatrix} V^H \right\|_{\Psi} \\ & = \left\| U_1 \begin{bmatrix} 0 & -B \\ B^H & 0 \end{bmatrix} U_1^H + U_2 \begin{bmatrix} 0 & B \\ -B^H & 0 \end{bmatrix} U_2^H \right\|_{\Psi} \\ & \leq 2 \left\| \begin{bmatrix} 0 & -B \\ B^H & 0 \end{bmatrix} \right\|_{\Psi}. \end{aligned}$$

Hence

$$\Phi(\gamma_1 - \alpha_1, \dots, \gamma_m - \alpha_m) \leq \Phi(\beta_1, \dots, \beta_m)$$

for all symmetric gauge function Φ on $\mathbb{R}^m$, which immediately gives

$$\Phi(\theta(\mathcal{X}, \mathcal{Z}) - \theta(\mathcal{X}, \mathcal{Y})) \leq \rho(\mathcal{Y}, \mathcal{Z}). \quad \square$$

If we specialize the metrics in Theorem 3.1 to the ones corresponding to the Ky Fan k -functions, the triangle inequality gives us

$$(3.7) \quad \sum_{j=1}^k \theta_j(\mathcal{X}, \mathcal{Z}) \leq \sum_{j=1}^k \theta_j(\mathcal{X}, \mathcal{Y}) + \sum_{j=1}^k \theta_j(\mathcal{Y}, \mathcal{Z}).$$

On the other hand, specializing Theorem 3.2 to the Ky Fan functions gives us

$$(3.8) \quad \sum_{j=1}^k \theta_{i_j}(\mathcal{X}, \mathcal{Z}) \leq \sum_{j=1}^k \theta_{i_j}(\mathcal{X}, \mathcal{Y}) + \sum_{j=1}^k \theta_j(\mathcal{Y}, \mathcal{Z})$$

for any $1 \leq i_1 < \dots < i_k \leq m$. In this sense, the inequalities in Theorem 3.2 cover the triangle inequalities in Theorem 3.1 as special cases.

Sometimes we are concerned with the perturbation of canonical angles when both subspaces involved are perturbed. In this case, the following version of Theorem 3.2 might be more convenient.

COROLLARY 3.3. *Let ρ be the angular metric corresponding to symmetric gauge function Φ . Then for $\mathcal{X}, \mathcal{Y}, \tilde{\mathcal{X}}, \tilde{\mathcal{Y}} \in \mathcal{G}_{m,n}$,*

$$(3.9) \quad \Phi(\theta(\tilde{\mathcal{X}}, \tilde{\mathcal{Y}}) - \theta(\mathcal{X}, \mathcal{Y})) \leq \rho(\tilde{\mathcal{X}}, \mathcal{X}) + \rho(\tilde{\mathcal{Y}}, \mathcal{Y}).$$

Proof. For any subspaces $\mathcal{X}, \mathcal{Y}$ and the perturbations of them $\tilde{\mathcal{X}}, \tilde{\mathcal{Y}}$, it is easy to show that

$$\begin{aligned} \Phi(\theta(\tilde{\mathcal{X}}, \tilde{\mathcal{Y}}) - \theta(\mathcal{X}, \mathcal{Y})) &= \Phi(\theta(\tilde{\mathcal{X}}, \tilde{\mathcal{Y}}) - \theta(\mathcal{X}, \tilde{\mathcal{Y}}) + \theta(\mathcal{X}, \tilde{\mathcal{Y}}) - \theta(\mathcal{X}, \mathcal{Y})) \\ &\leq \Phi(\theta(\tilde{\mathcal{X}}, \tilde{\mathcal{Y}}) - \theta(\mathcal{X}, \tilde{\mathcal{Y}})) + \Phi(\theta(\mathcal{X}, \tilde{\mathcal{Y}}) - \theta(\mathcal{X}, \mathcal{Y})). \\ &\leq \rho(\tilde{\mathcal{X}}, \mathcal{X}) + \rho(\tilde{\mathcal{Y}}, \mathcal{Y}). \quad \square \end{aligned}$$

Sun in [29] established certain bounds on $\Phi(\theta(\tilde{\mathcal{X}}, \tilde{\mathcal{Y}}) - \theta(\mathcal{X}, \mathcal{Y}))$ for the case when Φ is the Ky Fan 1-function (Hölder ∞ -norm) and when it is the Hölder 2-norm. Corollary 3.3 extends and improves the bounds in [29].

However, the analysis in [29] is more general in one direction: it applies to canonical angles between two subspaces with possibly different dimensions, which are also important in applications. Let us first extend the notion of canonical angles to this more general setting. For $\mathcal{X} \in \mathcal{G}_{m,n}$ and $\mathcal{Y} \in \mathcal{G}_{l,n}$, again let $X_1 \in \mathcal{M}_{n,m}$ and $Y_1 \in \mathcal{M}_{n,l}$ be matrices whose columns form orthonormal bases of $\mathcal{X}$ and $\mathcal{Y}$, respectively. The canonical angles between $\mathcal{X}$ and $\mathcal{Y}$ are defined to be

$$\theta_i(\mathcal{X}, \mathcal{Y}) = \arccos \sigma_{\min\{m,l\}-i+1}(X_1^H Y_1), \quad i = 1, 2, \dots, \min\{m, l\}.$$

Let us take a closer look at this definition for the case when $m < l$. By the singular value decomposition, we know that there exist $E_1 \in \mathcal{U}_m$ and $F_1 \in \mathcal{U}_l$ such that

$$E_1^H X_1^H Y_1 F_1 = [\text{diag}(\cos \theta_1(\mathcal{X}, \mathcal{Y}), \dots, \cos \theta_m(\mathcal{X}, \mathcal{Y})) \quad 0_{m, l-m}].$$

This shows that $\mathcal{Y}$ has an m -dimensional subspace $\mathcal{Y}_0$ such that $\theta(\mathcal{X}, \mathcal{Y}_0) = \theta(\mathcal{X}, \mathcal{Y})$ and $\mathcal{X} \perp (\mathcal{Y} \ominus \mathcal{Y}_0)$. A similar interpretation can be made for the case when $l < m$. Now we would like to ask whether Corollary 3.3 can be extended to the case when $\mathcal{X}, \tilde{\mathcal{X}} \in \mathcal{G}_{m,n}$ and $\mathcal{Y}, \tilde{\mathcal{Y}} \in \mathcal{G}_{l,n}$ after reconciling the two metrics in the right-hand side of (3.9) properly. We are not able to give an affirmative answer at this moment, though all indications show that this is possible. We nevertheless are able to treat another important special case when $l = n - m$. It is this case that is directly relevant to robust control [24], [37]. In this case, there are at most $\min\{m, n - m\}$ nonzero canonical angles between any two members of $\mathcal{G}_{\max\{m, n-m\}, n}$. Hence a symmetric gauge function Φ on $\mathbb{R}^{\min\{m, n-m\}}$ also defines a metric on $\mathcal{G}_{\max\{m, n-m\}, n}$ by applying to the $\min\{m, n - m\}$ largest canonical angles. We will denote the metric on $\mathcal{G}_{m,n}$ and $\mathcal{G}_{n-m,n}$ defined by Φ using the same symbol ρ . We then have $\rho(\mathcal{X}, \mathcal{Y}) = \rho(\mathcal{X}_\perp, \mathcal{Y}_\perp)$ for $\mathcal{X}, \mathcal{Y} \in \mathcal{G}_{m,n}$.

COROLLARY 3.4. *Let ρ be the angular metric on $\mathcal{G}_{m,n}$ and that $\mathcal{G}_{n-m,n}$ corresponding to symmetric gauge function Φ on $\mathbb{R}^{\min\{m, n-m\}}$. Then for $\mathcal{X}, \tilde{\mathcal{X}} \in \mathcal{G}_{m,n}$ and $\mathcal{Y}, \tilde{\mathcal{Y}} \in \mathcal{G}_{n-m,n}$,*

$$\Phi(\theta(\tilde{\mathcal{X}}, \tilde{\mathcal{Y}}) - \theta(\mathcal{X}, \mathcal{Y})) \leq \rho(\tilde{\mathcal{X}}, \mathcal{X}) + \rho(\tilde{\mathcal{Y}}, \mathcal{Y}).$$

Proof. It follows immediately from the fact that

$$\theta_i(\tilde{\mathcal{X}}, \tilde{\mathcal{Y}}) = \pi/2 - \theta_{\min\{m, n-m\}-i+1}(\tilde{\mathcal{X}}, \tilde{\mathcal{Y}}_\perp)$$

and

$$\theta_i(\mathcal{X}, \mathcal{Y}) = \pi/2 - \theta_{\min\{m, n-m\}-i+1}(\mathcal{X}, \mathcal{Y}_\perp). \quad \square$$

In the rest of this section, we work towards a metric counterpart of the result in (III) on the robustness of the matrix rank. Following [16], we define the nullity and deficiency of subspaces $\mathcal{X}$ and $\mathcal{Y}$ to be

$$\text{nul}(\mathcal{X}, \mathcal{Y}) := \dim(\mathcal{X} \cap \mathcal{Y}) \quad \text{and} \quad \text{def}(\mathcal{X}, \mathcal{Y}) := \text{codim}(\mathcal{X} + \mathcal{Y}),$$

respectively. Clearly, $\text{def}(\mathcal{X}, \mathcal{Y}) = \text{nul}(\mathcal{X}_\perp, \mathcal{Y}_\perp)$ and in the case when $\dim \mathcal{X} + \dim \mathcal{Y} = n$, we have $\text{def}(\mathcal{X}, \mathcal{Y}) = \text{nul}(\mathcal{X}, \mathcal{Y})$. Also, $\mathcal{X} \oplus \mathcal{Y} = \mathbb{F}^n$ if and only if $\text{nul}(\mathcal{X}, \mathcal{Y}) = 0$ and $\text{def}(\mathcal{X}, \mathcal{Y}) = 0$.

The robustness of the nullity and deficiency is of great interest in mathematics [16], statistics [13], and control theory [9], [25], [33], [37]. In particular, if we have subspaces $\mathcal{X} \in \mathcal{G}_{m,n}$ and $\mathcal{Y} \in \mathcal{G}_{l,n}$ with $\text{nul}(\mathcal{X}, \mathcal{Y}) < k$ for some $1 \leq k \leq \min\{m, l\}$ and if we also know the perturbed versions $\tilde{\mathcal{X}} \in \mathcal{G}_{m,n}$ and $\tilde{\mathcal{Y}} \in \mathcal{G}_{l,n}$ satisfy $\rho_1(\mathcal{X}, \tilde{\mathcal{X}}) \leq \alpha$ and $\rho_2(\mathcal{Y}, \tilde{\mathcal{Y}}) \leq \beta$, we wish to obtain the tightest condition on α and β to ensure $\text{nul}(\tilde{\mathcal{X}}, \tilde{\mathcal{Y}}) < k$. The same problem can be considered for the deficiency. The following theorem solves these problems for the special case when $m = l$.

THEOREM 3.5. *Let ρ be the angular metric corresponding to symmetric gauge function Φ . Let $\mathcal{X}, \mathcal{Y} \in \mathcal{G}_{m,n}$. Then for $\alpha \geq 0$, $\beta \geq 0$, and $1 \leq k \leq m$,*

1. *$\text{nul}(\tilde{\mathcal{X}}, \tilde{\mathcal{Y}}) < k$ for all $\tilde{\mathcal{X}}$ and $\tilde{\mathcal{Y}}$ satisfying $\rho(\mathcal{X}, \tilde{\mathcal{X}}) \leq \alpha$ and $\rho(\mathcal{Y}, \tilde{\mathcal{Y}}) \leq \beta$ if and only if*

$$\alpha + \beta < \Phi[0, \dots, 0, \theta_{m-k+1}(\mathcal{X}, \mathcal{Y}), \dots, \theta_m(\mathcal{X}, \mathcal{Y})];$$

2. *$\text{def}(\tilde{\mathcal{X}}, \tilde{\mathcal{Y}}) < k$ for all $\tilde{\mathcal{X}}$ and $\tilde{\mathcal{Y}}$ satisfying $\rho(\mathcal{X}, \tilde{\mathcal{X}}) \leq \alpha$ and $\rho(\mathcal{Y}, \tilde{\mathcal{Y}}) \leq \beta$ if and only if*

$$\alpha + \beta < \Phi[0, \dots, 0, \theta_{n-m-k+1}(\mathcal{X}_\perp, \mathcal{Y}_\perp), \dots, \theta_{n-m}(\mathcal{X}_\perp, \mathcal{Y}_\perp)].$$

Proof. We only need to prove statement 1. Statement 2 follows from $\text{def}(\mathcal{X}, \mathcal{Y}) = \text{nul}(\mathcal{X}_\perp, \mathcal{Y}_\perp)$. Suppose $\text{nul}(\tilde{\mathcal{X}}, \tilde{\mathcal{Y}}) \geq k$. Then $\theta_j(\tilde{\mathcal{X}}, \tilde{\mathcal{Y}}) = 0, j = m - k + 1, \dots, m$. By Corollary 3.3,

$$\begin{aligned} \delta &:= \Phi(0, \dots, 0, \theta_{m-k+1}(\mathcal{X}, \mathcal{Y}), \dots, \theta_m(\mathcal{X}, \mathcal{Y})) \\ &\leq \Phi(\theta(\mathcal{X}, \mathcal{Y}) - \theta(\tilde{\mathcal{X}}, \tilde{\mathcal{Y}})) \leq \rho(\mathcal{X}, \tilde{\mathcal{X}}) + \rho(\mathcal{Y}, \tilde{\mathcal{Y}}) \leq \alpha + \beta. \end{aligned}$$

This shows that if

$$\alpha + \beta < \delta,$$

then $\text{nul}(\tilde{\mathcal{X}}, \tilde{\mathcal{Y}}) < k$.

Now assume that $\alpha + \beta \geq \delta$. Then there exist $\alpha_1 \in [0, \alpha]$ and $\beta_1 \in [0, \beta]$ such that $\alpha_1 + \beta_1 = \delta$. By the CS decomposition, there exist $X = [X_1 \ X_2]$ and $Y = [Y_1 \ Y_2]$, where the columns of $X_1 = [x_1 \cdots x_m]$ and $X_2 = [x_{m+1} \cdots x_n]$ form orthonormal bases of $\mathcal{X}$ and $\mathcal{X}_\perp$, respectively, and the columns of $Y_1 = [y_1 \cdots y_m]$ and $Y_2 = [y_{m+1} \cdots y_n]$ form orthonormal bases for $\mathcal{Y}$ and $\mathcal{Y}_\perp$, respectively, such that

$$(3.10) \quad Y = X \begin{bmatrix} \cos(\text{diag } \theta(\mathcal{X}, \mathcal{Y})) & -\sin(\text{diag } \theta(\mathcal{X}, \mathcal{Y})) & 0 \\ \sin(\text{diag } \theta(\mathcal{X}, \mathcal{Y})) & \cos(\text{diag } \theta(\mathcal{X}, \mathcal{Y})) & 0 \\ 0 & 0 & I \end{bmatrix}.$$

Let $\lambda = \alpha_1/\delta$; then $1 - \lambda = \beta_1/\delta$. Also abbreviate $\theta_j(\mathcal{X}, \mathcal{Y})$ by θ_j , $j = 1, 2, \dots, m$. Define

$$\tilde{x}_j = x_j \cos(\lambda\theta_j) + x_{m+j} \sin(\lambda\theta_j), \quad j = m - k + 1, \dots, m,$$

$$\tilde{\mathcal{X}} = \text{span}(x_1, \dots, x_{m-k}, \tilde{x}_{m-k+1}, \dots, \tilde{x}_m), \quad \tilde{\mathcal{Y}} = \text{span}(y_1, \dots, y_{m-k}, \tilde{y}_{m-k+1}, \dots, \tilde{y}_m).$$

Also define

$$\tilde{X}_1 = [x_1 \quad \cdots \quad x_{m-k} \quad \tilde{x}_{m-k+1} \quad \cdots \quad \tilde{x}_m], \quad \tilde{Y}_1 = [y_1 \quad \cdots \quad y_{m-k} \quad \tilde{y}_{m-k+1} \quad \cdots \quad \tilde{y}_m].$$

Then the columns of $\tilde{X}_1$ and $\tilde{Y}_1$ form orthonormal bases of $\tilde{\mathcal{X}}$ and $\tilde{\mathcal{Y}}$, respectively. Furthermore,

$$X_1^H \tilde{X}_1 = \text{diag}(1, \dots, 1, \cos(\lambda\theta_{m-k+1}), \dots, \cos(\lambda\theta_m)).$$

It follows that

$$\rho(\mathcal{X}, \tilde{\mathcal{X}}) = \Phi(0, \dots, 0, \lambda\theta_{m-k+1}, \dots, \lambda\theta_m) = \alpha_1 \leq \alpha.$$

From (3.10), we see that $y_j = x_j \cos \theta_j + x_{m+j} \sin \theta_j$, $j = m - k + 1, \dots, m$. Hence

$$Y_1^H \tilde{Y}_1 = \text{diag}(1, \dots, 1, \cos((1 - \lambda)\theta_{m-k+1}), \dots, \cos((1 - \lambda)\theta_m)).$$

Similar to the above, we have

$$\rho(\mathcal{Y}, \tilde{\mathcal{Y}}) = \Phi[0, \dots, 0, (1 - \lambda)\theta_{m-k+1}, \dots, (1 - \lambda)\theta_m] = \beta_1 \leq \beta.$$

Since

$$\tilde{Y}_1^H \tilde{X}_1 = \text{diag}(\cos \theta_1, \dots, \cos \theta_{m-k}, 1, \dots, 1),$$

it follows that

$$\theta(\tilde{\mathcal{X}}, \tilde{\mathcal{Y}}) = (\theta_1, \dots, \theta_{m-k}, 0, \dots, 0);$$

which implies that $\text{nul}(\tilde{\mathcal{X}}, \tilde{\mathcal{Y}}) \geq k$. This proves the necessity of the condition. $\square$

We believe that carefully formulated similar statements to those in Theorem 3.5 for the case when $m \neq l$ are also valid, though a proof is not available yet at this moment. Nevertheless, the important special case when $l = n - m$ can be treated as in the proof of Theorem 3.5 by following Corollary 3.4 instead. We state this as a corollary.

COROLLARY 3.6. *Let ρ be the angular metric on $\mathcal{G}_{m,n}$ and that on $\mathcal{G}_{n-m,n}$ corresponding to a symmetric gauge function Φ on $\mathbb{R}^{\min\{m, n-m\}}$. Let $\mathcal{X} \in \mathcal{G}_{m,n}$ and $\mathcal{Y} \in \mathcal{G}_{n-m,n}$. Then for $\alpha \geq 0$, $\beta \geq 0$ and $1 \leq k \leq \min\{m, n - m\}$,*

1. $\text{nul}(\tilde{\mathcal{X}}, \tilde{\mathcal{Y}}) < k$ for all $\tilde{\mathcal{X}} \in \mathcal{G}_{m,n}$ and $\tilde{\mathcal{Y}} \in \mathcal{G}_{n-m,n}$ satisfying $\rho(\mathcal{X}, \tilde{\mathcal{X}}) \leq \alpha$ and $\rho(\mathcal{Y}, \tilde{\mathcal{Y}}) \leq \beta$ if and only if

$$\alpha + \beta < \Phi[0, \dots, 0, \theta_{m-k+1}(\mathcal{X}, \mathcal{Y}), \dots, \theta_{\min\{m, n-m\}}(\mathcal{X}, \mathcal{Y})];$$

2. $\text{def}(\tilde{\mathcal{X}}, \tilde{\mathcal{Y}}) < k$ for all $\tilde{\mathcal{X}} \in \mathcal{G}_{m,n}$ and $\tilde{\mathcal{Y}} \in \mathcal{G}_{n-m,n}$ satisfying $\rho(\mathcal{X}, \tilde{\mathcal{X}}) \leq \alpha$ and $\rho(\mathcal{Y}, \tilde{\mathcal{Y}}) \leq \beta$ if and only if

$$\alpha + \beta < \Phi[0, \dots, 0, \theta_{m-k+1}(\mathcal{X}_\perp, \mathcal{Y}_\perp), \dots, \theta_{\min\{m, n-m\}}(\mathcal{X}_\perp, \mathcal{Y}_\perp)].$$

4. When the triangle inequalities become equalities. Whenever we have an inequality, the analysis of the cases when the inequality becomes an equality is of special importance. Such analysis often reveals extra insight and structures possessed by the inequality. It also exhibits how sharp the inequality is. In applications, these extreme cases often represent either the worst or the best cases where special attention is needed. In the following, the equality cases for the triangle inequalities for the angular metrics and the gap metrics will be analyzed and compared. They will be characterized in terms of the direct rotations between the subspaces involved, which can be viewed as a natural generalization of the equality case relating three one-dimensional subspaces in $\mathbb{R}^n$. The proofs in this section turn out to be quite technical. We suggest that readers skip the proofs in their first reading.

In this section, we first characterize those triples of subspaces $\mathcal{X}, \mathcal{Y}, \mathcal{Z} \in \mathcal{G}_{m,n}$ such that

$$(4.1) \quad \rho(\mathcal{X}, \mathcal{Z}) = \rho(\mathcal{X}, \mathcal{Y}) + \rho(\mathcal{Y}, \mathcal{Z}),$$

i.e., the triangle inequality becomes an equality, either for a particular angular metric or for all angular metrics. An angular metric $\rho = \Phi \circ \theta$ is said to be strictly convex if Φ is a strictly convex symmetric gauge function, i.e., $\Phi(\xi + \zeta) = \Phi(\xi) + \Phi(\zeta)$ only if ξ and ζ are linearly dependent. The following theorem shows that if (4.1) is satisfied for a strictly convex angular metric, then $\mathcal{Y}$ has to be an intermediate position on the path when $\mathcal{X}$ is rotated to $\mathcal{Z}$ by a direct rotation, which in turn implies that (4.1) is satisfied for all angular metrics.

THEOREM 4.1. *For $\mathcal{X}, \mathcal{Y}, \mathcal{Z} \in \mathcal{G}_{m,n}$, the following statements are equivalent:*

1. *Equation (4.1) holds for all angular metrics.*
2. *Equation (4.1) holds for a strictly convex angular metric.*
3. *There exists a direct rotation from $\mathcal{X}$ to $\mathcal{Z}$,*

$$X \exp \begin{bmatrix} 0 & -C \\ C^H & 0 \end{bmatrix} X^H,$$

where $C = [\text{diag } \theta(\mathcal{X}, \mathcal{Z}) \quad 0_{m,n-2m}]$, and $\lambda \in [0, 1]$ such that

$$\mathcal{Y} = X \exp \begin{bmatrix} 0 & -\lambda C \\ \lambda C^H & 0 \end{bmatrix} X^H \mathcal{X}.$$

Proof. It is obvious that statement 1 implies statement 2. We only need to prove that statement 2 implies statement 3 and that 3 implies 1. We do the latter first. If

$$\mathcal{Y} = X \exp \begin{bmatrix} 0 & -\lambda C \\ \lambda C^H & 0 \end{bmatrix} X^H \mathcal{X},$$

then a direct rotation from $\mathcal{X}$ to $\mathcal{Y}$ is

$$R = X \exp \begin{bmatrix} 0 & -\lambda C \\ \lambda C^H & 0 \end{bmatrix} X^H$$

and one from $\mathcal{Y}$ to $\mathcal{Z}$ is

$$\begin{aligned} S &= \left(X \exp \begin{bmatrix} 0 & -\lambda C \\ \lambda C^H & 0 \end{bmatrix} \right) \exp \begin{bmatrix} 0 & -(1-\lambda)C \\ (1-\lambda)C^H & 0 \end{bmatrix} \\ &\quad \times \left(X \exp \begin{bmatrix} 0 & -\lambda C \\ \lambda C^H & 0 \end{bmatrix} \right)^H \\ &= X \exp \begin{bmatrix} 0 & -C \\ C^H & 0 \end{bmatrix} X^H X \exp \begin{bmatrix} 0 & \lambda C \\ -\lambda C^H & 0 \end{bmatrix} X^H. \end{aligned}$$

Then $\theta(\mathcal{X}, \mathcal{Y}) = \lambda\theta(\mathcal{X}, \mathcal{Z})$ and $\theta(\mathcal{Y}, \mathcal{Z}) = (1 - \lambda)\theta(\mathcal{X}, \mathcal{Z})$. This immediately implies (4.1) for all angular metrics.

To prove that 2 implies 3, let Φ be the strictly convex symmetric gauge function corresponding to the angular metric satisfying (4.1), and notice that equality (4.1) and the fact that $\theta(\mathcal{X}, \mathcal{Z})$ is weakly majorized by $\theta(\mathcal{X}, \mathcal{Y}) + \theta(\mathcal{Y}, \mathcal{Z})$ imply

$$\begin{aligned}\Phi(\theta(\mathcal{X}, \mathcal{Y})) + \Phi(\theta(\mathcal{Y}, \mathcal{Z})) &= \Phi(\theta(\mathcal{X}, \mathcal{Z})) \leq \Phi(\theta(\mathcal{X}, \mathcal{Y}) + \theta(\mathcal{Y}, \mathcal{Z})) \\ &\leq \Phi(\theta(\mathcal{X}, \mathcal{Y})) + \Phi(\theta(\mathcal{Y}, \mathcal{Z})).\end{aligned}$$

This forces

$$\Phi(\theta(\mathcal{X}, \mathcal{Z})) = \Phi(\theta(\mathcal{X}, \mathcal{Y}) + \theta(\mathcal{Y}, \mathcal{Z})) = \Phi(\theta(\mathcal{X}, \mathcal{Y})) + \Phi(\theta(\mathcal{Y}, \mathcal{Z})).$$

Since Φ is strictly convex, it follows from the second equality above that $\theta(\mathcal{X}, \mathcal{Y})$ and $\theta(\mathcal{Y}, \mathcal{Z})$ are linearly dependent. Also since $\theta(\mathcal{X}, \mathcal{Z})$ is weakly majorized by $\theta(\mathcal{X}, \mathcal{Y}) + \theta(\mathcal{Y}, \mathcal{Z})$, it follows from [30] that $\theta(\mathcal{X}, \mathcal{Z})$ is in the convex hull of

$$\{P(\theta(\mathcal{X}, \mathcal{Y}) + \theta(\mathcal{Y}, \mathcal{Z})) : P \text{ is a signed permutation matrix}\}.$$

Again since Φ is strictly convex, $\Phi(\theta(\mathcal{X}, \mathcal{Z})) = \Phi(\theta(\mathcal{X}, \mathcal{Y}) + \theta(\mathcal{Y}, \mathcal{Z}))$ only if $\theta(\mathcal{X}, \mathcal{Z})$ is one of the vertices of the convex hull, i.e., $\theta(\mathcal{X}, \mathcal{Z}) = P(\theta(\mathcal{X}, \mathcal{Y}) + \theta(\mathcal{Y}, \mathcal{Z}))$ for a particular signed permutation matrix P . Since both of them are nonnegative and are decreasingly ordered, we have $\theta(\mathcal{X}, \mathcal{Z}) = \theta(\mathcal{X}, \mathcal{Y}) + \theta(\mathcal{Y}, \mathcal{Z})$. In summary, we conclude that there exists $0 \leq \lambda \leq 1$ such that $\theta(\mathcal{X}, \mathcal{Y}) = \lambda\theta(\mathcal{X}, \mathcal{Z})$ and $\theta(\mathcal{Y}, \mathcal{Z}) = (1 - \lambda)\theta(\mathcal{X}, \mathcal{Z})$.

In the following, we consider only the nontrivial case when $0 < \lambda < 1$. By the discussion in section 2, we know that there exists a direct rotation from $\mathcal{X}$ to $\mathcal{Y}$ given by

$$R = \hat{X} \exp \begin{bmatrix} 0 & -\lambda C \\ \lambda C^H & 0 \end{bmatrix} \hat{X}^H$$

and a direct rotation from $\mathcal{X}$ to $\mathcal{Z}$ given by

$$T = \tilde{X} \exp \begin{bmatrix} 0 & -C \\ C^H & 0 \end{bmatrix} \tilde{X}^H,$$

where $\hat{X} = [\hat{X}_1 \ \hat{X}_2] \in \mathcal{U}_n$, $\tilde{X} = [\tilde{X}_1 \ \tilde{X}_2] \in \mathcal{U}_n$, and columns of $\hat{X}_1$ and those of $\tilde{X}_1$ both form (possibly different) orthonormal bases of $\mathcal{X}$. This shows that the columns of the matrices

$$X_1 = \tilde{X}_1, \quad Y_1 = \tilde{X} N \begin{bmatrix} \cos(\text{diag } \lambda\theta(\mathcal{X}, \mathcal{Z})) \\ \sin(\text{diag } \lambda\theta(\mathcal{X}, \mathcal{Z})) \\ 0_{n-2m, m} \end{bmatrix}, \quad Z_1 = \tilde{X} \begin{bmatrix} \cos(\text{diag } \theta(\mathcal{X}, \mathcal{Z})) \\ \sin(\text{diag } \theta(\mathcal{X}, \mathcal{Z})) \\ 0_{n-2m, m} \end{bmatrix}$$

form orthonormal bases of $\mathcal{X}$, $\mathcal{Y}$, and $\mathcal{Z}$, respectively, where $N = \tilde{X}^H \hat{X} = \begin{bmatrix} N_1 & 0 \\ 0 & N_2 \end{bmatrix}$, $N_1 \in \mathcal{U}_m$, $N_2 \in \mathcal{U}_{n-m}$.

By the definition of canonical angles, we know that there exist $E, F \in \mathcal{U}_m$ such that

$$\begin{aligned}(4.2) \quad & \cos(\text{diag } (1 - \lambda)\theta(\mathcal{X}, \mathcal{Z})) \\ &= E Z_1^H Y_1 F^H \\ &= E \cos(\text{diag } \theta(\mathcal{X}, \mathcal{Z})) N_1 \cos(\text{diag } \lambda\theta(\mathcal{X}, \mathcal{Z})) F^H \\ &\quad + E [\sin(\text{diag } \theta(\mathcal{X}, \mathcal{Z})) \ 0_{m, n-2m}] N_2 \begin{bmatrix} \sin(\text{diag } \lambda\theta(\mathcal{X}, \mathcal{Z})) \\ 0_{n-2m, m} \end{bmatrix} F^H.\end{aligned}$$

Let

$$\text{diag } \theta(\mathcal{X}, \mathcal{Z}) = \text{diag}(\phi_1 I_{m_1}, \phi_2 I_{m_2}, \dots, \phi_l I_{m_l}),$$

where $\pi/2 = \phi_1 > \phi_2 > \dots > \phi_l \geq 0$ and $m_1 + \dots + m_l = m$. By [27, p. 76], we have

$$(4.3) \quad \begin{aligned} & \text{Re tr} (E \cos(\text{diag } \theta(\mathcal{X}, \mathcal{Z})) N_1 \cos(\text{diag } \lambda \theta(\mathcal{X}, \mathcal{Z})) F^H) \\ & \leq \sum_{j=1}^l m_j \cos \phi_j \cos(\lambda \phi_j), \end{aligned}$$

where $\text{Re tr}(\cdot)$ denotes the real part of the trace. Similarly, we have

$$(4.4) \quad \begin{aligned} & \text{Re tr} \left[E \begin{bmatrix} \sin(\text{diag } \theta(\mathcal{X}, \mathcal{Z})) 0_{m, n-2m} \end{bmatrix} N_2 \begin{bmatrix} \sin(\text{diag } \lambda \theta(\mathcal{X}, \mathcal{Z})) \\ 0_{n-2m, m} \end{bmatrix} F^H \right] \\ & \leq \sum_{j=1}^l m_j \sin \phi_j \sin(\lambda \phi_j). \end{aligned}$$

From (4.2), (4.3), (4.4), and the fact that

$$\text{Re tr} (\cos(\text{diag} (1 - \lambda) \theta(\mathcal{X}, \mathcal{Z}))) = \sum_{j=1}^l m_j \cos \phi_j \cos(\lambda \phi_j) + \sum_{j=1}^l m_j \sin \phi_j \sin(\lambda \phi_j),$$

it follows that

$$(4.5) \quad \begin{aligned} & \text{Re tr} (\cos(\text{diag } \theta(\mathcal{X}, \mathcal{Z})) N_1 \cos(\text{diag } \lambda \theta(\mathcal{X}, \mathcal{Z})) F^H E) \\ & = \sum_{j=1}^l m_j \cos \phi_j \cos(\lambda \phi_j) \end{aligned}$$

and

$$\begin{aligned} & \text{Re tr} \left[\begin{bmatrix} \sin(\text{diag } \theta(\mathcal{X}, \mathcal{Z})) & 0_{m, n-2m} \end{bmatrix} N_2 \begin{bmatrix} \sin(\text{diag } \lambda \theta(\mathcal{X}, \mathcal{Z})) \\ 0_{n-2m, m} \end{bmatrix} F^H E \right] \\ & = \sum_{j=1}^l m_j \sin \phi_j \sin(\lambda \phi_j). \end{aligned}$$

Suppose $D = N_1 \cos(\text{diag } \lambda \theta(\mathcal{X}, \mathcal{Z})) F^H E$. It follows from the same lines as in the proof of [27, p. 76] that $D = \text{diag}(D_1, \dots, D_l)$, where $D_j \in \mathcal{M}_{m_j}$. Then we have

$$(4.6) \quad \begin{aligned} \text{Re tr} (\cos(\text{diag} (\theta(\mathcal{X}, \mathcal{Z}) D))) &= \text{Re tr} (\text{diag} (\cos \phi_1 D_1, \dots, \cos \phi_l D_l)) \\ &= \text{Re tr} (\cos \phi_2 D_2) + \dots + \text{Re tr} (\cos \phi_l D_l) \\ &\leq \sum_{i=1}^{m_2} \cos \phi_2 \sigma_i(D_2) + \dots + \sum_{i=1}^{m_l} \cos \phi_l \sigma_i(D_l) \\ &\leq \sum_{j=2}^l m_j \cos \phi_j \cos(\lambda \phi_j), \end{aligned}$$

where the second last inequality holds since

$$\operatorname{Re tr}(\cos \phi_j D_j) \leq \sum_{i=1}^{m_j} \cos \phi_j \sigma_i(D_j)$$

for each $j = 2, \dots, l$ and the last inequality follows from the fact that the singular values of all D_j , $j = 1, \dots, l$, all together are the singular values of $\cos(\operatorname{diag} \lambda \theta(\mathcal{X}, \mathcal{Z}))$.

Comparing (4.5) and (4.6), we see that inequalities in (4.6) have to be equalities. This is possible only if

$$\operatorname{Re tr}(\cos(\phi_j) D_j) = \sum_{i=1}^{m_j} \cos(\phi_j) \sigma_i(D_j) = m_j \cos \phi_j \cos(\lambda \phi_j)$$

for $j = 2, \dots, l$. So there must exist $P_j \in \mathcal{U}_{m_j}$ such that $D_j = \cos(\lambda \phi_j) P_j$ with $\operatorname{Re tr} P_j = m_j$. Hence $D_j = \cos(\lambda \phi_j) I_{m_j}$, $j = 2, \dots, l$. After determining $D_2, \dots, D_l$, the singular values D_1 have no choice but to equal $\cos(\lambda \phi_1)$, i.e., $D_1 = \cos(\lambda \phi_1) W$ for some $W \in \mathcal{U}_{m_1}$. Hence we have proved that

$$\begin{aligned} N_1 \cos(\operatorname{diag} \lambda \theta(\mathcal{X}, \mathcal{Z})) F^H E &= \operatorname{diag}(\cos(\lambda \phi_1) W, \cos(\lambda \phi_2) I_{m_2}, \dots, \cos(\lambda \phi_l) I_{m_l}) \\ &= \begin{bmatrix} W & 0 \\ 0 & I_{m-m_1} \end{bmatrix} \cos(\operatorname{diag} \lambda \theta(\mathcal{X}, \mathcal{Z})). \end{aligned}$$

Similarly, we can prove that

$$N_2 \begin{bmatrix} \sin(\operatorname{diag} \lambda \theta(\mathcal{X}, \mathcal{Z})) \\ 0 \end{bmatrix} F^H E = \begin{bmatrix} \sin(\operatorname{diag} \lambda \theta(\mathcal{X}, \mathcal{Z})) \\ 0_{n-2m, m} \end{bmatrix}.$$

Define

$$X = \tilde{X} \begin{bmatrix} W & 0 \\ 0 & I_{n-m_1} \end{bmatrix}.$$

Then

$$X \exp \begin{bmatrix} 0 & -C \\ C^H & 0 \end{bmatrix} X^H$$

is also a direct rotation from $\mathcal{X}$ to $\mathcal{Z}$. We can see that

$$Y_1 F^H E = \tilde{X} \begin{bmatrix} N_1 & 0 \\ 0 & N_2 \end{bmatrix} \begin{bmatrix} \cos(\operatorname{diag} \lambda \theta(\mathcal{X}, \mathcal{Z})) \\ \sin(\operatorname{diag} \lambda \theta(\mathcal{X}, \mathcal{Z})) \\ 0 \end{bmatrix} F^H E = X \begin{bmatrix} \cos(\operatorname{diag} \lambda \theta(\mathcal{X}, \mathcal{Z})) \\ \sin(\operatorname{diag} \lambda \theta(\mathcal{X}, \mathcal{Z})) \\ 0 \end{bmatrix}.$$

This shows

$$\mathcal{Y} = X \exp \begin{bmatrix} 0 & -\lambda C \\ \lambda C^H & 0 \end{bmatrix} X^H \mathcal{X}. \quad \square$$

The following example shows a case when direct rotations from $\mathcal{X}$ to $\mathcal{Y}$ are not unique and hence the paths from $\mathcal{X}$ to $\mathcal{Y}$ they define are not unique.

Example 1. Let $\mathcal{X}, \mathcal{Y}, \mathcal{Z} \in \mathcal{G}_{m, 2m}$ be spanned by the columns of matrices

$$\begin{bmatrix} I \\ 0 \end{bmatrix}, \quad \begin{bmatrix} (\cos \alpha) U \\ (\sin \alpha) V \end{bmatrix}, \quad \begin{bmatrix} 0 \\ I \end{bmatrix},$$

respectively. Here $\alpha \in [0, \pi/2]$ and $U, V \in \mathcal{U}_m$. Then $\theta(\mathcal{X}, \mathcal{Y}) = (\alpha, \dots, \alpha)$, $\theta(\mathcal{Y}, \mathcal{Z}) = (\pi/2 - \alpha, \dots, \pi/2 - \alpha)$, and $\theta(\mathcal{X}, \mathcal{Z}) = (\pi/2, \dots, \pi/2)$. The triangle equality holds for all angular metrics $\rho = \Phi \circ \theta$ for all α, U, V .

Notice that for the gap metric,

$$\sin \theta_1(\mathcal{X}, \mathcal{Z}) = \sin \theta_1(\mathcal{X}, \mathcal{Y}) + \sin \theta_1(\mathcal{Y}, \mathcal{Z})$$

if and only if $\mathcal{Y} = \mathcal{X}$ or $\mathcal{Y} = \mathcal{Z}$. Also we can show that

$$\Phi(\sin \theta(\mathcal{X}, \mathcal{Z})) = \Phi(\sin \theta(\mathcal{X}, \mathcal{Y})) + \Phi(\sin \theta(\mathcal{Y}, \mathcal{Z}))$$

for a strictly convex symmetric gauge function Φ if and only if $\mathcal{Y} = \mathcal{X}$ or $\mathcal{Y} = \mathcal{Z}$. Hence the triangle equality holds for a gap metric $\Phi \circ \sin \theta$, where Φ is either Φ_1 or any strictly convex symmetric gauge function, only in the trivial cases. The same thing can be said for the Hausdorff metrics.

The Ky Fan k -functions are not strictly convex, so one can expect that $\mathcal{X}, \mathcal{Y}, \mathcal{Z} \in \mathcal{G}_{m,n}$ may take more general positions when (4.1) is satisfied only for a Ky Fan function. This is indeed the case. It is easy to prove that

$$\Phi_m(\theta(\mathcal{X}, \mathcal{Z})) = \Phi_m(\theta(\mathcal{X}, \mathcal{Y})) + \Phi_m(\theta(\mathcal{Y}, \mathcal{Z}))$$

if there exists a direct rotation from $\mathcal{X}$ to $\mathcal{Z}$

$$X \exp \begin{bmatrix} 0 & -C \\ C^H & 0 \end{bmatrix} X^H,$$

where $C = [\text{diag } \theta(\mathcal{X}, \mathcal{Z}) \quad 0_{m,n-2m}]$, and $\Lambda = \text{diag } (\lambda_1, \dots, \lambda_m)$, where $\lambda_j \in [0, 1]$ such that

$$\mathcal{Y} = X \exp \begin{bmatrix} 0 & -\Lambda C \\ C^H \Lambda & 0 \end{bmatrix} X^H \mathcal{X}.$$

It would be nice to determine whether the condition is also necessary.

In contrast, for the gap metrics corresponding to the Ky Fan k -functions, we have the following results.

PROPOSITION 4.2. *Let $\mathcal{X}, \mathcal{Y}, \mathcal{Z} \in \mathcal{G}_{m,n}$. The equality*

$$(4.7) \quad \Phi_k(\sin \theta(\mathcal{X}, \mathcal{Z})) = \Phi_k(\sin \theta(\mathcal{X}, \mathcal{Y})) + \Phi_k(\sin \theta(\mathcal{Y}, \mathcal{Z}))$$

holds for a particular k with $1 \leq k \leq m$ if and only if the collection

$$\{\theta_1(\mathcal{X}, \mathcal{Y}), \dots, \theta_k(\mathcal{X}, \mathcal{Y}), \theta_1(\mathcal{Y}, \mathcal{Z}), \dots, \theta_k(\mathcal{Y}, \mathcal{Z})\}$$

contains $\theta_1(\mathcal{X}, \mathcal{Z}), \dots, \theta_k(\mathcal{X}, \mathcal{Z})$ together with additional k zeros.

Proof. The sufficiency is clear. To prove the necessity, denote $\alpha_j = \theta_j(\mathcal{X}, \mathcal{Y})$, $\beta_j = \theta_j(\mathcal{Y}, \mathcal{Z})$, and $\gamma_j = \theta_j(\mathcal{X}, \mathcal{Z})$, $j = 1, \dots, m$. Equality (4.7) becomes

$$(4.8) \quad \sum_{j=1}^k \sin \gamma_j = \sum_{j=1}^k (\sin \alpha_j + \sin \beta_j).$$

Let the columns of X_1, Y_1, Z_1 form orthonormal bases for subspaces $\mathcal{X}, \mathcal{Y}, \mathcal{Z}$, respectively. Denote

$$\begin{aligned} L &= X_1 X_1^H - Y_1 Y_1^H, \\ M &= Y_1 Y_1^H - Z_1 Z_1^H, \\ N &= X_1 X_1^H - Z_1 Z_1^H. \end{aligned}$$

Then the eigenvalues of L are $\sin \alpha_1, \dots, \sin \alpha_m, -\sin \alpha_m, \dots, -\sin \alpha_1, 0, \dots, 0$. Similar statements hold for M and N . Let $U \in \mathcal{U}_n$ satisfy $UNU^H = \text{diag}(N_1, N_2)$, where $N_1 \in \mathcal{M}_k$ has eigenvalues $\sin \gamma_1, \dots, \sin \gamma_k$. Let L_1 and M_1 be the leading $k \times k$ submatrices of ULU^H and UMU^H , respectively. Since

$$ULU^H + UMU^H = UNU^H,$$

it follows that

$$\text{tr } L_1 + \text{tr } M_1 = \text{tr } N_1.$$

It is well known that eigenvalues of a Hermitian matrix majorize its diagonal elements; see, for example, [20]. Then

$$(4.9) \quad \text{tr } L_1 \leq \sum_{j=1}^k \sin \alpha_j, \quad \text{tr } M_1 \leq \sum_{j=1}^k \sin \beta_j.$$

Equality (4.8) forces the equality signs in (4.9) to hold. By the extremal property in [19, Lemma 4.1], we have $ULU^H = \text{diag}(L_1, L_2)$ such that L_1 has eigenvalues $\sin \alpha_1, \dots, \sin \alpha_k$ and $UMU^H = \text{diag}(M_1, M_2)$ such that M_1 has eigenvalues $\sin \beta_1, \dots, \sin \beta_k$. Consequently, the unitary Hermitian matrices

$$2UX_1X_1^H U^H - I, \quad 2UY_1Y_1^H U^H - I, \quad 2UZ_1Z_1^H U^H - I,$$

whose differences are block diagonal, must share the same off-diagonal block, i.e., they take the respective forms

$$\begin{bmatrix} X_{11} & D \\ D^H & X_{22} \end{bmatrix}, \quad \begin{bmatrix} Y_{11} & D \\ D^H & Y_{22} \end{bmatrix}, \quad \begin{bmatrix} Z_{11} & D \\ D^H & Z_{22} \end{bmatrix},$$

where $D \in \mathcal{M}_{k, n-k}$. If we replace U by $\text{diag}(V_1, V_2)U$ and properly choose $V_1 \in \mathcal{U}_k$ and $V_2 \in \mathcal{U}_{n-k}$, we can diagonalize D by a singular value decomposition. Hence we can assume without loss of generality that $D = [\text{diag}(d_1 I_{r_1}, \dots, d_l I_{r_l}) \quad 0_{k, n-2k}]$, where $0 = d_1 < \dots < d_l \leq 1$.

Since

$$X_{11}X_{11}^H + DD^H = X_{11}^2 + DD^H = I_k,$$

it follows that there exist certain unitary Hermitian matrices P_j and $0 \leq g_l < \dots < g_1 = 1$ satisfying $g_j^2 + d_j^2 = 1$, $j = 1, \dots, l$, such that $X_{11} = \text{diag}(g_1 P_1, \dots, g_l P_l)$. Similarly, $Y_{11} = \text{diag}(g_1 Q_1, \dots, g_l Q_l)$ and $Z_{11} = \text{diag}(g_1 R_1, \dots, g_l R_l)$, where Q_j, R_j are unitary Hermitian matrices.

Consider matrices P_1, Q_1, R_1 . Let $W_1 \in \mathcal{U}_{r_1}$ be chosen such that $W_1 Q_1 W_1^H = \text{diag}(I_{s_1}, -I_{t_1}) := J_{s_1, t_1}$. Since $W_1 P_1 W_1^H - W_1 Q_1 W_1^H \geq 0$, it follows that $W_1 P_1 W_1^H = \text{diag}(I_{s_1}, F_1)$ with $F_1 \in \mathcal{U}_{t_1}$. Similarly, since $W_1 Q_1 W_1^H - W_1 R_1 W_1^H \geq 0$, we have $W_1 R_1 W_1^H = \text{diag}(G_1, -I_{t_1})$ with $G_1 \in \mathcal{U}_{s_1}$. Thus,

$$W_1(P_1 - Q_1)W_1^H = \text{diag}(0_{s_1}, F_1 + I_{t_1}), \quad W_1(Q_1 - R_1)W_1^H = \text{diag}(I_{s_1} - G_1, 0_{t_1}).$$

Applying the same arguments to P_j, Q_j, R_j , we have $W_j \in \mathcal{U}_{r_j}$ such that $W_j Q_j W_j^H = J_{s_j, t_j}$ and

$$W_j(P_j - Q_j)W_j^H = \text{diag}(0_{s_j}, F_j + I_{t_j}), \quad W_j(Q_j - R_j)W_j^H = \text{diag}(I_{s_j} - G_j, 0_{t_j})$$

for $j = 2, \dots, l$. Consequently, letting $W = \text{diag}(W_1, \dots, W_l)$ yields

$$\begin{aligned} WL_1W^H &= \frac{1}{2} \text{diag}(0_{s_1}, F_1 + I_{t_1}, \dots, 0_{s_l}, F_l + I_{t_l}), \\ WM_1W^H &= \frac{1}{2} \text{diag}(I_{s_1} - G_1, 0_{t_1}, \dots, I_{s_l} - G_l, 0_{t_l}), \end{aligned}$$

where $s = s_1 + \dots + s_l$, $t = t_1 + \dots + t_l$, and $t + s = k$. This gives

$$WN_1W^H = WL_1W^H + WM_1W^H = \frac{1}{2} \text{diag}(I_{s_1} - G_1, F_1 + I_{t_1}, \dots, I_{s_l} - G_l, F_l + I_{t_l}).$$

Now it is easy to see that L_1 has at most t nonzero eigenvalues $\sin \alpha_1, \dots, \sin \alpha_t$ and M_1 has at most s nonzero eigenvalues $\sin \beta_1, \dots, \sin \beta_s$, and they constitute the largest k eigenvalues of N_1 , which are $\sin \gamma_1, \dots, \sin \gamma_k$. This completes the proof. $\square$

If equality (4.7) holds for $k = 1$, then either $\mathcal{Y} = \mathcal{X}$ or $\mathcal{Y} = \mathcal{Z}$. This implies that the equality (4.7) holds for all $1 \leq k \leq m$. If equality (4.7) holds for any other k but not for $k = 1$, then both $\theta_k(\mathcal{X}, \mathcal{Y}) = \theta_k(\mathcal{Y}, \mathcal{Z}) = 0$. In this case, $\theta_j(\mathcal{X}, \mathcal{Y}) = \theta_j(\mathcal{Y}, \mathcal{Z}) = 0$ for all $j \geq k$. This implies that $\theta_j(\mathcal{X}, \mathcal{Z}) = 0$ for all $j > k$ since otherwise we would have $\sum_{j=1}^{k+1} \theta_j(\mathcal{X}, \mathcal{Z}) > \sum_{j=1}^{k+1} \theta_j(\mathcal{X}, \mathcal{Y}) + \sum_{j=1}^{k+1} \theta_j(\mathcal{Y}, \mathcal{Z})$. This shows that (4.7) holds when k is replaced by $k+1, k+2, \dots, m$. Because of this, we can define the smallest k with (4.7) satisfied as $k_{\min}$. Then $k_{\min} = 1$ if and only if $\mathcal{Y} = \mathcal{X}$ or $\mathcal{Y} = \mathcal{Z}$. When $k_{\min} > 1$, we know that $\theta_{k_{\min}}(\mathcal{X}, \mathcal{Z}) > 0$ but $\theta_{k_{\min}+1}(\mathcal{X}, \mathcal{Z}) = 0$. This means that $\dim \mathcal{X} \cap \mathcal{Z} = m - k_{\min}$.

THEOREM 4.3. *The equality (4.7) holds for some k with $1 \leq k \leq m$ if and only if there exists a direct rotation from $\mathcal{X}$ to $\mathcal{Z}$,*

$$X \exp \begin{bmatrix} 0 & -C \\ C^H & 0 \end{bmatrix} X^H,$$

where $C = [\text{diag}(\theta(\mathcal{X}, \mathcal{Z}) \quad 0_{m, n-2m})]$, and $\Lambda = \text{diag}(\lambda_1, \dots, \lambda_m)$, where $\lambda_j \in \{0, 1\}$ such that

$$\mathcal{Y} = X \exp \begin{bmatrix} 0 & -\Lambda C \\ C^H \Lambda & 0 \end{bmatrix} X^H \mathcal{X}.$$

Proof. The sufficiency is clear. To simplify the proof of the necessity, we will prove the following fact first: if $k_{\min} > 1$, then $\mathcal{X} \cap \mathcal{Z} \subset \mathcal{Y} \subset \mathcal{X} + \mathcal{Z}$. Adopting the same notation as in the proof of Proposition 4.2, we choose unitary matrix $U^H = [U_1^H \quad U_2^H \quad U_3^H]$ such that the columns of U_2^H span $\mathcal{X} \cap \mathcal{Z}$ and those of U_3^H span $(\mathcal{X} + \mathcal{Z})_{\perp}$. Then

$$UX_1X_1^H U^H = \text{diag}(\tilde{X}, I, 0), \quad UZ_1Z_1^H U^H = \text{diag}(\tilde{Z}, I, 0),$$

which gives

$$UNU^H = \text{diag}(\tilde{X} - \tilde{Z}, 0, 0),$$

where $\tilde{X} - \tilde{Z}$ has eigenvalues $\sin \gamma_1, \dots, \sin \gamma_{k_{\min}}, -\sin \gamma_{k_{\min}}, \dots, -\sin \gamma_1$. Furthermore, U_1 can be chosen so that

$$UNU^H = \text{diag}(N_{11}, N_{22}, 0, 0),$$

$N_{11} > 0$ and $N_{22} < 0$. Because of

$$ULU^H + UMU^H = UNU^H,$$

using twice the majorization and extremal property argument as in the beginning of the proof of Proposition 4.2, we can see that

$$ULU^H = \text{diag}(L_{11}, L_{22}, 0, 0), \quad UMU^H = \text{diag}(M_{11}, M_{22}, 0, 0).$$

This shows

$$UY_1Y_1^HU^H = UMU^H + UZ_1Z_1^HU^H = \text{diag}(\text{diag}(M_{11}, M_{22}) + \tilde{Z}, I, 0),$$

i.e., the orthogonal projection onto $\mathcal{Y}$ is identity when restricted to $\mathcal{X} \cap \mathcal{Z}$ and is zero when restricted to $(\mathcal{X} + \mathcal{Z})^\perp$. Therefore, $\mathcal{X} \cap \mathcal{Z} \subset \mathcal{Y} \subset \mathcal{X} + \mathcal{Z}$.

Back to the proof of the necessity, we assume without loss of generality that $k_{\min} = m > 1$ since the case when $k_{\min} = 1$ is trivial and it follows from $\mathcal{X} \cap \mathcal{Z} \subset \mathcal{Y}$ that for the case when $1 < k_{\min} < m$, we can work on $k_{\min}$ -dimensional subspaces $\mathcal{X}_0 = \mathcal{X} \ominus (\mathcal{X} \cap \mathcal{Z})$, $\mathcal{Y}_0 = \mathcal{Y} \ominus (\mathcal{X} \cap \mathcal{Z})$, $\mathcal{Z}_0 = \mathcal{Z} \ominus (\mathcal{X} \cap \mathcal{Z})$ instead. With this assumption, $\sin \theta_m(\mathcal{X}, \mathcal{Z}) > 0$. We also assume without loss of generality that $n = 2m$ since it follows from $\mathcal{Y} \subset \mathcal{X} + \mathcal{Z}$ that we can always confine our attention to a $2m$ -dimensional subspace of $\mathbb{F}^n$ containing $\mathcal{X}, \mathcal{Y}, \mathcal{Z} \subset \mathcal{X} + \mathcal{Z}$.

We will carry further the idea in the proof of Proposition 4.2. We know that the eigenvalues of $X_{11} - Z_{11}$ are given by $2 \sin \gamma_1, \dots, 2 \sin \gamma_m$ which are all positive and $X_{11} - Z_{11} = \text{diag}(g_1(P_1 - R_1), \dots, g_l(P_l - R_l))$. Since $W_j(P_j - R_j)W_j^H = \text{diag}(I_{s_j} - G_j, F_j + I_{t_j})$, $j = 1, \dots, l$, where G_j and F_j are unitary Hermitian matrices, we can see that $g_j > 0$, $F_j = I_{t_j}$, and $G_j = -I_{s_j}$. Hence $P_j = I_{r_j}$ and $R_j = -I_{r_j}$.

Notice that matrices $2UX_1X_1^HU^H - I$, $2UY_1Y_1^HU^H - I$, $2UZ_1Z_1^HU^H - I$ are unitary Hermitian whose traces are all zero. Also notice that $g_l = 1$ and $d_l = 0$. Hence if we replace U by

$$\text{diag}(W_1, \dots, W_l, \tilde{W}_1, \dots, W_l)U$$

with some properly chosen $\tilde{W}_1$, we will have

$$2UX_1X_1^HU^H - I = \begin{bmatrix} P & D \\ D & -P \end{bmatrix}, \quad 2UY_1Y_1^HU^H - I = \begin{bmatrix} Q & D \\ D & -Q \end{bmatrix},$$

$$\text{and} \quad 2UZ_1Z_1^HU^H - I = \begin{bmatrix} -P & D \\ D & P \end{bmatrix},$$

where

$$P = \text{diag}(g_1I_{r_1}, \dots, g_lI_{r_l}) \quad \text{and} \quad Q = \text{diag}(g_1J_{s_1, t_1}, \dots, g_lJ_{s_l, t_l}).$$

Therefore, we have

$$2UX_1X_1^HU^H = \begin{bmatrix} I + P & D \\ D & I - P \end{bmatrix}, \quad 2UY_1Y_1^HU^H = \begin{bmatrix} I + Q & D \\ D & I - Q \end{bmatrix},$$

$$\text{and} \quad 2UZ_1Z_1^HU^H = \begin{bmatrix} I - P & D \\ D & I + P \end{bmatrix}.$$

Since the column space of $UX_1X_1^H U^H$ is the same as the column space of UX_1 , we have

$$U\mathcal{X} = \text{span} \begin{bmatrix} I + P \\ D \end{bmatrix}, \quad U\mathcal{Z} = \text{span} \begin{bmatrix} D \\ I + P \end{bmatrix}.$$

Normalizing the bases and using trigonometry, we get

$$U\mathcal{X} = \text{span} \begin{bmatrix} \cos \frac{1}{2}(\frac{\pi}{2}I - C) \\ \sin \frac{1}{2}(\frac{\pi}{2}I - C) \end{bmatrix}, \quad U\mathcal{Z} = \text{span} \begin{bmatrix} \cos \frac{1}{2}(\frac{\pi}{2}I + C) \\ \sin \frac{1}{2}(\frac{\pi}{2}I + C) \end{bmatrix}.$$

Let $\Theta_{\mathcal{X}} = \frac{1}{2}(\frac{\pi}{2}I - C)$ and $\Theta_{\mathcal{Z}} = \frac{1}{2}(\frac{\pi}{2}I + C)$. Then $\Theta_{\mathcal{Z}} = \Theta_{\mathcal{X}} + C$. Immediately,

$$\begin{bmatrix} \cos \Theta_{\mathcal{Z}} \\ \sin \Theta_{\mathcal{Z}} \end{bmatrix} = \exp \begin{bmatrix} 0 & -\Theta_{\mathcal{X}} \\ \Theta_{\mathcal{X}}^H & 0 \end{bmatrix} \exp \begin{bmatrix} 0 & -C \\ C^H & 0 \end{bmatrix} \exp \begin{bmatrix} 0 & \Theta_{\mathcal{X}} \\ -\Theta_{\mathcal{X}}^H & 0 \end{bmatrix} \begin{bmatrix} \cos \Theta_{\mathcal{X}} \\ \sin \Theta_{\mathcal{X}} \end{bmatrix}.$$

Therefore,

$$\exp \begin{bmatrix} 0 & -\Theta_{\mathcal{X}} \\ \Theta_{\mathcal{X}}^H & 0 \end{bmatrix} \exp \begin{bmatrix} 0 & -C \\ C^H & 0 \end{bmatrix} \exp \begin{bmatrix} 0 & \Theta_{\mathcal{X}} \\ -\Theta_{\mathcal{X}}^H & 0 \end{bmatrix}$$

is a direct rotation from $U\mathcal{X}$ to $U\mathcal{Z}$. Set

$$\Lambda = \frac{1}{2} [I_m - \text{diag}(J_{t_l, s_l}, \dots, J_{t_1, s_1})].$$

Then

$$\begin{aligned} & \begin{bmatrix} I + Q \\ D \end{bmatrix} (I - \Lambda) + \begin{bmatrix} D \\ I - Q \end{bmatrix} \Lambda \\ &= \exp \begin{bmatrix} 0 & -\Theta_{\mathcal{X}} \\ \Theta_{\mathcal{X}}^H & 0 \end{bmatrix} \exp \begin{bmatrix} 0 & -\Lambda C \\ C^H \Lambda & 0 \end{bmatrix} \exp \begin{bmatrix} 0 & \Theta_{\mathcal{X}} \\ -\Theta_{\mathcal{X}}^H & 0 \end{bmatrix} \begin{bmatrix} I + P \\ D \end{bmatrix} \end{aligned}$$

which spans $U\mathcal{Y}$. Finally, if we let

$$X = U^H \exp \begin{bmatrix} 0 & -\Theta_{\mathcal{X}} \\ \Theta_{\mathcal{X}}^H & 0 \end{bmatrix},$$

then

$$X \exp \begin{bmatrix} 0 & -C \\ C^H & 0 \end{bmatrix} X^H$$

is a direct rotation from $\mathcal{X}$ to $\mathcal{Z}$, and

$$\mathcal{Y} = X \exp \begin{bmatrix} 0 & -\Lambda C \\ C^H \Lambda & 0 \end{bmatrix} X^H \mathcal{X}. \quad \square$$

Summarizing the discussions in this section, we conclude that the triangle inequalities for the angular metrics are tighter than those for the gap metrics in the sense that for a strictly convex symmetric gauge function or a Ky Fan k -function Φ , if

$$\Phi(\sin \theta(\mathcal{X}, \mathcal{Z})) = \Phi(\sin \theta(\mathcal{X}, \mathcal{Y})) + \Phi(\sin \theta(\mathcal{Y}, \mathcal{Z})),$$

then

$$\Phi(\theta(\mathcal{X}, \mathcal{Z})) = \Phi(\theta(\mathcal{X}, \mathcal{Y})) + \Phi(\theta(\mathcal{Y}, \mathcal{Z})).$$

5. Related questions and extensions. In this section, we will discuss three issues. The first is on the extremal properties of the direct rotation between two subspaces. The second is on possible generalization of the inequalities in the form of (3.7) and (3.8). The third is on the possibility to use functions more general than the symmetric gauge functions to define metrics on $\mathcal{G}_{m,n}$.

As we have seen in section 2, for $\mathcal{X}, \mathcal{Y} \in \mathcal{G}_{m,n}$, $m \leq n/2$, a direct rotation from $\mathcal{X}$ to $\mathcal{Y}$ is a map of the form

$$R = X \begin{bmatrix} \Gamma & -\Sigma & 0 \\ \Sigma & \Gamma & 0 \\ 0 & 0 & I_{n-2m} \end{bmatrix} X^H = \exp \left[X \begin{bmatrix} 0 & -A \\ A^H & 0 \end{bmatrix} X^H \right],$$

where $A = [\text{diag } \theta(\mathcal{X}, \mathcal{Y}) \quad 0_{m,n-2m}] \in \mathcal{M}_{m,n-m}$ and $X = [X_1 \quad X_2]$ is a special unitary matrix such that the columns of X_1 span $\mathcal{X}$ and those of X_2 span $\mathcal{X}_\perp$. Subspaces $\mathcal{X}, \mathcal{Y}$ are said to be acute if the canonical angles in between are all less than $\pi/2$, or equivalently, $\mathcal{X} \cap \mathcal{Y}_\perp = \{0\}$. It was shown in [3] that the direct rotation between acute subspaces is unique.

Davis and Kahan [3] studied some extremal properties of R among all unitary transformations from $\mathcal{X}$ to $\mathcal{Y}$. They showed that R differs from the identity map in the least amount in some sense; in particular, for all unitarily invariant norm $\|\cdot\|$,

1. $\|(I - V)|_{\mathcal{X}}\|$ is minimized when $V = R$,
2. $\|(I - V)(I - V^H)\|$ is minimized when $V = R$,
3. $\|I - V\|$ is minimized when $V = R$ if $\theta_1(\mathcal{X}, \mathcal{Y}) \leq \pi/3$.

A byproduct of the development in section 2 results in another extremal property, which roughly says that the direct rotation rotates the vectors in $\mathbb{F}^n$ with the smallest amount of angles. For a unitary matrix V , let $\log(V)$ be defined to be the unique skew-Hermitian matrix G with spectrum in $i(-\pi, \pi]$ such that $V = \exp(G)$.

THEOREM 5.1. *Let $\|\cdot\|$ be a unitarily invariant norm. Assume V is a unitary matrix satisfying $V\mathcal{X} = \mathcal{Y}$. Then $\|\log(V)\|$ is minimized when $V = R$.*

Proof. The proof will be sketchy since it follows from the same idea as in the development of section 2. We know that V only differs from R locally in $\mathcal{X}$ and $\mathcal{X}_\perp$; i.e., there exists a unitary matrix $Q = \text{diag}(Q_1, Q_2)$ with $Q_1 \in \mathcal{U}_m$ and $Q_2 \in \mathcal{U}_{n-m}$ such that

$$V = X \exp \begin{bmatrix} 0 & -A \\ A^H & 0 \end{bmatrix} Q X^H.$$

Hence

$$(5.1) \quad X^H V X = \exp \begin{bmatrix} 0 & -A \\ A^H & 0 \end{bmatrix} Q.$$

Taking the inverse, we get

$$X^H V^H X = Q^H \exp \begin{bmatrix} 0 & A \\ -A^H & 0 \end{bmatrix}.$$

Multiplying $J = \text{diag}(I_m, -I_{n-m})$ from both sides and noticing that J commutes with Q , we get

$$(5.2) \quad J X^H V^H X J = Q^H \exp \begin{bmatrix} 0 & -A \\ A^H & 0 \end{bmatrix}.$$

Multiplying (5.1) and (5.2), we obtain

$$\exp \begin{bmatrix} 0 & -2A \\ 2A^H & 0 \end{bmatrix} = X^H V X J X^H V^H X J.$$

Let $G = \log(V)$, we have

$$\exp \begin{bmatrix} 0 & -2A \\ 2A^H & 0 \end{bmatrix} = \exp [U_1 G U_1^H + U_2 G^H U_2^H]$$

for some $U_1, U_2 \in \mathcal{U}_n$. The rest of the proof follows from that of Theorem 3.1. $\square$

The extremal property of the direct rotation for $\|\log(V)\|$ is free of the unpleasant condition required in that for $\|I - V\|$.

Example 2. Compare the reflection $V = \begin{bmatrix} \cos \theta & \sin \theta \\ \sin \theta & -\cos \theta \end{bmatrix}$ and direct rotation $R = \begin{bmatrix} \cos \theta & -\sin \theta \\ \sin \theta & \cos \theta \end{bmatrix}$ mapping $\text{span} \begin{bmatrix} 1 \\ 0 \end{bmatrix}$ to $\text{span} \begin{bmatrix} \cos \theta \\ \sin \theta \end{bmatrix}$. Clearly $\pi = \|\log(V)\|_2 \geq \|\log(R)\|_2 = 2\theta$, whereas $\|I - V\|_2 = 2$ and $\|I - R\|_2 = 4 \sin(\theta/2)$, which can be ordered either way depending on θ , as pointed out in [3].

The second issue concerns the possible extension of inequalities in the form of (3.7)–(3.8) on canonical angles between three subspaces, which resemble the corresponding inequalities (3.1)–(3.2) on singular values of matrix sums. In the case when $\theta(\mathcal{X}, \mathcal{Y})$ and $\theta(\mathcal{Y}, \mathcal{Z})$ are sufficiently small so that $\theta_1(\mathcal{X}, \mathcal{Y}) + \theta_1(\mathcal{Y}, \mathcal{Z}) < \pi/2$, inequalities (3.7)–(3.8) immediately follow inequalities (3.1)–(3.2) after taking the logarithm from both sides of (2.5). There are some recent results on developing more inequalities of the form

$$(5.3) \quad \sum_{t \in T} \sigma_t(C) \leq \sum_{r \in R} \sigma_r(A) + \sum_{s \in S} \sigma_s(B),$$

where $R = \{r_1 < \dots < r_k\}$, $S = \{s_1 < \dots < s_k\}$, $T = \{t_1 < \dots < t_k\}$ are increasingly ordered subsets of $\{1, \dots, \min\{m, n\}\}$ with the same cardinality $k \leq \min\{m, n\}$ [8]. An example of a family of such inequalities beyond those in (3.1)–(3.2) is given by R, S, T satisfying $r_k + s_k \leq \min\{m, n\} + k$ and $t_j = r_j + s_j - j$ for $1 \leq j \leq k$. It is natural to conjecture that

$$(5.4) \quad \sum_{t \in T} \theta_t(\mathcal{X}, \mathcal{Z}) \leq \sum_{r \in R} \theta_r(\mathcal{X}, \mathcal{Y}) + \sum_{s \in S} \theta_s(\mathcal{Y}, \mathcal{Z})$$

whenever the sets R, S, T make inequality (5.3) satisfied (with $\min\{m, n\}$ replaced by m). However, this is not immediately clear, even in the case when $\theta(\mathcal{X}, \mathcal{Y})$ and $\theta(\mathcal{Y}, \mathcal{Z})$ are sufficiently small, due to the unpleasant fact that the right-hand side of (2.5) involves the sum of three terms, two of which are related to $\theta(\mathcal{Y}, \mathcal{Z})$.

The last issue is whether we can use more general functions of the canonical angles to define unitarily metrics on $\mathcal{G}_{m,n}$. One may consider a function $\Psi : [0, \pi/2]^m \rightarrow \mathbb{R}$ and define $\delta(\mathcal{X}, \mathcal{Y}) = \Psi(\theta(\mathcal{X}, \mathcal{Y}))$ for $\mathcal{X}, \mathcal{Y} \in \mathcal{G}_{m,n}$. It would be interesting to determine the necessary and sufficient conditions on Ψ for δ to be a metric on $\mathcal{G}_{m,n}$. Evidently, such a function δ always satisfies $\delta(\mathcal{X}, \mathcal{Y}) = \delta(\mathcal{Y}, \mathcal{X})$ and $\delta(U\mathcal{X}, U\mathcal{Y}) = \delta(\mathcal{X}, \mathcal{Y})$ for all $U \in \mathcal{U}_n$. To ensure that $\delta(\mathcal{X}, \mathcal{Y}) \geq 0$ so that equality holds if and only if $\mathcal{X} = \mathcal{Y}$, the function Ψ must satisfy the following condition:

- (P1) Ψ is positive, i.e., $\Psi(\xi) \geq 0$ for all $\xi \in [0, \pi/2]^m$, and the equality holds if and only if $\xi = 0$.

To ensure triangle inequality, the following is necessary:

(P2) Ψ is subadditive, i.e., $\Psi(\xi + \zeta) \leq \Psi(\xi) + \Psi(\zeta)$ for any $\xi, \zeta \in [0, \pi/2]^m$ satisfying $\xi + \zeta \in [0, \pi/2]^m$.

Suppose Ψ also satisfies the following:

(P3) Ψ is strongly Schur convex (or strongly isotone), i.e., $\Psi(\xi) \leq \Psi(\zeta)$ whenever $\xi, \zeta \in [0, \pi/2]^m$ and ξ is weakly majorized by ζ .

Then we have

$$\begin{aligned} \delta(\mathcal{X}, \mathcal{Z}) &= \Psi(\theta(\mathcal{X}, \mathcal{Z})) \leq \Psi(\theta(\mathcal{X}, \mathcal{Y}) + \theta(\mathcal{Y}, \mathcal{Z})) \\ &\leq \Psi(\theta(\mathcal{X}, \mathcal{Y})) + \Psi(\theta(\mathcal{Y}, \mathcal{Z})) = \delta(\mathcal{X}, \mathcal{Y}) + \delta(\mathcal{Y}, \mathcal{Z}), \end{aligned}$$

i.e., function δ satisfies the triangle inequality and hence is a unitarily invariant metric. This may allow us to go a small step beyond the set of symmetric gauge functions. However, it is not clear at this stage what functions satisfy (P1), (P2), and (P3) other than symmetric gauge functions and those of the form $f \circ \Phi_1$, where f is an increasing subadditive scalar function. Also, (P3) is far from a necessary condition for δ to be a metric on $\mathcal{G}_{m,n}$. It is not satisfied by the gap metrics defined in (3.3), where $\Psi(\xi) = \Phi(\sin \xi)$, nor the Hausdorff metric defined in (3.4), where $\Psi(\xi) = \Phi(2 \sin(\xi/2))$. It would be interesting to weaken the condition (P3) so that it is satisfied by all known metrics so far, or even becomes necessary.

Example 3. Consider $\Psi(\xi_1, \xi_2) = \sin \xi_1 + \sin \xi_2$. Then the corresponding function δ is the gap metric on $\mathcal{G}_{2,n}$ corresponding to the Ky Fan 2-function. However, for $\xi = (\pi/4, \pi/4)$ and $\zeta = (\pi/2, 0)$, it follows that ξ is majorized by ζ , but $\Psi(\xi) = \sqrt{2} > 1 = \Psi(\zeta)$.

Acknowledgments. We thank Professor Chandler Davis for some helpful discussions. In particular, he suggested a different approach (using integration theory) which may lead to a proof of Theorem 3.1. We also would like to thank Nam-Kiu Tsing, Yimin Wei, Roy Mathias, and Pierre-Antoine Absil for helpful discussions, and Wasin So for bringing the reference [31] to the authors' attention.

REFERENCES

- [1] S. N. AFRIAT, *Orthogonal and oblique projectors and the characteristics of pairs of vector spaces*, Proc. Cambridge Philos. Soc., 53 (1957), pp. 800–816.
- [2] S. AGNIHOTRI AND C. WOODWARD, *Eigenvalues of products of unitary matrices and quantum Schubert calculus*, Math. Res. Lett., 5 (1998), pp. 817–836.
- [3] C. DAVIS AND W. M. KAHAN, *The rotation of eigenvectors by a perturbation. III*, SIAM J. Numer. Anal., 7 (1970), pp. 1–46.
- [4] K. DE COCK AND B. DE MOOR, *Subspace angles between ARMA models*, Systems Control Lett., 46 (2002), pp. 265–270.
- [5] C. ECKART AND G. YOUNG, *The approximation of one matrix by another of lower rank*, Psychometrika, 1 (1936), pp. 211–218.
- [6] A. EDELMAN, T. A. ARIAS, AND S. T. SMITH, *The geometry of algorithms with orthogonality constraints*, SIAM J. Matrix Anal. Appl., 20 (1998), pp. 303–353.
- [7] K. FAN, *Maximum properties and inequalities for the eigenvalues of completely continuous operators*, Proc. Nat. Acad. Sci. U.S.A., 37 (1951), pp. 760–766.
- [8] W. FULTON, *Eigenvalues, invariant factors, highest weights, and Schubert calculus*, Bull. Amer. Math. Soc., 37 (2000), pp. 209–249.
- [9] T. T. GEORGIOU AND M. C. SMITH, *Optimal robustness in the gap metric*, IEEE Trans. Automat. Control, 35 (1990), pp. 673–685.
- [10] I. C. GOHBERG AND M. G. KREIN, *Introduction to the Theory of Linear Nonselfadjoint Operators*, AMS, Providence, RI, 1969.

- [11] G. H. GOLUB AND H. ZHA, *Perturbation analysis of the canonical correlations of matrix pairs*, Linear Algebra Appl., 210 (1994), pp. 3–28.
- [12] H. HOTELLING, *Analysis of a complex of statistical variables into principal components*, J. Educ. Psychol., 24 (1933), pp. 417–441 and 498–520.
- [13] H. HOTELLING, *Relations between two sets of variates*, Biometrika, 28 (1936), pp. 321–377.
- [14] C. JORDAN, *Mémoire sur les formes bilinéaires*, J. Math. Pures Appl., 19 (1874), pp. 35–54.
- [15] C. JORDAN, *Essai sur la géométrie à n dimensions*, Bull. Soc. Math., 3 (1875), pp. 103–174.
- [16] T. KATO, *Perturbation Theory for Linear Operators*, Springer-Verlag, Berlin, New York, 1995.
- [17] V. C. KLEMA AND A. J. LAUB, *The singular value decomposition: Its computation and some applications*, IEEE Trans. Automat. Control, AC-25 (1980), pp. 164–176.
- [18] A. V. KNYAZEV AND M. E. ARGENTATI, *Principal angles between subspaces in an A -based scalar product: Algorithms and perturbation estimates*, SIAM J. Sci. Comput., 23 (2002), pp. 2008–2040.
- [19] C. K. LI, *Matrices with some extremal properties*, Linear Algebra Appl., 101 (1988), pp. 255–267.
- [20] A. MARSHALL AND I. OLKIN, *Inequalities: Theory of Majorization and Its Applications*, Academic Press, Boston, 1979.
- [21] R. J. MARTIN, *A metric for ARMA processes*, IEEE Trans. Signal Process., 48 (2000), pp. 1164–1169.
- [22] L. MIRSKY, *Results and problems in the theory of doubly stochastic matrices*, Z. Wahrsch. Verw. Gebiete, 1 (1963), pp. 319–334.
- [23] J. MIAO AND A. BEN-ISREAL, *On principal angles between subspaces in $\mathbb{R}^n$* , Linear Algebra Appl., 171 (1992), pp. 81–98.
- [24] L. QIU AND E. J. DAVISON, *Pointwise gap metrics on transfer matrices*, IEEE Trans. Automat. Control, 37 (1992), pp. 741–758.
- [25] L. QIU AND E. J. DAVISON, *Feedback stability under simultaneous gap metric uncertainties in plant and controller*, Systems Control Lett., 18 (1992), pp. 9–22.
- [26] E. SCHMIDT, *Zur theorie der linearen und nichtlinearen integralgleichungen*, Math. Ann., 63 (1907), pp. 433–476.
- [27] G. W. STEWART AND J.-G. SUN, *Matrix Perturbation Theory*, Academic Press, Boston, 1990.
- [28] G. W. STEWART, *Error and perturbation bounds for subspaces associated with certain eigenvalue problems*, SIAM Rev., 15 (1973), pp. 727–764.
- [29] J.-G. SUN, *Perturbation of angles between linear subspaces*, J. Comput. Math., 5 (1987), pp. 58–61.
- [30] R. C. THOMPSON, *Singular values, diagonal elements, and convexity*, SIAM J. Appl. Math., 32 (1977), pp. 39–63.
- [31] R. C. THOMPSON, *Proof of a conjectured exponential formula*, Linear Multilinear Algebra, 9 (1986), pp. 187–197.
- [32] P. VAN OVERSCHEE AND B. DE MOOR, *Subspace Identification for Linear Systems: Theory–Implementation–Applications*, Kluwer Academic, Boston, 1996.
- [33] G. VINNICOMBE, *Frequency domain uncertainty and the graph topology*, IEEE Trans. Automat. Control, 38 (1993), pp. 1371–1383.
- [34] J. VON NEUMANN, *Some matrix-inequalities and metrization of matrix-space*, Toms. Univ. Rev., 1 (1937), pp. 286–300.
- [35] P.-A. WEDIN, *On angles between subspaces*, in Matrix Pencils, Springer-Verlag, New York, 1983, pp. 263–285.
- [36] A. WEINSTEIN, *Almost invariant submanifolds for compact group actions*, J. Eur. Math. Soc. (JEMS), 2 (2000), pp. 53–86.
- [37] H. K. WIMMER, *Canonical angles of unitary spaces and perturbations of direct complements*, Linear Algebra Appl., 287 (1999), pp. 373–379.
- [38] Y.-C. WONG, *Differential geometry of Grassmann manifold*, Proc. Natl. Acad. Sci. U.S.A., 57 (1967), pp. 589–594.

LEAST-SQUARES COVARIANCE MATRIX ADJUSTMENT*

STEPHEN BOYD[†] AND LIN XIAO[‡]

Abstract. We consider the problem of finding the smallest adjustment to a given symmetric $n \times n$ matrix, as measured by the Euclidean or Frobenius norm, so that it satisfies some given linear equalities and inequalities, and in addition is positive semidefinite. This least-squares covariance adjustment problem is a convex optimization problem, and can be efficiently solved using standard methods when the number of variables (i.e., entries in the matrix) is modest, say, under 1000. Since the number of variables is $n(n+1)/2$, this corresponds to a limit around $n = 45$. Malick [*SIAM J. Matrix Anal. Appl.*, 26 (2005), pp. 272–284] studies a closely related problem and calls it the semidefinite least-squares problem. In this paper we formulate a dual problem that has no matrix inequality or matrix variables, and a number of (scalar) variables equal to the number of equality and inequality constraints in the original least-squares covariance adjustment problem. This dual problem allows us to solve far larger least-squares covariance adjustment problems than would be possible using standard methods. Assuming a modest number of constraints, problems with $n = 1000$ are readily solved by the dual method. The dual method coincides with the dual method proposed by Malick when there are no inequality constraints and can be obtained as an extension of his dual method when there are inequality constraints. Using the dual problem, we show that in many cases the optimal solution is a low rank update of the original matrix. When the original matrix has structure, such as sparsity, this observation allows us to solve very large least-squares covariance adjustment problems.

Key words. matrix nearness problems, covariance matrix, least-squares, semidefinite least-squares

AMS subject classifications. 15A18, 90C06, 90C22, 90C46, 93E24

DOI. 10.1137/040609902

1. The problem. This paper concerns the following problem. We are given a symmetric matrix G , and seek X , the nearest (in the least-squares sense) symmetric matrix that is positive semidefinite and in addition satisfies some given linear equalities and inequalities. This can be expressed as the optimization problem

$$(1.1) \quad \begin{aligned} & \text{minimize} && (1/2)\|X - G\|_F^2 \\ & \text{subject to} && X \succeq 0, \\ & && \text{Tr } A_i X = b_i, \quad i = 1, \dots, p, \\ & && \text{Tr } C_j X \leq d_j, \quad j = 1, \dots, m. \end{aligned}$$

Here the matrices X , G , A_i , and C_j are $n \times n$ and symmetric (and real). The (real) matrix X is the optimization variable; the matrices G , A_i , and C_j , and the scalars b_i and d_j , are the problem data. We note that the optimization variable X has dimension $n(n+1)/2$, i.e., it contains $n(n+1)/2$ independent scalar variables. In (1.1), Tr denotes the trace of a matrix, $\succeq$ denotes matrix inequality, and $\|\cdot\|_F$

*Received by the editors June 11, 2004; accepted for publication (in revised form) by L. Vandenberghe April 4, 2005; published electronically November 22, 2005.

<http://www.siam.org/journals/simax/27-2/60990.html>

[†]Department of Electrical Engineering, Stanford University, Stanford, CA 94305-9510 (boyd@stanford.edu).

[‡]Center for the Mathematics of Information, California Institute of Technology, Pasadena, CA 91125-9300 (lxiao@caltech.edu).

denotes the Frobenius norm, i.e.,

$$\|U\|_F = (\text{Tr } U^T U)^{1/2} = \left(\sum_{i,j=1}^n U_{ij}^2 \right)^{1/2}.$$

Since any real-valued linear function f on the set of $n \times n$ symmetric matrices can be expressed as $f(U) = \text{Tr } AU$ for some symmetric $n \times n$ matrix A , we see that the constraints in (1.1) are a general set of p linear equality constraints and m linear inequality constraints. For reasons to be explained below, we refer to problem (1.1) as the *least-squares covariance adjustment problem* (LSCAP).

LSCAP is a type of *matrix nearness problem*, i.e., the problem of finding a matrix that satisfies some property and is nearest to a given one. In the least-squares covariance adjustment problem, we use the Frobenius norm, the given matrix is symmetric, and the property is that the matrix belongs to a polyhedron, i.e., the solution set of some linear equalities and inequalities. See Higham [6] for a recent survey of matrix nearness problems. A related geometric interpretation of the LSCAP (1.1) is as follows: We are given a symmetric matrix G , and wish to compute the (Frobenius norm) projection of G onto the intersection of a general polyhedron, described by linear equalities and inequalities, and the cone of positive semidefinite matrices.

LSCAP (1.1) comes up in several contexts. One is in making adjustments to a symmetric matrix so that it is consistent with prior knowledge or assumptions, and is a valid covariance matrix. We interpret X as the covariance matrix of a zero mean random n -vector z , and suppose that the variances of some linear functions of z are known, e.g.,

$$\mathbf{E} \|F_i^T z\|^2 = \text{Tr } F_i^T X F_i = b_i, \quad i = 1, \dots, p.$$

We are given G , which is some approximation of X (e.g., a sample covariance matrix), and wish to find the covariance matrix nearest G that is also consistent with the prior information. This results in problem (1.1) with $A_i = F_i F_i^T$ (using $\text{Tr } \|F_i X\|^2 = \text{Tr } X(F_i F_i^T)$). If we are also given some lower or upper bounds on the variances of some linear combinations of z , we obtain an LSCAP with inequality constraints.

We will also consider a special case of the basic problem (1.1), in which there is only one equality constraint, defined by a matrix of rank one:

$$(1.2) \quad \begin{aligned} & \text{minimize} && (1/2) \|X - G\|_F^2 \\ & \text{subject to} && X \succeq 0, \quad c^T X c = b. \end{aligned}$$

This is the problem of finding the covariance matrix closest to G that matches the given variance of one given linear function of the underlying random variable.

1.1. Previous work. Some special cases of the LSCAP have simple solutions, or have been addressed in the literature. The simplest case is when there are no constraints on X other than positive semidefiniteness. In this case the LSCAP reduces to finding the projection of a symmetric matrix onto the positive semidefinite cone, which has a well-known solution based on the eigenvalue decomposition of G (given in section 1.3). Another example is the problem of finding the nearest correlation matrix, i.e., the nearest symmetric positive semidefinite matrix with unit diagonal elements. This particular problem is addressed in Higham [7], where an alternating projections method is proposed for solving the problem.

There is a large literature on matrix nearness problems; see, for example, the references in the survey [6]. In much of this work, the original matrix (which we denote G) is nonsymmetric, which complicates the problem a bit. For example, in [5], Higham considers the problem of finding the nearest positive semidefinite matrix to a given (nonsymmetric) matrix with no other constraints.

Our approach and methods are related to others that have been proposed for other matrix nearness problems. In particular, we find ideas related to the dual of the original problem in, e.g., [7]; the idea of exploiting structure to evaluate the projection of a matrix onto the positive semidefinite cone also appears in [7]. Our method for the special case (1.2) involves a one parameter search, which can be done by a bisection algorithm, or a guarded Newton method. This is similar to methods proposed in [4, 5] and several of the papers cited in Higham's survey paper for (other) matrix nearness problems.

By far the closest previous work is the recent paper [11] by Malick, which appeared while this paper was under review. Malick considers the general problem of computing the Euclidean projection of a point onto the intersection of a cone and an affine set, and the special case in which the cone is the positive semidefinite cone, which he calls *semidefinite least-squares* (SDLS). His SDLS problem coincides with our LSCAP, with no equality constraints; the inequality constraints can, however, be handled by introducing slack variables, and including the slack nonnegativity constraints in the cone, so SDLS can be said to be more general than LSCAP. Our approach and Malick's approach are very similar with some differences that we will point out in what follows. In fact, some of our ideas and Malick's can be combined to create a method that can very efficiently solve large-scale SDLS problems.

1.2. Basic properties of LSCAP. The LSCAP (1.1) is a *convex optimization problem*, since the objective is convex, the constraint functions are linear, and the semidefiniteness inequality $X \succeq 0$ is convex. (The constraint $X \succeq 0$ is called a generalized or cone constraint, or more specifically, a linear matrix inequality.) If the problem (1.1) is feasible, then it has a unique solution, since the objective function is strictly convex and has bounded sublevel sets. We will denote the solution as X^* .

The observation that the LSCAP is a convex optimization problem have several important implications. The first is that the LSCAP is readily solvable, by standard interior-point methods, at least when the number of variables is modest, say, under 1000 (which corresponds to n around 45). General purpose interior-point solvers (that do not exploit problem structure) have a complexity that scales at least as fast as the number of variables cubed, so the overall complexity of solving the LSCAP problem, using this approach, scales at least as fast as n^6 . For background on convex optimization and interior-point methods, see, e.g., [3].

The second consequence of convexity of LSCAP is that its associated *dual problem* (defined below) is sharp (provided a technical condition holds) and can be used to solve the original LSCAP. This is the central idea of our method, and is described in more detail in section 2.

The LSCAP problem is closely related to *semidefinite programming* (SDP) [18, 17, 3]. Like LSCAP, SDP is an optimization problem with a symmetric matrix variable, a semidefiniteness constraint, and linear equality and inequality constraints. In an SDP, the objective is a linear function; in LSCAP the objective is (a simple) convex quadratic function. In fact, it is possible to express the LSCAP problem as an SDP, using a standard trick for converting a quadratic objective into a linear matrix inequality and a linear objective (see, e.g., [18]). In principal, this allows us to solve the

LSCAP using standard algorithms for solving SDPs. Without exploiting any particular structure in the resulting SDP, however, this approach will be quite inefficient. The observation that LSCAP (in the SDLS form) is convex, and can be formulated as an SDP, but one that is difficult to solve using the standard methods can be found in Malick [11] and Higham [7].

1.3. Projection on the positive semidefinite cone. Here we introduce some notation and standard material that we will use in what follows. For a symmetric $n \times n$ matrix X , we sort the eigenvalues in decreasing order,

$$\lambda_1 \geq \lambda_2 \geq \cdots \geq \lambda_n,$$

and use the notation $\lambda_i(X)$, for example, to specify the matrix if it is not clear.

The positive and negative semidefinite parts of X , denoted by X_+ and X_- , respectively, are defined implicitly by the conditions

$$X = X_+ - X_-, \quad X_+ = X_+^T \succeq 0, \quad X_- = X_-^T \succeq 0, \quad X_+ X_- = 0.$$

The positive semidefinite part X_+ is the projection of X onto the positive semidefinite cone, i.e., we have

$$\|X - X_+\|_F = \|X_-\|_F \leq \|X - Z\|_F$$

for any positive semidefinite Z . In a similar way, $\|X + Z\|_F$ is minimized, over all positive semidefinite matrices Z , by the choice $Z = X_-$ (see, e.g., [3, section 8.1.1] or [8]).

We can express the positive and negative semidefinite parts explicitly as

$$X_+ = \sum_{\lambda_i > 0} \lambda_i q_i q_i^T, \quad X_- = \sum_{\lambda_i < 0} -\lambda_i q_i q_i^T,$$

where $X = \sum_{i=1}^n \lambda_i q_i q_i^T$ is an eigendecomposition of X , i.e., $q_1, \dots, q_n$ is a set of orthonormal eigenvectors of X with corresponding eigenvalues $\lambda_1, \dots, \lambda_n$. We also note, for future reference, that to compute X_+ and X_- , it suffices to find the negative eigenvalues of X and the associated eigenvectors; X_+ can then be computed as $X_+ = X + X_-$.

2. The dual problem.

2.1. Lagrangian and dual function. We introduce the Lagrange multipliers $\nu_1, \dots, \nu_p$ associated with the equality constraints $\mu_1, \dots, \mu_m$ associated with the inequality constraints, and the symmetric $n \times n$ matrix Z associated with the matrix inequality $X \succeq 0$ (which we write as $-X \preceq 0$). The Lagrangian of problem (1.1) is then

$$\begin{aligned} L(X, Z, \nu, \mu) &= (1/2)\|X - G\|_F^2 - \mathbf{Tr} Z X + \sum_{i=1}^p \nu_i (\mathbf{Tr} A_i X - b_i) + \sum_{j=1}^m \mu_j (\mathbf{Tr} C_j X - d_j) \\ &= (1/2)\|X - G\|_F^2 + \mathbf{Tr} X \left(-Z + \sum_{i=1}^p \nu_i A_i + \sum_{j=1}^m \mu_j C_j \right) - \nu^T b - \mu^T d \\ &= (1/2)\|X - G\|_F^2 + \mathbf{Tr} X (-Z + A(\nu) + C(\mu)) - \nu^T b - \mu^T d, \end{aligned}$$

where we use the notation

$$A(\nu) = \sum_{i=1}^p \nu_i A_i, \quad C(\mu) = \sum_{j=1}^m \mu_j C_j.$$

To form the dual function, we must minimize the Lagrangian over X , which we can do by setting the gradient with respect to X to zero (since L is convex quadratic in X). This yields

$$X - G - Z + A(\nu) + C(\mu) = 0,$$

so the X that minimizes L is given by

$$(2.1) \quad X = G - A(\nu) - C(\mu) + Z.$$

Substituting this expression for X back into the Lagrangian we obtain the dual function:

$$\begin{aligned} g(Z, \nu, \mu) &= \inf_X L(X, Z, \nu, \mu) \\ &= (1/2)\| -A(\nu) - C(\mu) + Z \|_F^2 - \nu^T b - \mu^T d \\ &\quad + \text{Tr}(G - A(\nu) - C(\mu) + Z)(-Z + A(\nu) + C(\mu)) \\ &= -(1/2)\| -A(\nu) - C(\mu) + Z + G \|_F^2 + (1/2)\|G\|_F^2 - \nu^T b - \mu^T d. \end{aligned}$$

This is a concave quadratic function of the dual variables Z , ν , and μ .

2.2. Dual problem and properties. The dual problem is

$$(2.2) \quad \begin{array}{ll} \text{maximize} & -(1/2)\| -A(\nu) - C(\mu) + Z + G \|_F^2 + (1/2)\|G\|_F^2 - \nu^T b - \mu^T d \\ \text{subject to} & Z \succeq 0, \quad \mu \succeq 0, \end{array}$$

where the symbol $\succeq$ between vectors means elementwise, and the (dual) variables are Z , ν , and μ . This problem is similar to the original LSCAP: it has a symmetric matrix variable (as well as two vector variables), a quadratic objective, a positive semidefiniteness constraint, and a set of (scalar) nonnegativity constraints.

Weak duality always holds for the dual problem (2.2): if Z , ν , and μ are dual feasible, i.e., $Z \succeq 0$ and $\mu \succeq 0$, then the dual objective is a lower bound on the optimal value of the LSCAP (1.1). If the LSCAP (1.1) is *strictly feasible*, i.e., there exists an $X \succ 0$ that satisfies the linear equalities and inequalities in (1.1), then *strong duality* holds: there exist Z^* , ν^* , μ^* that are optimal for the dual problem (2.2) with dual objective equal to the optimal value of the LSCAP (1.1). Moreover, we can recover the optimal solution of the LSCAP (1.1) from the dual optimal variables using

$$(2.3) \quad X^* = G - A(\nu^*) - C(\mu^*) + Z^*$$

(which is the choice of X that minimizes the Lagrangian, when $Z = Z^*$, $\nu = \nu^*$, and $\mu = \mu^*$). This follows from convexity of the original LSCAP, and strict convexity of the Lagrangian with respect to the primal variable X ; see, e.g., [3].

Although we will end up deriving this result directly, general convex optimization theory tells us that X^* and Z^* satisfy the *complementarity condition* $Z^* X^* = 0$. Since X^* and Z^* are both positive semidefinite, this means that the intersection of their ranges is $\{0\}$.

2.3. Simplified dual problem. It is possible to analytically maximize over the variable Z in (2.2), using the fact (from section 1.3) that

$$\| -A(\nu) - C(\mu) + Z + G \|_F^2$$

is minimized over the choice of $Z \succeq 0$ by choosing

$$(2.4) \quad Z = (G - A(\nu) - C(\mu))_- ,$$

which gives

$$\begin{aligned} \| -A(\nu) - C(\mu) + Z + G \|_F^2 &= \| (G - A(\nu) - C(\mu))_+ \|_F^2 \\ &= \sum_{\lambda_i > 0} \lambda_i^2 (G - A(\nu) - C(\mu)) . \end{aligned}$$

Using the result (2.4), we can simplify the dual problem (2.2) to

$$(2.5) \quad \begin{array}{ll} \text{maximize} & -(1/2) \| (G - A(\nu) - C(\mu))_+ \|_F^2 + (1/2) \| G \|_F^2 - \nu^T b - \mu^T d \\ \text{subject to} & \mu \succeq 0, \end{array}$$

where the variables are ν and μ . We will refer to this as the *simplified dual LSCAP problem*. The simplified dual problem has no matrix variable or inequality and has a number of (scalar) variables equal to the number of equality and inequality constraints in the original LSCAP.

The simplified dual problem is, of course, a convex optimization problem (since the objective is concave, and is to be maximized). We also note that the objective function is differentiable, but not twice differentiable. To see this we define ϕ , for symmetric W , as

$$\phi(W) = (1/2) \| W_+ \|_F^2 = (1/2) \sum_i (\max\{0, \lambda_i(W)\})^2 ,$$

the sum of the squares of the positive eigenvalues. Its gradient can be expressed as

$$(2.6) \quad \nabla \phi(W) = W_+ ,$$

by which we mean the following: for symmetric V , we have

$$\phi(W + V) = \phi(W) + \text{Tr } V W_+ + o(V) .$$

This can be verified several ways, for example, using standard perturbation theory for the eigenvalues of a symmetric matrix, or results from Borwein and Lewis [2, section 5.2].

From the formula (2.6), we conclude that $\nabla \phi(W)$ is continuous, indeed, with Lipschitz constant one: for any symmetric V and W ,

$$\| \nabla \phi(V) - \nabla \phi(W) \|_F \leq \| V - W \|_F .$$

Whenever W has no zero eigenvalues, $\nabla \phi(W)$ is differentiable; but when W has a zero eigenvalue, $\nabla \phi(W)$ is not differentiable. In other words, ϕ is twice differentiable, except at points W with a zero eigenvalue.

Using (2.6) we can find expressions for the gradient of the objective of the simplified dual (2.5), which we denote

$$\psi(\nu, \mu) = -(1/2) \|(G - A(\nu) - C(\mu))_+\|_F^2 + (1/2) \|G\|_F^2 - \nu^T b - \mu^T d.$$

We have

$$\frac{\partial \psi}{\partial \nu_i} = \mathbf{Tr}(G - A(\nu) - C(\mu))_+ A_i - b_i, \quad \frac{\partial \psi}{\partial \mu_j} = \mathbf{Tr}(G - A(\nu) - C(\mu))_+ C_j - d_j.$$

Defining $X(\nu, \mu)$ as $X(\nu, \mu) = (G - A(\nu) - C(\mu))_+$, we see that the gradients of the dual objective can be expressed as

$$\frac{\partial \psi}{\partial \nu_i} = \mathbf{Tr} A_i X(\nu, \mu) - b_i, \quad \frac{\partial \psi}{\partial \mu_j} = \mathbf{Tr} C_j X(\nu, \mu) - d_j,$$

which are exactly the residuals for the constraints in the primal LSCAP with $X = X(\nu, \mu)$.

The simplified dual (2.5) coincides with the dual derived by Malick for the case when there are no inequality constraints in the LSCAP, and is obtained as a simple extension of his dual, when inequality constraints are included (see [11, Remark 4.3]). He also observes that the dual objective is differentiable, and has a gradient with Lipschitz constant one.

From (2.3) and (2.4) we find that

$$(2.7) \quad X^* = (G - A(\nu^*) - C(\mu^*))_+.$$

The formula (2.7) shows that the optimal solution of the LSCAP problem is always obtained by adding a linear combination of the constraint data matrices to the original matrix, and then projecting the result onto the positive semidefinite cone.

More generally, suppose that ν and μ are *any* feasible values of the dual variables (i.e., $\mu_j \geq 0$), not necessarily optimal. Then the dual objective $\psi(\nu, \mu)$ gives a lower bound on the optimal value of the primal LSCAP. The matrix $X(\nu, \mu)$ is always positive semidefinite (since it is the positive semidefinite part of $G - A(\nu) - C(\mu)$), but does not satisfy the original equality and inequality constraints, unless ν and μ are optimal.

2.4. Rank of optimal adjustment. Consider $X^* - G$, which is the optimal adjustment made to G . We have

$$(2.8) \quad \begin{aligned} X^* - G &= (G - A(\nu^*) - C(\mu^*))_+ - G \\ &= -A(\nu^*) - C(\mu^*) + (G - A(\nu^*) - C(\mu^*))_-. \end{aligned}$$

This shows that the optimal adjustment made to G is a linear combination of the constraint matrices, plus the negative part of $G - A(\nu^*) - C(\mu^*)$.

We can draw many interesting conclusions from this observation. We start by observing that the number of negative eigenvalues of $G - A(\nu^*) - C(\mu^*)$ is no more than the number of negative eigenvalues of G , plus the sum of the ranks of A_i , plus the sum of the ranks of C_j . In other words, we have

$$(2.9) \quad \mathbf{Rank}(G - A(\nu^*) - C(\mu^*))_- \leq \mathbf{Rank}(G_-) + \sum_i \mathbf{Rank}(A_i) + \sum_j \mathbf{Rank}(C_j).$$

We also have

$$\mathbf{Rank}(-A(\nu^*) - C(\mu^*)) \leq \sum_i \mathbf{Rank}(A_i) + \sum_j \mathbf{Rank}(C_j),$$

so, using the result (2.8) above, we conclude

$$\mathbf{Rank}(X^* - G) \leq \mathbf{Rank}(G_-) + 2 \sum_i \mathbf{Rank}(A_i) + 2 \sum_j \mathbf{Rank}(C_j).$$

This shows that if G is positive semidefinite, and there are only a small number of constraints, with low rank data matrices, then the optimal adjustment is low rank, i.e., X^* is a low rank update of G .

As another example, suppose that G is a *factor model covariance matrix*; i.e., it can be expressed as a sum of a nonnegative diagonal matrix and a positive semidefinite matrix of low rank, say, k . (This means that the underlying random variables can be explained by k common factors, plus some independent noise in each component.) We also assume that the constraint matrices A_i and C_j are rank one and positive semidefinite. For this example, the rank of the optimal adjustment $X^* - G$ cannot exceed $2(p + m)$, so the optimal adjusted covariance matrix X^* will also be a factor model (indeed, with the same diagonal part as G , and a number of additional factors not exceeding twice the number of equalities and inequalities).

3. Solution via simplified dual. In many cases, we can solve the LSCAP (1.1) efficiently via the simplified dual problem (2.5), which has differentiable (but not twice differentiable) objective, a number of variables equal to the number of constraints in the original LSCAP, and nonnegativity constraints on some of the variables.

Several methods can be used to solve the general simplified dual problem. One very simple method is the projected (sub)gradient method, described in detail in section 3.1; another possibility is a conjugate gradient method that can handle nonnegativity constraints. Various special cases of the simplified dual can be handled by more specialized methods. For example, if there is only one equality constraint, the dual objective function can be shown to be twice differentiable, so Newton's method can be used. (This is described in more detail in section 3.4.3.)

If there are no inequality constraints in the original LSCAP (1.1), the simplified dual is an unconstrained problem, so any method for unconstrained minimization of a differentiable (but not twice differentiable) function can be used. For this case, Malick proposes a quasi-Newton method (BFGS) for the simplified dual problem [11]. Such a method will yield faster convergence than a simple gradient method for LSCAP problems with a few thousand equality constraints. For problems with more than a few thousand constraints, methods that do not require forming or storing a $p \times p$ matrix, such as conjugate gradients, might be preferable.

3.1. Dual projected gradient algorithm. In this section we describe in detail one simple method that can be used to solve the simplified dual problem (2.5). The projected (sub)gradient method repeats the following steps:

1. *Update X .* Set $X := (G - A(\nu) - C(\mu))_+$.
2. *Projected gradient update for μ and ν .*
 - (a) Evaluate primal residuals/dual gradients,

$$\frac{\partial \psi}{\partial \nu_i} = \mathbf{Tr} A_i X - b_i, \quad \frac{\partial \psi}{\partial \mu_j} = \mathbf{Tr} C_j X - d_j.$$

(b) Set

$$\nu_i := \nu_i + \alpha(\mathbf{Tr} A_i X - b_i), \quad \mu_j := (\mu_j + \alpha(\mathbf{Tr} C_j X - d_j))_+.$$

Here $\alpha > 0$ is a step size parameter. The iteration is stopped when all ν_i and μ_j , the equality constraint residuals and the positive parts of the inequality constraint residuals, are small enough.

Assuming the original LSCAP is strictly feasible, this algorithm is guaranteed to converge, i.e., ν , μ , and X converge to their optimal values, provided α is small enough. This follows from standard analysis of the subgradient algorithm, specialized to the case of differentiable objective; see, for example, [15, 1, 13, 10, 16].

We can give a specific bound on α that guarantees convergence. In the general case, convergence is guaranteed whenever $\alpha < 2/L$, where L is a Lipschitz constant for the gradient of the dual objective, i.e., the mapping from (ν, μ) to $(\partial\psi/\partial\nu, \partial\psi/\partial\mu)$. We can derive a Lipschitz constant for this mapping by noting that it is the composition of three mappings:

- The affine mapping from (ν, μ) to $G - A(\nu) - C(\mu)$.
- Projection onto the positive semidefinite cone, which yields X .
- The affine mapping from X to $(\partial\psi/\partial\nu, \partial\psi/\partial\mu)$.

The first affine mapping has a Lipschitz constant

$$\left(\sum_{i=1}^p \|A_i\|^2 + \sum_{j=1}^m \|C_j\|^2 \right)^{1/2},$$

where the norm is the spectral norm (i.e., maximum singular value). The projection has a Lipschitz constant one. The final affine mapping has a Lipschitz constant

$$\left(\sum_{i=1}^p \|A_i\|^2 + \sum_{j=1}^m \|C_j\|^2 \right)^{1/2}.$$

Thus, the gradient has a Lipschitz constant

$$L = \sum_{i=1}^p \|A_i\|^2 + \sum_{j=1}^m \|C_j\|^2.$$

While we are guaranteed that the algorithm converges for $\alpha < 2/L$, such a choice usually yields slow convergence.

We mention another step size strategy that guarantees convergence, and does not depend on the problem data at all. We choose the step size as a function of the iteration, i.e., we use step size α_k in iteration k . Convergence is guaranteed provided the step size sequence α_k satisfies the simple conditions

$$\alpha_k \geq 0, \quad \lim_{k \rightarrow \infty} \alpha_k = 0, \quad \sum_k \alpha_k = \infty.$$

For example, the algorithm is guaranteed to converge, for any problem data, with the universal choice $\alpha_k = 1/k$. (See, e.g., [15, 16] for more on step size strategies for projected subgradient methods.)

3.2. Complexity analysis. No matter what method is used to solve the simplified dual problem, each iteration requires evaluating the gradient of the dual objective, i.e., steps 1 and 2(a) of the projected gradient algorithm described above. For the projected gradient algorithm, this evaluation dominates the effort per step; for other methods, such as conjugate gradients or quasi-Newton methods, the cost of an iteration is at least the cost of evaluating the dual objective gradient. In this section and the next we analyze the cost of evaluating the dual objective gradient.

We first assume no structure in the problem data, i.e., G , A_i , C_j are given as full, high rank $n \times n$ matrices. We compute X by first forming $A(\nu)$ and $C(\mu)$, which costs order $n^2(p + m)$ flops, then computing the complete eigendecomposition of $G - A(\nu) - C(\mu)$, which is order n^3 flops, and finally forming X (which is order n^3 flops). The cost of step 2(a) is $n^2(p + m)$, so the overall cost of one dual objective gradient evaluation is therefore order $n^2 \max\{n, p + m\}$.

If there are few constraints (compared to n), the cost is dominated by the complete eigendecomposition of $G - A(\nu) - C(\mu)$, which has a cost around $10n^3$. One dual objective gradient evaluation (and therefore one iteration of the dual projected gradient algorithm) can be carried out on a current personal computer, for $n = 1000$, in at most a few seconds; assuming on the order of several hundred iterations, we can solve such an LSCAP in minutes. This dimension is far beyond the size of an LSCAP that can be solved using a standard interior-point method directly, since the problem has order 10^6 variables (entries in X).

3.3. Exploiting structure. In many interesting cases we can exploit structure in the LSCAP data to reduce the complexity of a dual objective gradient evaluation far below order n^3 . In this section we describe a generic example that illustrates the basic idea.

We assume that G is a positive definite matrix for which we can carry out matrix-vector multiplication efficiently, such as a sparse matrix, block diagonal matrix, or a diagonal matrix plus a matrix of low rank, given in factored form. We also assume that the data matrices A_i and C_j are all low rank and given in factored form (such as LDL^T). We let k denote the total of the ranks of the constraint data matrices, and we assume that $k \ll n$. Following (2.9) we see that for any ν and μ , we have $\text{Rank}(G - A(\nu) - C(\mu))_- \leq k$.

The matrix-vector multiplication $y \rightarrow (G - A(\nu) - C(\mu))y$ can be carried out efficiently, since Gy is efficiently computed, and $A_i y$ and $C_j y$ are efficiently computed using their factored forms. Therefore, we can use a Lanczos or subspace iteration method to compute the k eigenvalues $\lambda_{n-k-1}, \dots, \lambda_n$, as well as the associated eigenvectors, of $G - A(\nu) - C(\mu)$ (see, e.g., [12, 14]). This can be done very efficiently, since these methods require only that we multiply a given vector by $G - A(\nu) - C(\mu)$. Choosing the negative eigenvalues from these, and using the associated eigenvectors, we form $(G - A(\nu) - C(\mu))_-$, storing it in factored form. We then “form” $X = (G - A(\nu) - C(\mu))_+ = (G - A(\nu) - C(\mu)) + (G - A(\nu) - C(\mu))_-$, storing it as G (or really, the subroutine that computes Gy) along with the low rank, factored representation of $(G - A(\nu) - C(\mu))_-$.

To evaluate the gradient of the dual objective, we must compute $\text{Tr } A_i X$ and $\text{Tr } C_j X$. But these are readily computed using the factored form of A_i and C_j , and using the fact that $y \rightarrow Xy$ can be efficiently computed.

We note that this observation applies for *any* method for solving the simplified dual problem, and not just the dual projected gradient approach described above. In particular, it can be used with Malick’s BFGS method for the SDLS problem.

3.4. Special case: One rank one constraint. In this section we consider the special case with one rank one equality constraint (1.2),

$$\begin{aligned} & \text{minimize} && (1/2)\|X - G\|_F^2 \\ & \text{subject to} && X \succeq 0, \quad c^T X c = b, \end{aligned}$$

where $G \succeq 0$. Without loss of generality we can assume that $\|c\| = 1$. We assume that $b > 0$, so this problem is strictly feasible, which implies that the solution has the form

$$X^* = (G - \nu^* c c^T)_+,$$

where ν^* satisfies $c^T (G - \nu^* c c^T)_+ c = b$ (see (2.7)). The residual $c^T (G - \nu c c^T)_+ c - b$ is monotone nonincreasing in ν . We now find a lower and upper bound on ν^* .

We define $\underline{\nu}$ as the value of ν which satisfies $c^T (G - \nu c c^T) c = b$, i.e., $\underline{\nu} = c^T G c - b$. If $G - \underline{\nu} c c^T \succeq 0$, then $\nu^* = \underline{\nu}$. Otherwise,

$$c^T (G - \underline{\nu} c c^T)_+ c - b \geq c^T (G - \underline{\nu} c c^T) c - b = 0,$$

which implies that $\nu^* \geq \underline{\nu}$. Thus, $\underline{\nu}$ is a lower bound on the optimal solution ν^* .

Now we derive an upper bound $\bar{\nu}$ on ν^* . We first observe that

$$\psi(\nu) = -(1/2)\|(G - \nu c c^T)_+\|_F^2 + (1/2)\|G\|_F^2 - \nu b \leq (1/2)\|G\|_F^2 - \nu b.$$

Since the primal objective function is nonnegative, the optimal dual variable ν^* must have a nonnegative dual objective value. Using the inequality above, we have

$$0 \leq \psi(\nu^*) \leq (1/2)\|G\|_F^2 - \nu^* b.$$

From this we obtain

$$\nu^* \leq \bar{\nu} = \|G\|_F^2 / (2b).$$

3.4.1. Bisection. We can now find ν^* by bisection, starting from the initial bounds $\underline{\nu}$ and $\bar{\nu}$. At each iteration we must compute λ_n , the unique negative eigenvalue of $G - \nu c c^T$, and the associated unit eigenvector q , and then evaluate

$$\begin{aligned} c^T (G - \nu c c^T)_+ c - b &= c^T (G - \nu c c^T) c - \lambda_n (c^T q)^2 - b \\ &= c^T G c - b - \nu - \lambda_n (c^T q)^2 \\ &= \tilde{\nu} - \nu - \lambda_n (c^T q)^2. \end{aligned}$$

We have observed that in most cases, ν^* is much closer to the lower bound $\underline{\nu}$ than to the upper bound $\bar{\nu}$. This suggests carrying out the bisection using the geometric mean of the current bounds as the next iterate, instead of the usual arithmetic mean. This was suggested by Byers for another matrix nearness problem [4].

3.4.2. Exploiting structure. We can exploit structure in G to compute the initial bounds and carry out the bisection iterations efficiently. Suppose, for example, that G is large, but it is easy to compute the matrix-vector product Gy . Then we can compute λ_n by a power method, for example, with a shift to ensure the method converges to the negative eigenvalue. If we can efficiently compute $(G - \nu c c^T)^{-1}y$ (e.g., if G is sparse, and we are able to find a sufficiently sparse Cholesky factor), we can compute the eigenvalue λ_n and eigenvector q using inverse iteration with shifts, which requires only a handful of iterations.

The upper bound requires computing $\|G\|_F^2$, which can also be done efficiently in many cases, by exploiting structure in G . This is clear when G is sparse, since $\|G\|_F^2$ is just the sum of the squares of its nonzero entries. As a more interesting example,

suppose G is a factor matrix, given as $D_1 + LD_2L^T$, where L has $k \ll n$ columns and D_1 and D_2 are diagonal. Even though G is full, its Frobenius norm can be efficiently computed using the formula

$$\begin{aligned}\|G\|_F^2 &= \|D_1\|_F^2 + 2 \operatorname{Tr} D_1 LD_2 L^T + \operatorname{Tr} LD_2 L^T LD_2 L^T \\ &= \|D_1\|_F^2 + 2 \operatorname{Tr} (LD_2)^T (D_1 L) + \operatorname{Tr} (D_2 L^T L) (D_2 L^T L) \\ &= \|D_1\|_F^2 + 2 \operatorname{Tr} (LD_2)^T (D_1 L) + \|D_2 L^T L\|_F^2.\end{aligned}$$

The right-hand side can be calculated in order $k^2 n$ flops (as compared to kn^2 , if we form the matrix G and then the sum of the squares of its entries).

3.4.3. Newton's method. For this special case, the residual $r(\nu) = c^T(G - \nu cc^T)_+ c - b$ is differentiable (in fact, analytic), so we can use Newton's method to find its root ν^* . (This is the same as using Newton's method to maximize the dual objective function, since r is the derivative of the dual objective.) The derivative of the residual (i.e., the second derivative of the dual objective) is given by

$$\dot{r} = -1 - \dot{\lambda}_n (c^T q)^2 - 2\lambda_n (c^T q)(c^T \dot{q}),$$

where we use the superscript dot to denote the derivative with respect to ν .

Now we recall some results from standard perturbation theory (see, e.g., [9]). Suppose the matrix A is symmetric, depends on a scalar parameter ν , and has isolated eigenvalue λ with associated unit eigenvector q . Then we have

$$\dot{\lambda} = q^T \dot{A} q, \quad \dot{q} = -(A - \lambda I)^\dagger \dot{A} q,$$

where $U^\dagger$ denotes the Moore–Penrose pseudoinverse. Applying these results here, we have

$$\dot{\lambda}_n = -(c^T q)^2, \quad \dot{q} = (c^T q)(G - \nu cc^T - \lambda_n I)^\dagger c.$$

Substituting these expressions into the equation above, we obtain

$$\dot{r} = -1 + (c^T q)^4 - 2\lambda_n (c^T q)^2 c^T (G - \nu cc^T - \lambda_n I)^\dagger c.$$

Now we describe a simple guarded Newton method, which maintains an interval $[l, u]$ known to contain ν^* at each step, and whose width decreases by at least a fixed fraction each step (thus guaranteeing convergence). For an interval $[l, u]$ and $\alpha > 0$, we let $\alpha[l, u]$ denote the interval scaled by α about its center, i.e.,

$$\alpha[l, u] = [(u+l)/2 - \alpha(u-l)/2, (u+l)/2 + \alpha(u-l)/2].$$

The guarded Newton method starts with the initial interval $[l, u] = [\underline{\nu}, \bar{\nu}]$, an initial value ν inside the interval (such as $\nu = (u+l)/2$), and a guard parameter value α that satisfies $0 \leq \alpha < 1$. The following steps are then repeated, until $r(\nu)$ is small enough:

1. *Calculate Newton update.* Find the (pure) Newton update $\nu_{\text{nt}} = \nu - r(\nu)/\dot{r}(\nu)$.
2. *Project onto guard interval.* Set ν to be the projection of ν_{nt} onto the interval $\alpha[l, u]$.
3. *Update guard interval.* If $r(\nu) > 0$, set $l = \nu$; otherwise set $u = \nu$.

When $\alpha = 0$ this guarded Newton algorithm reduces to bisection. For $\alpha > 0$, the algorithm has quadratic terminal convergence. As a result, it converges to high accuracy very quickly; on the other hand, it requires computing $(G - \lambda_n cc^T)^\dagger c$ at each iteration. If no structure is exploited, this does not increase the order of the

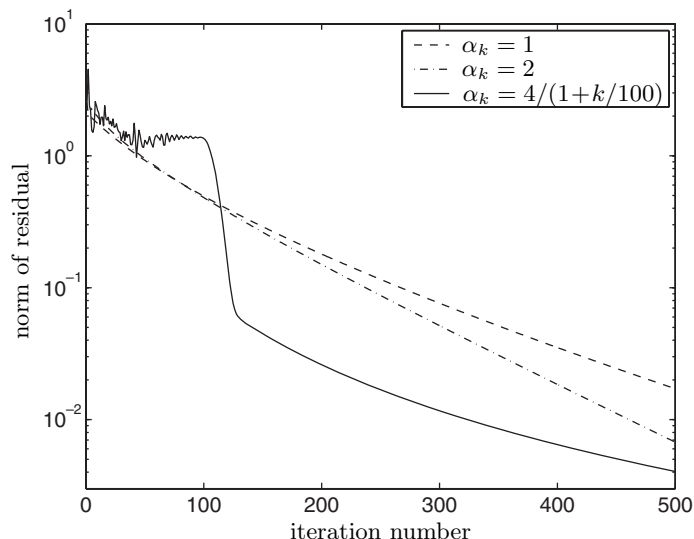


FIG. 1. Norm of residual for the large example.

computational cost (which is still n^3). But this method makes it difficult or impossible to exploit structure such as sparsity in G .

4. Examples. Our first example is a large problem. The matrix G is a sparse positive semidefinite matrix with dimension $n = 10^4$ and around 10^6 nonzero entries. It was generated by choosing a random sparse matrix Z , with unit Gaussian nonzero entries, and setting $G = Z^T Z$. (The density of Z was chosen so that G has around 10^6 nonzero entries.) The problem has $p = 50$ equality constraints and $m = 50$ inequality constraints. The coefficient matrices A_i and C_j are all rank one, $A_i = a_i a_i^T$, $C_j = c_j c_j^T$ with a_i and c_j chosen randomly and normalized. The coefficients b_i and d_j were chosen as $b_i = \eta_i a_i^T G a_i$, where η_i are random with uniform distribution on $[0.5, 1.5]$.

The method described in section 3.1 was used with constant step $\alpha = 1$, constant step size $\alpha = 2$, and diminishing step size rule with $\alpha_k = 4/(1 + k/100)$. The convergence of the norm of the residual,

$$\left(\sum_{i=1}^p (\text{Tr } A_i X - b_i)^2 + \sum_{j=1}^m (\max \{0, \text{Tr } C_j X - d_j\})^2 \right)^{1/2},$$

is shown in Figure 1, and the convergence of the dual objective is given in Figure 2. Convergence to modest accuracy occurs in about 400 iterations.

Our second example has the special form (1.2), with G a dense randomly chosen positive semidefinite 100×100 matrix, and the vector c randomly generated and normalized. We solved this problem using the bisection method, using both the arithmetic and geometric means, and the guarded Newton method with parameter $\alpha = 0.9$. The convergence of the residual magnitude is shown in Figure 3. As expected, the Newton method converges to very high accuracy in fewer steps.

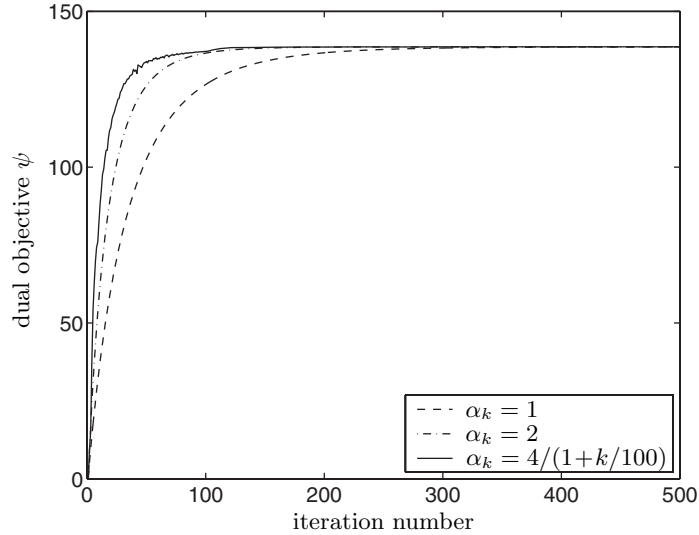


FIG. 2. Dual objective for the large example.

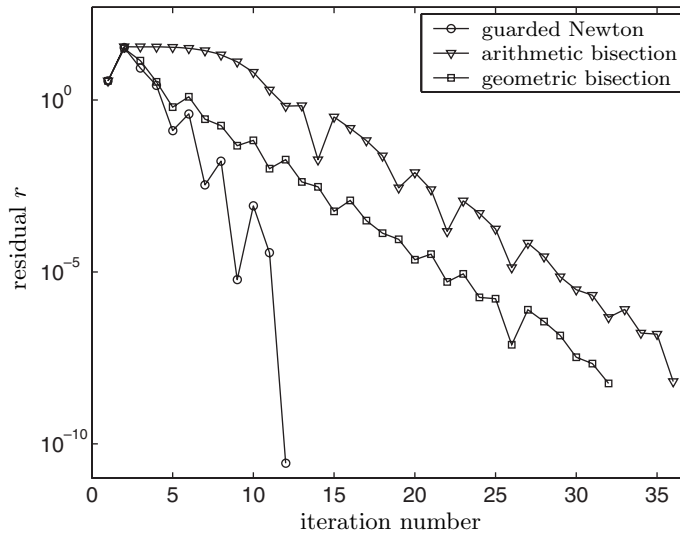


FIG. 3. Residual magnitude for the small example with one rank-one equality constraint.

Acknowledgments. We are grateful to Andrew Ng for suggesting the problem to us and to two anonymous reviewers for very useful suggestions and comments.

REFERENCES

- [1] D. BERTSEKAS, *Nonlinear Programming*, 2nd ed., Athena Scientific, Belmont, MA, 1999.
- [2] J. BORWEIN AND A. LEWIS, *Convex Analysis and Nonlinear Optimization*, Canad. Math. Soc. Books Math., Springer-Verlag, New York, 2000.
- [3] S. BOYD AND L. VANDENBERGHE, *Convex Optimization*, Cambridge University Press, Cambridge, UK, 2004. Available online from www.stanford.edu/~boyd/cvxbook.html.

- [4] R. BYERS, *A bisection method for measuring the distance of a stable matrix to the unstable matrices*, SIAM J. Sci. Statist. Comput., 9 (1988), pp. 875–881.
- [5] N. HIGHAM, *Computing a nearest symmetric positive semidefinite matrix*, Linear Algebra Appl., 103 (1988), pp. 103–118.
- [6] N. HIGHAM, *Matrix nearness problems and applications*, in Applications of Matrix Theory, M. Gover and S. Barnett, eds., Oxford University Press, Oxford, 1989, pp. 1–27.
- [7] N. HIGHAM, *Computing the nearest correlation matrix—A problem from finance*, IMA J. Numer. Anal., 22 (2002), pp. 329–343.
- [8] J.-B. HIRIART-URRUTY AND C. LEMARÉCHAL, *Fundamentals of Convex Analysis*, Springer-Verlag, New York, 2001.
- [9] T. KATO, *A Short Introduction to Perturbation Theory for Linear Operators*, Springer-Verlag, New York, 1982.
- [10] E. LEVITIN AND B. POLYAK, *Constrained minimization methods*, USSR Comput. Math. Math. Phys., 6 (1966), pp. 1–50.
- [11] J. MALICK, *A dual approach to semidefinite least-squares problems*, SIAM J. Matrix Anal. Appl., 26 (2005), pp. 272–284.
- [12] B. PARLETT, *The Symmetric Eigenvalue Problem*, Prentice-Hall, Englewood Cliffs, NJ, 1980.
- [13] B. POLYAK, *Introduction to Optimization*, Optimization Software, New York, 1987.
- [14] Y. SAAD, *Numerical Methods for Large Eigenvalue Problems*, Manchester University Press, Manchester, UK, 1992.
- [15] N. SHOR, *Minimization Methods for Non-Differentiable Functions*, Springer Ser. Comput. Math., Springer-Verlag, New York, 1985.
- [16] N. SHOR, *Nondifferentiable Optimization and Polynomial Problems*, Kluwer, Norwell, MA, 1998.
- [17] M. TODD, *Semidefinite optimization*, Acta Numer., 10 (2001), pp. 515–560.
- [18] L. VANDENBERGHE AND S. BOYD, *Semidefinite programming*, SIAM Rev., 38 (1996), pp. 49–95.

TAILS OF CONDITION NUMBER DISTRIBUTIONS*

ALAN EDELMAN[†] AND BRIAN D. SUTTON[†]

Abstract. Let κ be the condition number of an m -by- n matrix with independent standard Gaussian entries, either real ($\beta = 1$) or complex ($\beta = 2$). The major result is the existence of a constant C (depending on m , n , and β) such that $P[\kappa > x] < C x^{-\beta}$ for all x . As $x \rightarrow \infty$, the bound is asymptotically tight. An analytic expression is given for the constant C , and simple estimates are given, one involving a Tracy–Widom largest eigenvalue distribution. All of the results extend beyond real and complex entries to general β .

Key words. random matrix, condition number, Wishart

AMS subject classifications. 15A12, 15A52

DOI. 10.1137/040614256

1. Introduction. Take an m -by- n matrix G with iid standard Gaussian entries. What is the distribution of its condition number? The *condition number* κ is defined as the ratio of the largest to smallest singular values,

$$\kappa = \frac{\sigma_m(G)}{\sigma_1(G)} = \sqrt{\frac{\lambda_m(W)}{\lambda_1(W)}},$$

or, equivalently, as the square root of the ratio of the extreme eigenvalues of the *Wishart matrix* $W = GG^*$. This paper derives an upper bound on the tail $P[\kappa > x]$ for all x . As $x \rightarrow \infty$, the bound is tight. The results apply to the following cases:

- square or rectangular;
- finite or infinite;
- real or complex, as well as extensions to “general β .”

Wishart matrices arise commonly in the statistics literature, for example in principal component analysis. The standard reference is [8]. Traditionally, a Wishart matrix of parameters m , n , and β is defined as GG^* , where G is a matrix of iid standard Gaussian entries. G is m -by- n and has either real or complex Gaussian entries, depending on the value of β , either 1 or 2. (This paper assumes an identity covariance matrix.) The distributions of the largest and smallest eigenvalues of a Wishart matrix, expressed in terms of zonal polynomials and hypergeometric functions, can be found in [8].

The class of Wishart matrices has been expanded beyond real and complex entries [3, 4]. The *generalized Wishart matrix* $W_{m,n}^\beta$ is defined for nonintegral n and any $\beta > 0$. The generalized Wishart matrices are not always expressible as $W = GG^*$, so the condition number is defined in the general case to be

$$\kappa = \sqrt{\lambda_m/\lambda_1}.$$

Interest in random condition numbers first arose in the numerical linear algebra community for their use in measuring the sensitivity of problems to perturbations.

*Received by the editors August 31, 2004; accepted for publication (in revised form) by N. J. Higham April 11, 2005; published electronically November 22, 2005.

<http://www.siam.org/journals/simax/27-2/61425.html>

[†]Department of Mathematics, Massachusetts Institute of Technology, Cambridge, MA 02139-4307 (edelman@math.mit.edu, bsutton@math.mit.edu). The research of the second author was supported by a National Science Foundation Graduate Research Fellowship.

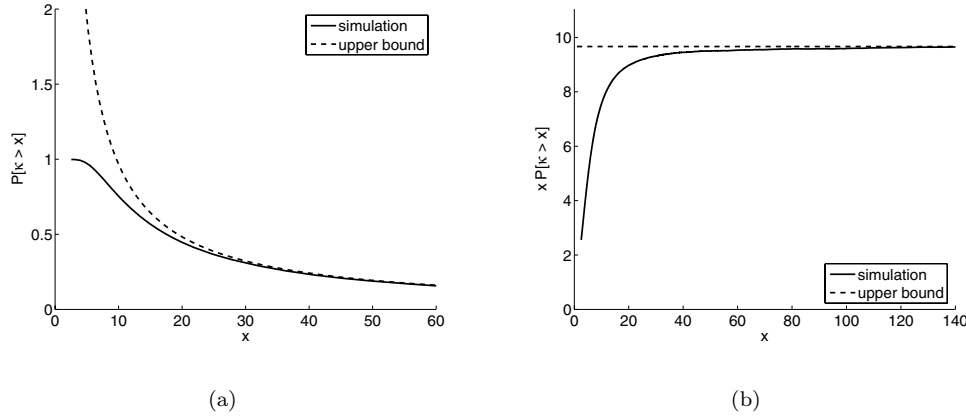


FIG. 1.1. The condition number of a 6-by-6 matrix of independent real Gaussians. (a) The cdf $P[\kappa > x]$ and an asymptotically tight upper bound. (b) The same cdf and upper bound after rescaling (multiplying by x).

References include [1, 2, 11, 13]. The limiting distribution of the condition number, as the size of the matrix m approaches infinity, can be found in [5]. This paper considers finite m as well as large m limits.

The large m asymptotics take advantage of the recent observation that the largest eigenvalue of a Wishart matrix approaches a Tracy–Widom distribution as $m \rightarrow \infty$ [6]. This is another example of the convergence of the approaches to Wishart matrices and Gaussian ensembles.

Figure 1.1 plots the cdf of κ , taken from $W_{6,6}^1$, before and after rescaling. By cdf, we mean $P[\kappa > x]$, rather than the more common $P[\kappa < x]$. In this particular case, the tail of the cdf is asymptotic to x^{-1} , so multiplying by x produces a nonzero limiting value at infinity, shown in the second plot. The dashed curve is an upper bound that is asymptotically tight, the main focus of this paper.

In the special case $W_{m,m}^1$, the main theorem is as follows.

THEOREM 1.1. *Let κ be the condition number of an m -by- m matrix of iid real standard Gaussian entries. Then*

$$P[\kappa > x] \sim \frac{\sqrt{2}\Gamma(\frac{1}{2}(m+1))}{\Gamma(\frac{1}{2}m)}\mu x^{-1} \quad (x \rightarrow \infty),$$

and the right-hand side is an upper bound for all values of x . The constant μ is

$$\mu = E[\sigma_{\max}(G_{m-1,m+1})] = E[\lambda_{\max}(W_{m-1,m+1}^1)^{1/2}],$$

the mean of the largest singular value of an $(m-1)$ -by- $(m+1)$ matrix of independent real standard Gaussian entries.

The constant

$$\lim_{x \rightarrow \infty} xP[\kappa > x] = \frac{\sqrt{2}\Gamma(\frac{1}{2}(m+1))}{\Gamma(\frac{1}{2}m)}\mu$$

is expressed analytically in terms of a multivariate hypergeometric function. Because multivariate hypergeometrics are difficult to compute, we provide estimates based

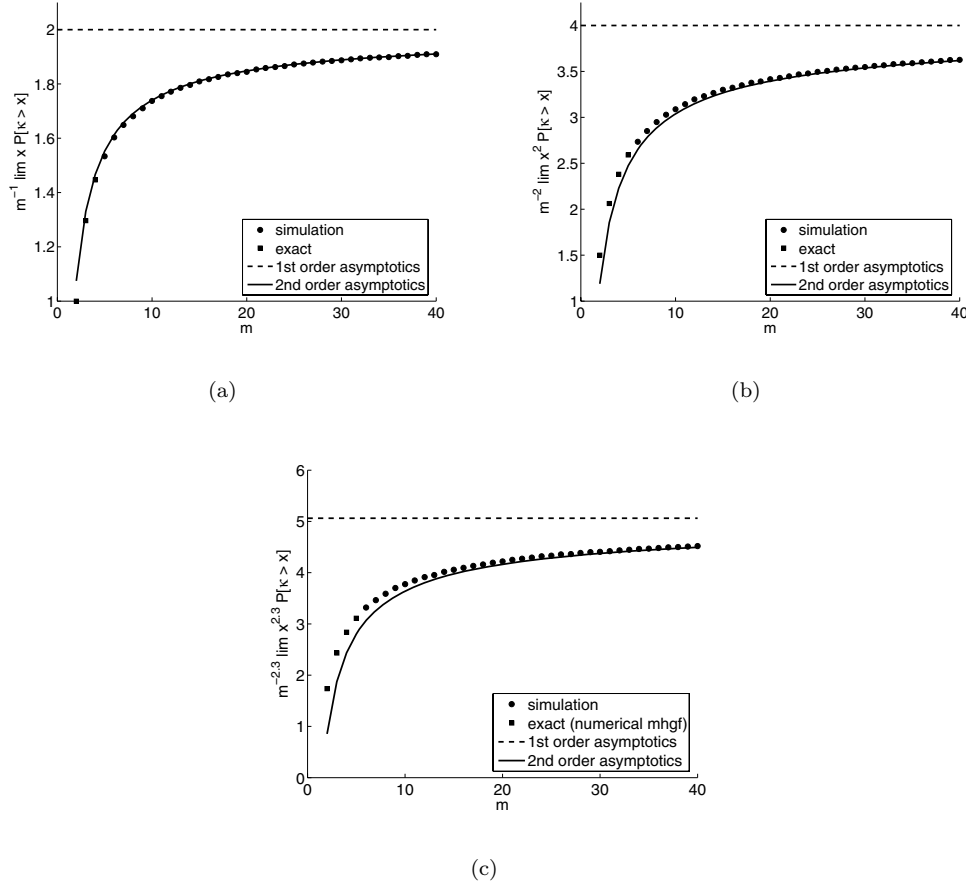


FIG. 1.2. Dependence on the size of the matrix m . The large x limiting value of $(\frac{x}{m})^\beta P[\kappa > x]$ is plotted against m . The asymptotics are developed in section 4. (a) Real case ($\beta = 1, m = n$). (b) Complex case ($\beta = 2, m = n$). (c) Neither real nor complex ($\beta = 2.3, m = n$).

on large m asymptotics. As can be seen in Figure 1.2(a), a combination of exact computation and large m asymptotics produces an estimate for every value of m that agrees well with simulations in the real square case. Figure 1.2(b) and 1.2(c) show similar estimates for $\beta = 2$ and $\beta = 2.3$, respectively. The asymptotic analysis, based on the Tracy–Widom largest eigenvalue distributions, is heuristic and is supported numerically by Figure 1.2.

Our results can be compared with the literature, specifically [1]. First, this paper enjoys greater generality; the cited paper restricts itself to real square matrices. In this restricted case, the reference shows that $\frac{x}{m} P[\kappa > x] < 5.60$. As evident from Figure 1.2(a), the value 5.60 can likely be replaced by 2, and even smaller numbers for finite m . This paper provides, for each m , n , and β , an exact expression for the constant, not just a universal upper bound.

Section 2 describes generalized Wishart matrices in more detail. Section 3 presents the main theorem in full generality for all values of β . Section 4 provides estimates on the constant in the asymptotic relation. Section 5 considers asymptotics of κ near 1. Section 6 derives the exact cdf for the condition number of 2-by-2 matrices.

2. Generalized Wishart matrices. Traditionally, a real Wishart matrix is defined as the symmetrization GG^T of an m -by- n matrix G of independent Gaussian entries (mean 0, variance 1). The m -by- m Wishart matrix is said to have n degrees of freedom, $n > m - 1$. Letting $\lambda_1 \leq \dots \leq \lambda_m$ be the eigenvalues of GG^T , it follows that the condition number of G is $\kappa = \sqrt{\lambda_m/\lambda_1}$.

The traditional complex Wishart matrix is defined as GG^* . Here, G has complex Gaussian entries with real and imaginary parts having mean 0 and variance 1.

To overcome the limitations of integral n and $\beta = 1, 2$, a tridiagonal Wishart matrix has been developed [3, 4] in the spirit of [10]. The tridiagonal model has three parameters— m , n , and β —and is denoted by $W_{m,n}^\beta$. It is a real m -by- m matrix distributed as follows:

$$W_{m,n}^\beta = BB^T,$$

$$B \sim \begin{bmatrix} \chi_{\beta n} & & & & \\ \chi_{\beta(m-1)} & \chi_{\beta(n-1)} & & & \\ & \ddots & \ddots & & \\ & & \chi_\beta & \chi_{\beta(n-m+1)} & \end{bmatrix},$$

$$m \in \mathbb{N}, \quad n \in \mathbb{R}, \quad \beta \in \mathbb{R}, \quad n > m - 1, \quad \beta > 0.$$

The entries of B are independent, and χ_r denotes a χ -distributed random variable with r degrees of freedom. Remember that n need not be an integer. However, a special case occurs when n is an integer and $\beta = 1$ (respectively, $\beta = 2$). Then $W_{m,n}^\beta$ has the same eigenvalue distribution as the traditional real (complex) Wishart matrix GG^T (GG^*), in which G is m -by- n .

The joint density of the eigenvalues of a Wishart matrix is well known and can be found in [4, 8]. The constant term involves the multivariate Gamma function

$$\Gamma_m^\beta(z) = \pi^{\beta m(m-1)/4} \prod_{i=1}^m \Gamma\left(z + \frac{\beta}{2}(i-1)\right).$$

PROPOSITION 2.1. *Let $\lambda_1 \leq \dots \leq \lambda_m$ be the eigenvalues of the m -by- m Wishart matrix with parameters n and β . The joint density for these eigenvalues is*

$$(2.1) \quad m! c_{m,n}^\beta \prod_{j < i} |\lambda_i - \lambda_j|^\beta \prod_i \lambda_i^{p-1} e^{-\sum_i \lambda_i/2}$$

with $p = \frac{\beta}{2}(n - m + 1)$ and

$$(2.2) \quad c_{m,n}^\beta = \frac{\pi^{\beta m(m-1)/2}}{2^{\beta mn/2}} \frac{\Gamma(1 + \frac{\beta}{2})^m}{\Gamma_m^\beta(1 + \frac{\beta}{2}) \Gamma_m^\beta(p)}.$$

Because the quantity $\frac{\beta}{2}(n - m + 1)$ arises so often, the symbol p will be used throughout,

$$p = \frac{\beta}{2}(n - m + 1).$$

It is important to remember that p depends on m , n , and β .

3. Asymptotics at $\kappa = \infty$. We begin by bounding the joint density of the extreme eigenvalues of a Wishart matrix. Loosely speaking, when λ_m/λ_1 is known to be large, the smallest eigenvalue is essentially independent from the other eigenvalues, following a gamma distribution. The remaining eigenvalues appear to be taken from a smaller Wishart matrix.

More precisely this can be expressed as follows.

LEMMA 3.1. *Let $f(\lambda_1, \lambda_m)$ be the joint density of the extreme eigenvalues of $W_{m,n}^\beta$. The density is bounded above by*

$$f(\lambda_1, \lambda_m) \leq pC\lambda_1^{p-1}e^{-\lambda_1/2}g_{m-1,n+1}^\beta(\lambda_m),$$

where $g_{m-1,n+1}^\beta$ is the density of the largest eigenvalue of $W_{m-1,n+1}^\beta$.

Also, for fixed λ_m ,

$$f(\lambda_1, \lambda_m) \sim pC\lambda_1^{p-1}g_{m-1,n+1}^\beta(\lambda_m) \quad (\lambda_1 \rightarrow 0).$$

The constant C is

$$C = \frac{m}{p} \frac{c_{m,n}^\beta}{c_{m-1,n+1}^\beta},$$

where $c_{m,n}^\beta$ is defined in (2.2).

Proof. The factors of $f(\lambda_1, \lambda_m)$ may be grouped conveniently as

$$\begin{aligned} f(\lambda_1, \lambda_m) &= pC[\lambda_1^{p-1}e^{-\lambda_1/2}] \\ &\quad \times (m-1)!c_{m-1,n+1}^\beta \int \cdots \int \prod_{2 \leq j < i \leq m} |\lambda_i - \lambda_j|^\beta \prod_{i=2}^m [|\lambda_i - \lambda_1|^\beta \lambda_i^{p-1}] \\ &\quad \times e^{-\sum_{i=2}^m \lambda_i/2} d\lambda_2 \cdots d\lambda_{m-1}, \end{aligned}$$

where the integration is taken over the region $\lambda_1 \leq \lambda_2 \leq \cdots \leq \lambda_{m-1} \leq \lambda_m$.

To obtain the upper bound, use the estimate $|\lambda_i - \lambda_1|^\beta \leq \lambda_i^\beta$ and expand the region of integration; compare with (2.1).

For the asymptotic result, we need to show that

$$\begin{aligned} f(\lambda_1, \lambda_m) &= pC[\lambda_1^{p-1}e^{-\lambda_1/2}] \\ &\quad \times (m-1)!c_{m-1,n+1}^\beta \int \cdots \int \prod_{2 \leq j < i \leq m} |\lambda_i - \lambda_j|^\beta \prod_{i=2}^m [|\lambda_i - \lambda_1|^\beta \lambda_i^{p-1}] \\ &\quad \times e^{-\sum_{i=2}^m \lambda_i/2} d\lambda_2 \cdots d\lambda_{m-1} \end{aligned}$$

approaches $g_{m-1,n+1}^\beta(\lambda_m)$, since this quantity is a nonzero constant. To see this, use the dominated convergence theorem. As $\lambda_1 \rightarrow 0$, the integrand converges upwards to the joint density of the eigenvalues of $W_{m-1,n+1}^\beta$. All eigenvalues except for the largest are integrated out, leaving the marginal density of the largest eigenvalue. $\square$

Now we can present our main theorem. One of the constants involved will be a moment of the largest eigenvalue of a Wishart matrix. Specifically, μ will denote the expected value of $\lambda_{\max}^p$, in which $\lambda_{\max}$ is the largest eigenvalue of a Wishart matrix of parameters $m-1$, $n+1$, and β :

$$(3.1) \quad \mu = E[\lambda_{\max}(W_{m-1,n+1}^\beta)^p].$$

(To clarify this expression, $p = \frac{\beta}{2}(n - m + 1)$, not $\frac{\beta}{2}((n + 1) - (m - 1) + 1)$.) μ can be expressed as an integral involving a hypergeometric function; see section 4.

THEOREM 3.2. *Let κ be the condition number associated with a Wishart matrix of parameters m , n , and β . Then, for $x \geq 1$,*

$$P[\kappa > x] \leq C\mu x^{-2p}.$$

As x tends to infinity, the right-hand side gives the first order asymptotics for the tail of the condition number,

$$P[\kappa > x] \sim C\mu x^{-2p} \quad (x \rightarrow \infty).$$

C is the same constant from Lemma 3.1, and depends on m , n , and β . μ is defined in (3.1).

Proof. Denoting the joint density of λ_1, λ_m by $f(\lambda_1, \lambda_m)$ as in the lemma, we have

$$\begin{aligned} \int_0^{x^{-2}\lambda_m} f(\lambda_1, \lambda_m) d\lambda_1 &\leq pCg_{m-1,n+1}^\beta(\lambda_m) \int_0^{x^{-2}\lambda_m} \lambda_1^{p-1} d\lambda_1 \\ &= C\lambda_m^p g_{m-1,n+1}^\beta(\lambda_m) x^{-2p}. \end{aligned}$$

Therefore, the upper bound is immediate:

$$\begin{aligned} P[\kappa > x] &= \int_0^\infty \int_0^{x^{-2}\lambda_m} f(\lambda_1, \lambda_m) d\lambda_1 d\lambda_m \\ &\leq Cx^{-2p} \int_0^\infty \lambda_m^p g_{m-1,n+1}^\beta(\lambda_m) d\lambda_m \\ &= C\mu x^{-2p}. \end{aligned}$$

For the asymptotics, we need to compute the limit of

$$\frac{1}{C} \frac{1}{\mu} x^{2p} P[\kappa > x] = \frac{1}{\mu} \int_0^\infty \left[\frac{1}{C} x^{2p} \int_0^{x^{-2}\lambda_m} f(\lambda_1, \lambda_m) d\lambda_1 \right] d\lambda_m$$

as $x \rightarrow \infty$. By the second part of the lemma, the integrand w.r.t. $d\lambda_m$ is dominated by and converges pointwise to

$$\lambda_m^p g_{m-1,n+1}^\beta(\lambda_m)$$

as $x \rightarrow \infty$. By the dominated convergence theorem, the entire integral converges to μ , so that

$$\frac{1}{C} \frac{1}{\mu} x^{2p} P[\kappa > x] \rightarrow 1,$$

as desired. $\square$

The following corollary applies to square matrices of real or complex Gaussians, in particular.

COROLLARY 3.3. *If $m = n$, then*

$$P[\kappa > x] \sim \frac{1}{2^{\beta/2-1}\Gamma(1+\beta)} \frac{\Gamma(\frac{\beta}{2}(m+1))}{\Gamma(\frac{\beta}{2}m)} \mu x^{-\beta} \quad (x \rightarrow \infty).$$

From numerical evidence, e.g., Figure 1.1(b), the rescaled tail appears to be monotonic, but a proof remains an open question.

CONJECTURE 3.4. *For any m, n , and β , the rescaled tail $x^{2p} P[\kappa > x]$ is an increasing function of x .*

The following theorem provides a lower bound on the tail for some values of m , n , and β . A lower bound of a different flavor which applies to the real square case ($\beta = 1$, $m = n$) is given in [1].

THEOREM 3.5. *When $p \geq 1$, the condition number is bounded below, for all $x \geq 1$, by*

$$P[\kappa > x] \geq C\mu x^{-2p} - \frac{m}{2} \frac{pC}{p+1} \nu x^{-2p-2},$$

where C , p , and μ are defined as before, and

$$\nu = E[\lambda_{\max}(W_{m-1, n+1}^\beta)^{p+2}].$$

Proof. Perform the change of variables $\rho_i = \lambda_{i+1} - \lambda_1$, $i = 1, \dots, m-1$, and use the estimate $(\rho_i + \lambda_1)^{p-1} \geq \rho_i^{p-1}$ to find the following bound on the joint density of the extreme eigenvalues:

$$\begin{aligned} f(\lambda_1, \lambda_m) &= pC [\lambda_1^{p-1} e^{-m\lambda_1/2}] (m-1)! c_{m-1, n+1}^\beta \int \cdots \int \prod_{1 \leq j < i \leq m-1} |\rho_i - \rho_j|^\beta \\ &\quad \times \prod_{i=1}^{m-1} [\rho_i^\beta (\rho_i + \lambda_1)^{p-1}] e^{-\sum_{i=1}^{m-1} \rho_i/2} d\rho_1 \cdots d\rho_{m-2} \\ &\geq pC [\lambda_1^{p-1} e^{-m\lambda_1/2}] g_{m-1, n+1}^\beta(\rho_{m-1}) \\ &= pC [\lambda_1^{p-1} e^{-m\lambda_1/2}] g_{m-1, n+1}^\beta(\lambda_m - \lambda_1), \end{aligned}$$

where the integral is taken over the region $0 \leq \rho_1 \leq \cdots \leq \rho_{m-1} = \lambda_m - \lambda_1$. Then the tail is bounded by

$$\begin{aligned} P[\kappa > x] &= \int_0^\infty \int_0^{x^{-2}\lambda_m} f(\lambda_1, \lambda_m) d\lambda_1 d\lambda_m \\ &\geq pC \int_0^\infty \left(\int_0^{x^{-2}\lambda_m} \lambda_1^{p-1} e^{-m\lambda_1/2} d\lambda_1 \right) g_{m-1, n+1}^\beta(\lambda_m - \lambda_1) d\lambda_m. \end{aligned}$$

Applying the change of variables $s = \lambda_1$, $t = \lambda_m - \lambda_1$, and estimating $e^{-m\lambda_1/2} \geq 1 - \frac{m}{2}\lambda_1$, the lower bound becomes

$$\begin{aligned} P[\kappa > x] &\geq pC \int_0^\infty \left[\int_0^{x^{-2}(1-x^{-2})^{-1}t} s^{p-1} \left(1 - \frac{m}{2}s \right) ds \right] g_{m-1, n+1}^\beta(t) dt \\ &\geq pC \int_0^\infty \left[\int_0^{x^{-2}t} s^{p-1} \left(1 - \frac{m}{2}s \right) ds \right] g_{m-1, n+1}^\beta(t) dt \\ &= C\mu x^{-2p} - \frac{m}{2} \frac{pC}{p+1} \nu x^{-2p-2}. \quad \square \end{aligned}$$

4. Estimates for the constant. Recall the main theorem,

$$P[\kappa > x] \sim C\mu x^{-2p} \quad (x \rightarrow \infty).$$

In this section, we give an exact formula and heuristic estimates for the constant $C\mu$. Then we compare the estimates with numerical data in the cases $\beta = 1, 2, 2.3$. The value 2.3 was chosen arbitrarily in order to illustrate the approach when $\beta \neq 1, 2$.

The estimates are based on large m asymptotics. They will be restricted to the “square” case, $m = n$, for ease of presentation. The arguments work just as well for $m \neq n$.

4.1. Exact value. The constant C was specified earlier, and

$$\mu = E[\lambda_{\max}(W_{m-1,n+1}^\beta)^p].$$

The cdf for $\lambda_{\max}(W_{m-1,n+1}^\beta)$ is

$$\frac{\Gamma_{m-1}^\beta\left(\frac{\beta}{2}(m-2)+1\right)}{\Gamma_{m-1}^\beta\left(\frac{\beta}{2}(n+m-1)+1\right)} \times \left(\frac{x}{2}\right)^{(m-1)(n+1)\beta/2} {}_1F_1^\beta\left(\frac{\beta}{2}(n+1), \frac{\beta}{2}(n+m-1), -\frac{x}{2}I_{m-1}\right),$$

where Γ_{m-1}^β is a multivariate gamma function and ${}_1F_1^\beta$ is a multivariate hypergeometric function [3]. Therefore, the p th moment is an integral over the corresponding measure.

4.2. First order asymptotics. First, we will develop a crude estimate, which will motivate a more refined estimate. Start by applying Stirling’s formula to Corollary 3.3. If $m = n$, then

$$\begin{aligned} C &= \frac{1}{2^{\beta/2-1}\Gamma(1+\beta)} \frac{\Gamma\left(\frac{\beta}{2}(m+1)\right)}{\Gamma\left(\frac{\beta}{2}m\right)} \\ &\sim \frac{\beta^{\beta/2-1}}{2^{\beta-1}\Gamma(\beta)} m^{\beta/2} \quad (m \rightarrow \infty). \end{aligned}$$

We shall argue heuristically in a moment that the mean of the largest eigenvalue of $W_{m-1,m+1}^\beta$ is asymptotic to $4\beta m$, and the standard deviation is dominated by the mean. Hence, an immediate estimate for μ is

$$\begin{aligned} \mu &= E[\lambda_{\max}(W_{m-1,m+1}^\beta)^{\beta/2}] \\ &\sim 2^\beta \beta^{\beta/2} m^{\beta/2} \quad (m \rightarrow \infty) \end{aligned}$$

giving

$$(4.1) \quad \lim_{x \rightarrow \infty} x^\beta P[\kappa > x] = C\mu \sim \frac{2\beta^{\beta-1}}{\Gamma(\beta)} m^\beta \quad (m \rightarrow \infty).$$

The estimate (4.1) is plotted for $\beta = 1, 2, 2.3$ in Figure 1.2. For $\beta = 1$, the estimate is the horizontal line at 2. For $\beta = 2$, it is the horizontal line at 4. For $\beta = 2.3$, it is the horizontal line at ≈ 5.1 . It is clear that this asymptotic estimate is only valid for very large matrices.

As an extra sanity check, this estimate can be compared with the large m limiting distribution of the condition number when $\beta = 1, 2$.

PROPOSITION 4.1 (see [5]). *Let κ be the condition number associated with $W_{m,m}^1$. Then*

$$\lim_{m \rightarrow \infty} P[\kappa/m > x] = 1 - e^{-2/x - 2/x^2}.$$

Let $\tilde{\kappa}$ be the condition number associated with $W_{m,m}^2$. Then

$$\lim_{m \rightarrow \infty} P[\tilde{\kappa}/m > x] = 1 - e^{-4/x^2}.$$

Taking series expansions,

$$\lim_{m \rightarrow \infty} \frac{1}{m} P[\kappa > x] \sim 2/x \quad (x \rightarrow \infty)$$

and

$$\lim_{m \rightarrow \infty} \frac{1}{m^2} P[\tilde{\kappa} > x] \sim 4/x^2 \quad (x \rightarrow \infty).$$

In these two cases, at least, the limits $x \rightarrow \infty$ and $m \rightarrow \infty$ can be interchanged.

4.3. The Tracy–Widom distributions. For a more refined estimate, the result of Johnstone on the largest eigenvalue of a Wishart matrix will be key. It states that as $m \rightarrow \infty$, the largest eigenvalue of an m -by- m Wishart matrix, appropriately scaled, converges in distribution to a Tracy–Widom largest eigenvalue distribution.

The Tracy–Widom distributions were originally discovered as the limiting distributions of the largest eigenvalues of Gaussian ensembles. Their cdf's are expressed in terms of a Painlevé transcendent and can be found in [6, 12]. For our purposes, just the means of the distributions will suffice:

$$E[TW_1] \approx -1.21, \quad E[TW_2] \approx -1.77, \quad E[TW_{2.3}] \approx -1.8,$$

in which TW_β is a random variable following the Tracy–Widom distribution of parameter β . (The means for $\beta = 1$ and $\beta = 2$ were computed using a numerical ode solver and the Tracy–Widom densities. The mean for $\beta = 2.3$ was computed by Monte Carlo simulation.)

Although Johnstone's theorem handles “rectangular” cases ($m \neq n$), we present only the “square” case. Note that Johnstone's complex Wishart matrix differs from ours by a factor of $\frac{1}{2}$.

THEOREM 4.2 (see [6]). *Fix $\beta = 1$ or $\beta = 2$. Let l_m be the largest eigenvalue of $W_{m-1,m+1}^\beta$, and scale each random variable in the sequence by*

$$(4.2) \quad \tilde{l}_m = \frac{l_m - 4\beta m}{2^{4/3}\beta m^{1/3}}.$$

Then the sequence $\tilde{l}_m$ converges in distribution to the Tracy–Widom largest eigenvalue distribution for β .

CONJECTURE 4.3. *The theorem also holds for general β .*

Although the generalized Tracy–Widom distributions ($\beta \neq 1, 2, 4$) have no known closed form solutions, the distributions can be sampled quite efficiently when $\beta \geq 1$ [4, 9].

TABLE 4.1

Large m asymptotic approximation to $\lim_{x \rightarrow \infty} x^\beta P[\kappa > x]$ when $m = n$. These estimates are based on Conjecture 4.3.

β	Large $m = n$ asymptotics for $C\mu = \lim_{x \rightarrow \infty} x^\beta P[\kappa > x]$
1	$(\sqrt{m} - \frac{1}{4\sqrt{m}})(\sqrt{m} + \sqrt{m-1} + \frac{1}{2^{2/3}}E[TW_1]m^{-1/6})$
General	$\frac{2\beta^\beta}{\Gamma(1+\beta)}m^\beta(1 + \frac{\beta-2}{4}m^{-1})(1 + 2^{-5/3}\beta E[TW_\beta]m^{-2/3})$

4.4. Second order asymptotics. Now, we can compute a second order approximation to the constant $C\mu$. Stirling's second order approximation gives

$$C = \frac{1}{2^{\beta/2-1}\Gamma(1+\beta)} \frac{\Gamma(\frac{\beta}{2}(m+1))}{\Gamma(\frac{\beta}{2}m)}$$

$$\sim \frac{\beta^{\beta/2}}{2^{\beta-1}\Gamma(1+\beta)} m^{\beta/2} \left(1 + \frac{\beta-2}{4}m^{-1}\right) \quad (m \rightarrow \infty),$$

and Johnstone's result suggests

$$\begin{aligned} \mu &= E[\lambda_{\max}(W_{m-1,m+1}^\beta)^{\beta/2}] \\ &\sim E[(4\beta m + 2^{4/3}\beta m^{1/3}TW_\beta)^{\beta/2}] \\ &\sim E\left[(4\beta m)^{\beta/2} + \frac{\beta}{2}(4\beta m)^{\beta/2-1}2^{4/3}\beta m^{1/3}TW_\beta\right] \\ &\sim 2^\beta \beta^{\beta/2} m^{\beta/2} (1 + 2^{-5/3}\beta E[TW_\beta]m^{-2/3}) \quad (m \rightarrow \infty). \end{aligned}$$

To avoid questions of convergence, we simply label this estimate as heuristic. When $\beta = 1$ and m is small, the moment is better approximated by

$$\begin{aligned} \mu &\sim \beta^{\beta/2}(\sqrt{m} + \sqrt{m-1})^\beta + 2^{\beta-5/3}\beta^{\beta/2+1}E[TW_\beta]m^{\beta/2-2/3} \\ &\sim \sqrt{m} + \sqrt{m-1} + 2^{-2/3}E[TW_1]m^{-1/6} \quad (m \rightarrow \infty), \end{aligned}$$

as suggested by [6]. (Of course, this estimate agrees with the previous one asymptotically. There may be a generalization of this formula to general β , providing a reasonably accurate estimate for small m without modifying the large m asymptotics, but we are not aware of one.)

Multiplying the estimates gives Table 4.1.

4.5. Small m . For small values of m , the constant $C\mu$ can be computed directly. In the cases $\beta = 1, 2$, the constant can be computed symbolically in Mathematica; see Table 4.2(a) and (b). For other values of β , the moment can be evaluated numerically using (4.1); see Table 4.2(c). The numerical entries in this table were computed with the help of Plamen Koev's software for multivariate hypergeometric functions [7].

5. Asymptotics at $\kappa = 1$. Asymptotics can also be derived near $\kappa = 1$. The proofs are broken into two cases, $p \leq 1$ and $p \geq 1$.

Throughout this section

$$D = m! c_{m,n}^\beta \frac{I}{\beta \binom{m}{2} + m - 1} \left(\frac{2}{m}\right)^{\beta \binom{m}{2} + mp} \Gamma\left(\beta \binom{m}{2} + mp\right),$$

TABLE 4.2
Preferred approximation to $\lim_{x \rightarrow \infty} x^\beta P[\kappa > x]$ when $m = n$.

(a) Real case	
$m = n$	$\beta = 1$: Preferred approximation to $C\mu = \lim_{x \rightarrow \infty} xP[\kappa > x]$
2	2
3	$11/(2\sqrt{2})$
4	$2^{-13/2}(2510\sqrt{2}/9 + 105\pi - 210 \tan^{-1}(\sqrt{2}))$
≥ 5	$(\sqrt{m} - \frac{1}{4\sqrt{m}})(\sqrt{m} + \sqrt{m-1} + \frac{1}{2^{2/3}}E[TW_1]m^{-1/6})$
(b) Complex case	
$m = n$	$\beta = 2$: Preferred approximation to $C\mu = \lim_{x \rightarrow \infty} x^2P[\kappa > x]$
2	6
3	297/16
4	1332815/34992
5	4512225660125/69657034752
≥ 6	$4m^2(1 + 2^{-2/3}E[TW_2]m^{-2/3})$
(c) Neither real nor complex	
$m = n$	$\beta = 2.3$: Preferred approximation to $C\mu = \lim_{x \rightarrow \infty} x^{2.3}P[\kappa > x]$
2	8.54
3	30.44
4	68.81
5	126.03
≥ 6	$4m^2(1 + 2^{-2/3}E[TW_{2.3}]m^{-2/3})$

where

$$I = \int \cdots \int \prod_{1 \leq j < i \leq m} |t_i - t_j|^\beta dt_2 \cdots dt_{m-1},$$

the integral taken over the region $0 = t_1 \leq t_2 \leq \cdots \leq t_{m-1} \leq t_m = 1$.

LEMMA 5.1. *Let κ be the condition number associated with a Wishart matrix of parameters m , n , and β , and suppose that $p \leq 1$. Then for all $x \geq 1$, the following upper and lower bounds hold:*

$$\begin{aligned} P[\kappa \leq x] &\leq D(x^2 - 1)^{\beta \binom{m}{2} + m - 1}, \\ P[\kappa \leq x] &\geq D \left(\frac{x^2 - 1}{x^2} \right)^{\beta \binom{m}{2} + m - 1}. \end{aligned}$$

Proof. Begin by changing to variables $\lambda_1, t_2, \dots, t_{m-1}, \Delta$:

$$\begin{aligned} \lambda_1 &= \lambda_1, \\ t_i &= \frac{\lambda_i - \lambda_1}{\lambda_m - \lambda_1}, \quad i = 2, \dots, m-1, \\ \Delta &= \lambda_m - \lambda_1. \end{aligned}$$

The associated Jacobian matrix is lower triangular, and the Jacobian is Δ^{m-2} . Introducing the shorthand $t_1 = 0$ and $t_m = 1$, the joint density of $\lambda_1, t_2, \dots, t_{m-1}$,

Δ is

$$\begin{aligned} & f(\lambda_1, t_2, \dots, t_{m-1}, \Delta) \\ &= m! c_{m,n}^\beta \Delta^{\beta \binom{m}{2} + m - 2} e^{-m\lambda_1/2} \prod_{1 \leq j < i \leq m} |t_i - t_j|^\beta \prod_{i=1}^m (\lambda_1 + t_i \Delta)^{p-1} e^{-(\Delta/2) \sum_{i=1}^m t_i}. \end{aligned}$$

Because

$$P[\kappa \leq x] = \int_0^\infty \int_0^{\lambda_1(x^2-1)} \left[\int \cdots \int f(\lambda_1, t_2, \dots, t_{m-1}, \Delta) dt_2 \cdots dt_{m-1} \right] d\Delta d\lambda_1$$

with the inner integral taken over the region $0 = t_1 \leq t_2 \leq \cdots \leq t_{m-1} \leq t_m = 1$, we have

$$(5.1) \quad P[\kappa \leq x] = m! c_{m,n}^\beta \int_0^\infty \int_0^{\lambda_1(x^2-1)} \Delta^{\beta \binom{m}{2} + m - 2} g(\lambda_1, \Delta) d\Delta d\lambda_1,$$

where

$$(5.2) \quad g(\lambda_1, \Delta) = \frac{\int \cdots \int f(\lambda_1, t_2, \dots, t_{m-1}, \Delta) dt_2 \cdots dt_{m-1}}{m! c_{m,n}^\beta \Delta^{\beta \binom{m}{2} + m - 2}}.$$

The asymptotic decay rate $(x^2 - 1)^{\beta \binom{m}{2} + m - 1}$ seems inevitable.

To obtain the bounds, observe

$$\begin{aligned} g(\lambda_1, \Delta) &\leq \lambda_1^{m(p-1)} e^{-m\lambda_1/2} I, \\ g(\lambda_1, \Delta) &\geq (\lambda_1 + \Delta)^{m(p-1)} e^{-m(\lambda_1 + \Delta)/2} I. \end{aligned}$$

Therefore,

$$\begin{aligned} P[\kappa \leq x] &\leq m! c_{m,n}^\beta I \int_0^\infty \lambda_1^{m(p-1)} e^{-m\lambda_1/2} \left(\int_0^{\lambda_1(x^2-1)} \Delta^{\beta \binom{m}{2} + m - 2} d\Delta \right) d\lambda_1 \\ &= D(x^2 - 1)^{\beta \binom{m}{2} + m - 1}, \\ P[\kappa \leq x] &\geq m! c_{m,n}^\beta I \int_0^\infty \int_0^{\lambda_1(x^2-1)} \Delta^{\beta \binom{m}{2} + m - 2} (\lambda_1 + \Delta)^{m(p-1)} e^{-m(\lambda_1 + \Delta)/2} d\Delta d\lambda_1 \\ &= m! c_{m,n}^\beta I \int_0^\infty \lambda_m^{m(p-1)} e^{-m\lambda_m/2} \left(\int_0^{\lambda_m(x^2-1)/x^2} \Delta^{\beta \binom{m}{2} + m - 2} d\Delta \right) d\lambda_m \\ &= D \left(\frac{x^2 - 1}{x^2} \right)^{\beta \binom{m}{2} + m - 1}. \quad \square \end{aligned}$$

LEMMA 5.2. *Let κ be the condition number associated with a Wishart matrix of parameters m , n , and β , and suppose that $p \geq 1$. Then for all $x \geq 1$, the following upper and lower bounds hold:*

$$\begin{aligned} P[\kappa \leq x] &\leq Dx^2(x^2 - 1)^{\beta \binom{m}{2} + m - 1}, \\ P[\kappa \leq x] &\geq Dx^{-2(\beta \binom{m}{2} + mp)}(x^2 - 1)^{\beta \binom{m}{2} + m - 1}. \end{aligned}$$

Proof. (5.1) and (5.2) from the proof of the previous lemma are still true. Now that $p \geq 1$, we have the bounds

$$\begin{aligned} g(\lambda_1, \Delta) &\leq (\lambda_1 + \Delta)^{m(p-1)} e^{-m\lambda_1/2} I, \\ g(\lambda_1, \Delta) &\geq \lambda_1^{m(p-1)} e^{-m(\lambda_1 + \Delta)/2} I. \end{aligned}$$

Therefore,

$$\begin{aligned} P[\kappa \leq x] &\leq m! c_{m,n}^\beta I \int_0^\infty \int_0^{\lambda_1(x^2-1)} \Delta^{\beta(\frac{m}{2})+m-2} (\lambda_1 + \Delta)^{m(p-1)} e^{-m\lambda_1/2} d\Delta d\lambda_1 \\ &\leq m! c_{m,n}^\beta I \int_0^\infty \lambda_m^{m(p-1)} e^{-m(\lambda_m/x^2)/2} \left(\int_0^{\lambda_m(x^2-1)/x^2} \Delta^{\beta(\frac{m}{2})+m-2} d\Delta \right) d\lambda_m \\ &= Dx^2(x^2-1)^{\beta(\frac{m}{2})+m-1}, \\ P[\kappa \leq x] &\geq m! c_{m,n}^\beta I \int_0^\infty \int_0^{\lambda_1(x^2-1)} \Delta^{\beta(\frac{m}{2})+m-2} \lambda_1^{m(p-1)} e^{-m(\lambda_1 + \Delta)/2} d\Delta d\lambda_1 \\ &\geq m! c_{m,n}^\beta I \int_0^\infty \lambda_1^{m(p-1)} e^{-m(\lambda_1 x^2)/2} \left(\int_0^{\lambda_1(x^2-1)} \Delta^{\beta(\frac{m}{2})+m-2} d\Delta \right) d\lambda_1, \\ &= Dx^{-2(\beta(\frac{m}{2})+mp)} (x^2-1)^{\beta(\frac{m}{2})+m-1}. \quad \square \end{aligned}$$

THEOREM 5.3. *Let κ be the condition number associated with a Wishart matrix of parameters m , n , and β . Then*

$$P[\kappa \leq x] \sim D(x^2-1)^{\beta(\frac{m}{2})+m-1} \quad (x \rightarrow 1).$$

6. Two-by-two matrices. Performing the change of variables $\kappa = \sqrt{\lambda_2/\lambda_1}$ and integrating out λ_1 in the joint density (2.1), one arrives at the density of the condition number of a 2-by-2 Wishart matrix,

$$f(x) = \frac{4\Gamma(1 + \frac{\beta}{2})\Gamma(\beta n)}{\Gamma(1 + \beta)\Gamma(p)\Gamma(\frac{\beta}{2}n)} \frac{x^{2p-1}(x^2-1)^\beta}{(x^2+1)^{\beta n}}.$$

Near $\kappa = \infty$, the density is asymptotic to

$$f(x) \sim \frac{4\Gamma(1 + \frac{\beta}{2})\Gamma(\beta n)}{\Gamma(p)\Gamma(1 + \beta)\Gamma(\frac{\beta}{2}n)} x^{-2p-1} \quad (x \rightarrow \infty).$$

Integrating,

$$P[\kappa > x] \sim \frac{2\Gamma(1 + \frac{\beta}{2})\Gamma(\beta n)}{\Gamma(1 + p)\Gamma(1 + \beta)\Gamma(\frac{\beta}{2}n)} x^{-2p} \quad (x \rightarrow \infty).$$

If $n = 2 = m$, this simplifies to

$$P[\kappa > x] \sim \frac{2\Gamma(2\beta)}{\Gamma(1 + \beta)\Gamma(\beta)} x^{-\beta} \quad (x \rightarrow \infty).$$

The prediction from Corollary 3.3 is

$$P[\kappa > x] \sim \frac{\Gamma(\frac{3\beta}{2})}{2^{\beta/2-1}\Gamma(1+\beta)\Gamma(\beta)} \mu x^{-\beta} \quad (x \rightarrow \infty)$$

with

$$\mu = E[\lambda_{\max}(W_{1,3}^\beta)^{\beta/2}] = E[\chi_{3\beta}^\beta] = \frac{2^{\beta/2}\Gamma(2\beta)}{\Gamma(3\beta/2)},$$

so that

$$P[\kappa > x] \sim \frac{2\Gamma(2\beta)}{\Gamma(1+\beta)\Gamma(\beta)} x^{-\beta} \quad (x \rightarrow \infty).$$

The main theorem is verified in this case.

REFERENCES

- [1] J.-M. AZAÏS AND M. WSCHBOR, *Upper and lower bounds for the tails of the distribution of the condition number of a Gaussian matrix*, SIAM J. Matrix Anal. Appl., 26 (2005), pp. 426–440.
- [2] J. W. DEMMEL, *The probability that a numerical analysis problem is difficult*, Math. Comp., 50 (1988), pp. 449–480.
- [3] I. DUMITRIU, *Eigenvalue Statistics for Beta-Ensembles*, Ph.D. thesis, Massachusetts Institute of Technology, Cambridge, MA, 2003.
- [4] I. DUMITRIU AND A. EDELMAN, *Matrix models for beta ensembles*, J. Math. Phys., 43 (2002), pp. 5830–5847.
- [5] A. EDELMAN, *Eigenvalues and Condition Numbers of Random Matrices*, Ph.D. thesis, Massachusetts Institute of Technology, Cambridge, MA, 1989.
- [6] I. M. JOHNSTONE, *On the distribution of the largest eigenvalue in principal components analysis*, Ann. Statist., 29 (2001), pp. 295–327.
- [7] P. KOEV AND A. EDELMAN, *Efficient evaluation of the hypergeometric function of matrix argument*, Math. Comp., submitted.
- [8] R. J. MUIRHEAD, *Aspects of Multivariate Statistical Theory*, Wiley Series Probab. Math. Statist., John Wiley & Sons, New York, 1982.
- [9] P.-O. PERSSON, *Numerical Methods for Eigenvalue Distributions of Random Matrices*, <http://arxiv.org/abs/math-ph/0501068> (19 March 2005).
- [10] J. W. SILVERSTEIN, *The smallest eigenvalue of a large-dimensional Wishart matrix*, Ann. Probab., 13 (1985), pp. 1364–1368.
- [11] S. SMALE, *On the efficiency of algorithms of analysis*, Bull. Amer. Math. Soc. (N.S.), 13 (1985), pp. 87–121.
- [12] C. A. TRACY AND H. WIDOM, *Airy kernel and Painlevé II*, in *Isomonodromic Deformations and Applications in Physics* (Montréal, QC, 2000), CRM Proc. Lect. Notes 31, J. Harnad and A. Its, eds., American Mathematical Society, Providence, RI, 2002, pp. 85–96.
- [13] J. VON NEUMANN, *Collected Works, Vol. V: Design of Computers, Theory of Automata and Numerical Analysis*, A Pergamon Press Book, Macmillan, New York, 1963.

PERTURBATION THEORY FOR FACTORIZATIONS OF LU TYPE THROUGH SERIES EXPANSIONS*

FROILÁN M. DOPICO[†] AND JUAN M. MOLERA[†]

Abstract. Component- and normwise perturbation bounds for the block LU factorization and block LDL* factorization of Hermitian matrices are presented. We also obtain, as a consequence, perturbation bounds for the usual pointwise LU, LDL*, and Cholesky factorizations. Some of these latter bounds are already known, but others improve previous results. All the bounds presented are easily proved by using series expansions. Given a square matrix $A = LU$ having the LU factorization, and a perturbation E , the LU factors of the matrix $A + E = \tilde{L}\tilde{U}$ are written as two convergent series of matrices: $\tilde{L} = \sum_{k=0}^{\infty} L_k$ and $\tilde{U} = \sum_{k=0}^{\infty} U_k$, where $L_k = O(\|E\|^k)$, $U_k = O(\|E\|^k)$, and $L_0 = L$, $U_0 = U$. We present expressions for the matrices L_k and U_k in terms of L , U , and E . The domain and the rate of convergence of these series are studied. Simple bounds on the remainders of any order of these series are found, which significantly improve the bounds on the second-order terms existing in the literature. This is useful when first-order perturbation analysis is used.

Key words. LU factorization, Cholesky factorization, block LU factorization, diagonal pivoting method, block LDL^T factorization, perturbation theory, series expansion

AMS subject classifications. 65F35, 15A23

DOI. 10.1137/040612142

1. Introduction. Let A be an $n \times n$ matrix whose leading principal submatrices are nonsingular. Then there exists a unique unit lower triangular matrix L and a unique upper triangular matrix U such that $A = LU$. This is known as LU factorization of A . The LU factorization is one of the most important matrix factorizations appearing in numerical analysis and has several variants that are used in different contexts: Cholesky, LDL*, block LU, and block LDL* factorizations. As usual L^* denotes the conjugate transpose of L .

Traditionally, the LU factorization and its variants have been used to solve linear systems of equations, while in solving spectral problems orthogonal factorizations have been preferred because of their good stability properties [15]. However, in the last decade the LU factorization has been employed to solve structured spectral problems [12, 19]. For most applications related to the solution of linear systems it is the backward error and not the forward error of the LU factorization that matters; however, for the application of LU to the computation of the singular value decomposition with a high relative accuracy, what is needed instead is to compute the LU factors with small forward errors [12]. The question of how big the forward errors are may be answered by combining backward errors with an adequate perturbation theory for the LU factorization. This paper presents a new broad-scoped and unifying approach to this theory, which allows us to get, for the usual point factorizations, some bounds that are already known and, furthermore, to improve previously existing bounds. Most important, this general analysis also includes, for the first time, perturbation bounds for the block LU factorization. The block LU factorization appears

*Received by the editors July 23, 2004; accepted for publication (in revised form) by N. J. Higham May 6, 2005; published electronically November 22, 2005. This work was partially supported by the Ministerio de Educación y Ciencia of Spain through grant BFM 2003-00223.

<http://www.siam.org/journals/simax/27-2/61214.html>

[†]Departamento de Matemáticas, Universidad Carlos III de Madrid, Avenida de la Universidad 30, 28911 Leganés, Spain (dopico@math.uc3m.es, molera@math.uc3m.es).

in different instances [17, Chapter 13], and it is used in the most common algorithms to solve Hermitian indefinite systems of linear equations [1].

In the literature there are several papers dealing with the perturbation theory of usual, or pointwise, LU or Cholesky factorizations. In these papers, first-order perturbation bounds are frequently used. In this respect, we will say that strict or rigorous perturbation bounds are those without higher-order terms.

The first works [3, 23, 26] dealing with this problem obtained normwise perturbation bounds. Sun [27, 28] got an improvement over previously existing results by proving componentwise perturbation bounds. In [24], Stewart shed new light on the problem by giving explicit formulae for the first-order terms of the series of the perturbed factors of LU, Cholesky, and QR factorizations. Moreover, normwise bounds on the second-order terms, i.e., the remainder of the series, were also presented.

There has also been relevant work on the true normwise condition number for the Cholesky factorization by Chang, Paige, and Stewart [8, 25]. Ideas similar to those in [8] have also been used in [9] to study the normwise condition number of the LU factorization and in [10] to give a structured perturbation analysis of the Cholesky factorization. Another structured analysis of component- and normwise condition numbers of the LU factors of tridiagonal matrices has been done in [4], where easily computable expressions of the true condition numbers are presented. In these works, many different techniques have been used. One of the goals of this paper is to present a unifying perturbation theory for LU-type factorizations.

In this work, expressions for the terms of any order in the series expansions of the perturbed LU or Cholesky factors are presented. These series are easily and naturally extended to cover the block LU factorization and block LDL* factorization of Hermitian indefinite matrices. Extremely simple componentwise upper bounds for all the terms in these series will be obtained. Summing up the series for these upper bounds, reliable bounds on the remainders of any order of the series and rigorous component- and normwise perturbation bounds for the LU factors are proved. The bounds for the block factorizations are the first existing perturbation bounds for these factorizations. Some of the bounds for the usual factorizations are already known, but others improve previously existing bounds.

The paper is organized as follows. In section 2 the series of the LU factors are written as two convergent series of matrices using basic results. The domain of convergence of these series is also established. In section 3 simple componentwise bounds for the terms of these series are proved. This is a technical section which contains the crucial result on which the rest of the paper is based: Theorem 3.6, which remains valid for block factorizations. As a simple consequence, in section 4 the main theorems on the perturbation of the LU factors of a matrix are stated; they are a comprehensive summary of the component- and normwise convergence properties of the series. Moreover, it is shown how to get strict component- and normwise perturbation bounds from these series. Along this line, section 5 presents a rigorous perturbation theory for the block LU factorization. Using the results in section 5, and properties of Hermitian matrices, perturbation bounds for the block LDL* factorization are developed in section 6. The case of D being diagonal is considered and compared with previous results. Results for the Cholesky factorization are presented in section 7.

2. Series for the LU factors of a matrix. We will deal with a nonsingular $n \times n$ complex matrix $A = [a_{ij}]$ such that its leading principal submatrices are nonsingular. Therefore, A has a unique LU factorization: $A = LU$. Throughout this paper $\|\cdot\|$ will denote a family of absolute, consistent matrix norms; i.e., if $|A|$ is the matrix

with entries $(|A|)_{ij} = |a_{ij}|$, then $|A| \leq |B|$ implies $\|A\| \leq \|B\|$ and $\|AB\| \leq \|A\|\|B\|$. Norms of this type are the Frobenius norm, the 1-norm, and the ∞ -norm, but not the 2-norm. Notice that if $A(1:k, 1:k)$ denotes the k th leading principal submatrix of A , then $\|A(1:k, 1:k)\| \leq \|A\|$.

Let us consider a perturbation $\tilde{A} = A + E$ of A and assume that $\|L^{-1}EU^{-1}\| < 1$. In this case $\tilde{A}$ has also a unique LU factorization because the matrix $I + L^{-1}EU^{-1}$ has all its leading principal submatrices nonsingular and has the following unique LU factorization:

$$(2.1) \quad I + L^{-1}EU^{-1} = \mathcal{L}\mathcal{U}.$$

Therefore,

$$(2.2) \quad \tilde{A} = L(I + L^{-1}EU^{-1})U = (L\mathcal{L})(\mathcal{U}U) \equiv \tilde{L}\tilde{U}$$

is the unique LU factorization of the nonsingular matrix $\tilde{A}$, because it is well known that if the LU factorization of a nonsingular matrix exists, then it is unique.

The set of matrices E for which we can guarantee that $I + L^{-1}EU^{-1}$ has a unique LU factorization can be enlarged by using the spectral radius of $|L^{-1}EU^{-1}|$, denoted by $\rho(|L^{-1}EU^{-1}|)$. Notice that

$$(2.3) \quad \rho(|L^{-1}EU^{-1}|) < 1$$

implies that $I + L^{-1}EU^{-1}$ has its leading principal submatrices nonsingular. The reason is that for any $n \times n$ matrix F and for any k such that $1 \leq k \leq n$,

$$(2.4) \quad \rho(F(1:k, 1:k)) \leq \rho(|F(1:k, 1:k)|) \leq \rho(|F|)$$

holds; see [18, Theorem 8.1.18, Corollary 8.1.20]. Besides this, $\rho(|L^{-1}EU^{-1}|) \leq \|L^{-1}EU^{-1}\|$ implies that (2.3) remains valid for a wider set of perturbation matrices E than $\|L^{-1}EU^{-1}\| < 1$, which can be more easily checked in practice.

Equation (2.2) shows that for getting a Taylor-like series for the LU factors of $\tilde{A}$ it is enough to study the series of the LU factors of perturbations $I + F$ of the identity matrix. For the sake of simplicity, let us introduce a complex variable z , write the perturbation as $I + zF$, and consider the LU factors of this matrix as complex functions of z , i.e., $I + zF = \mathcal{L}(z)\mathcal{U}(z)$ (at the end we will set $z = 1$ and $\mathcal{L} = \mathcal{L}(1)$, $\mathcal{U} = \mathcal{U}(1)$). We know that these LU factors exist and are unique in the open disk $D_F = \{z \in \mathbb{C} : |z| < \frac{1}{\rho(|F|)}\}$. Moreover, the entries of $\mathcal{L}(z)$ and $\mathcal{U}(z)$ are rational functions of z with nonzero denominator in D_F . This can be seen by examining the steps of Gaussian elimination or, directly, by using the well-known determinantal formulas for the entries of the LU factors of a matrix (see [17, p. 161]). Thus the entries of $\mathcal{L}(z)$ and $\mathcal{U}(z)$ are analytic functions of z in D_F , and the corresponding Taylor series around $z = 0$ are convergent power series in D_F . These scalar series can be compactly written in matrix notation for every $z \in D_F$ as follows:

$$\mathcal{L}(z) = \sum_{k=0}^{\infty} z^k \mathcal{L}_k \quad \text{and} \quad \mathcal{U}(z) = \sum_{k=0}^{\infty} z^k \mathcal{U}_k.$$

The goal is to get simple expressions for the lower triangular matrices $\mathcal{L}_k$ and the upper triangular matrices $\mathcal{U}_k$. Taking into account that inside D_F we can sum up and

multiply in the usual way the Taylor series of the entries of the matrices $\mathcal{L}(z)$ and $\mathcal{U}(z)$ to get the convergent power series, we have

$$I + zF = \mathcal{L}(z)\mathcal{U}(z) = \sum_{k=0}^{\infty} \left(\sum_{j=0}^k \mathcal{L}_j \mathcal{U}_{k-j} \right) z^k.$$

Consequently,

$$(2.5) \quad I = \mathcal{L}_0 \mathcal{U}_0, \quad F = \mathcal{L}_0 \mathcal{U}_1 + \mathcal{L}_1 \mathcal{U}_0,$$

$$(2.6) \quad 0 = \mathcal{L}_0 \mathcal{U}_k + \mathcal{L}_1 \mathcal{U}_{k-1} + \cdots + \mathcal{L}_{k-1} \mathcal{U}_1 + \mathcal{L}_k \mathcal{U}_0 \quad \text{for } k \geq 2.$$

Notice that $\mathcal{L}(z)$ has diagonal entries equal to 1; therefore, $\mathcal{L}_0$ has also diagonal entries equal to 1 and $\mathcal{L}_k$ for $k \geq 1$ are strictly lower triangular matrices. Using this in (2.5) and (2.6), we get the following recurrence relation for the matrices $\mathcal{L}_k$ and $\mathcal{U}_k$:

$$\begin{aligned} \mathcal{L}_0 = \mathcal{U}_0 = I, \quad \mathcal{L}_1 + \mathcal{U}_1 &= F, \\ \mathcal{L}_k + \mathcal{U}_k &= -\mathcal{L}_1 \mathcal{U}_{k-1} - \mathcal{L}_2 \mathcal{U}_{k-2} - \cdots - \mathcal{L}_{k-1} \mathcal{U}_1 \quad \text{for } k \geq 2. \end{aligned}$$

The key point to obtain $\mathcal{L}_k$ and $\mathcal{U}_k$ for $k \geq 1$ is that $\mathcal{L}_k$ is strictly lower triangular and $\mathcal{U}_k$ is upper triangular. The previous discussion can be summarized in the next theorem, where we have set $z = 1$ and the following notation is used: Given a matrix B with entries b_{ij} , we call B_L and B_U , respectively, the *strictly* lower triangular part and the upper triangular part of B , i.e.,

$$(2.7) \quad (B_L)_{ij} = \begin{cases} b_{ij} & \text{if } i > j, \\ 0 & \text{otherwise,} \end{cases} \quad (B_U)_{ij} = \begin{cases} b_{ij} & \text{if } i \leq j, \\ 0 & \text{otherwise.} \end{cases}$$

THEOREM 2.1. *Let F be an $n \times n$ complex matrix, such that $\rho(|F|) < 1$. Then*

1. $I + F$ has a unique LU factorization: $I + F = \mathcal{L}\mathcal{U}$;
- 2.

$$\mathcal{L} = \sum_{k=0}^{\infty} \mathcal{L}_k \quad \text{and} \quad \mathcal{U} = \sum_{k=0}^{\infty} \mathcal{U}_k,$$

where

$$(2.8) \quad \mathcal{L}_0 = I, \quad \mathcal{U}_0 = I; \quad \mathcal{L}_1 = F_L, \quad \mathcal{U}_1 = F_U,$$

$$(2.9) \quad \mathcal{L}_k = (-\mathcal{L}_1 \mathcal{U}_{k-1} - \mathcal{L}_2 \mathcal{U}_{k-2} - \cdots - \mathcal{L}_{k-1} \mathcal{U}_1)_L \quad \text{for } k \geq 2,$$

$$(2.10) \quad \mathcal{U}_k = (-\mathcal{L}_1 \mathcal{U}_{k-1} - \mathcal{L}_2 \mathcal{U}_{k-2} - \cdots - \mathcal{L}_{k-1} \mathcal{U}_1)_U \quad \text{for } k \geq 2;$$

3. $\mathcal{L}_k = O(\|F\|^k)$ and $\mathcal{U}_k = O(\|F\|^k)$.

The last item can be easily proven by using an inductive argument. In fact it can be proven (from Theorem 2.2) that $\|\mathcal{L}_k + \mathcal{U}_k\| \leq C_{k-1} \|F\|^k$ for $k \geq 1$. The numbers C_k appear in many counting problems; they are known as the *Catalan numbers* [20, p. 315] and they grow quickly with k . We will see in section 3 that a more elaborate argument allows us to bound $|\mathcal{L}_k + \mathcal{U}_k| \leq |F|^k$.

The first-order terms were obtained in [24]; now we can easily write explicit expressions for the higher-order terms. For instance,

$$(2.11) \quad \mathcal{L}_2 + \mathcal{U}_2 = -F_L F_U,$$

$$(2.12) \quad \mathcal{L}_3 + \mathcal{U}_3 = F_L (F_L F_U)_U + (F_L F_U)_L F_U,$$

$$(2.13) \quad \begin{aligned} \mathcal{L}_4 + \mathcal{U}_4 &= -F_L (F_L (F_L F_U)_U)_U - F_L ((F_L F_U)_L F_U)_U \\ &\quad - (F_L F_U)_L (F_L F_U)_U - (F_L (F_L F_U)_U)_L F_U - ((F_L F_U)_L F_U)_L F_U. \end{aligned}$$

For a series to be useful from the point of view of applications, its rate of convergence has to be fast enough. Taking into account that the number of terms in the recurrence relation of $\mathcal{L}_k$ and $\mathcal{U}_k$ increases very fast with k , a convenient analysis of the rate of convergence of these series requires a careful approach that will be undertaken in the following sections. We begin this by giving a more explicit description of $\mathcal{L}_k$ and $\mathcal{U}_k$. Notice that the next result holds for any square matrix F , independently of the magnitude of its spectral radius.

THEOREM 2.2. *Let F be an $n \times n$ complex matrix, and $\mathcal{L}_k$ and $\mathcal{U}_k$, $k \geq 1$, be the matrices defined by the recurrence relation (2.8)–(2.10). Then $\mathcal{L}_k + \mathcal{U}_k$ is $(-1)^{k+1}$ times the sum of all the matrices that can be formed with the following symbolic rules.*

1. *Specify, by using parentheses, all the possible different orders to multiply $F^k = F F \cdots F$.*

2. *Given an order of multiplication, every individual multiplication CD is converted to $C_L D_U$.*

Before the proof of this theorem, let us give an example showing how to use it.

EXAMPLE 1. *The product $F^3 = FFF$ can be computed in two orders, $F(FF)$ and $(FF)F$. In any of these orders there are two individual multiplications, one inside the parentheses and the other outside. The second rule of the previous theorem applies to the inner and outer multiplications to yield $F_L(F_L F_U)_U$ and $(F_L F_U)_L F_U$. Therefore, Theorem 2.2 gives $\mathcal{L}_3 + \mathcal{U}_3 = F_L(F_L F_U)_U + (F_L F_U)_L F_U$, which is the result obtained in (2.12) by using Theorem 2.1.*

Proof of Theorem 2.2. The proof is by induction on k . The result is trivially true for $k = 1, 2$. Assume that it is true for any j such that $1 \leq j \leq k$. It is known from (2.8)–(2.10) that

$$(2.14) \quad \mathcal{L}_{k+1} + \mathcal{U}_{k+1} = -\mathcal{L}_1 \mathcal{U}_k - \mathcal{L}_2 \mathcal{U}_{k-1} - \cdots - \mathcal{L}_k \mathcal{U}_1.$$

In any order of multiplication of F^{k+1} , specified with parentheses, there is one and only one multiplication operator which remains outside all parentheses. This is the final multiplication to be performed. Notice that $\mathcal{L}_j = (\mathcal{L}_j + \mathcal{U}_j)_L$ and $\mathcal{U}_j = (\mathcal{L}_j + \mathcal{U}_j)_U$; therefore, $-\mathcal{L}_j \mathcal{U}_{k+1-j}$ is $(-1)^{k+2}$ times the sum of all the matrices that can be formed as follows: (1) consider all the possible orders to multiply F^{k+1} in which the final multiplication is between the factors j and $j+1$; (2) apply to each of these orders the second rule in Theorem 2.2. Substituting this sum in (2.14) for $1 \leq j \leq k$ the theorem is proved, because the set of all the possible orders to multiply F^{k+1} is the union of the sets of all the possible orders to multiply F^{k+1} in which the final multiplication is between the factors j and $j+1$ for $j = 1, \dots, k$. $\square$

3. Componentwise bounds on $\mathcal{L}_k$ and $\mathcal{U}_k$. The matrices $\mathcal{L}_k$ and $\mathcal{U}_k$ appearing in Theorem 2.1 and characterized in Theorem 2.2 are involved functions of F ; however, they can be simply bounded by powers of the absolute value of F .

THEOREM 3.1. *Let F be an $n \times n$ complex matrix, and let $\mathcal{L}_k$ and $\mathcal{U}_k$, $k \geq 1$, be the matrices defined by the recurrence relation (2.8)–(2.10); then*

$$|\mathcal{L}_k + \mathcal{U}_k| \leq |F|^k \quad \text{for } k \geq 1.$$

Therefore, $|\mathcal{L}_k| \leq (|F|^k)_L$ and $|\mathcal{U}_k| \leq (|F|^k)_U$.

The rest of this paper is based on this theorem, and on its generalized version Theorem 3.6. The proof of both theorems is the goal of this section. Since the proof is long, although elementary, the reader who is interested only in applications can skip this section. It is worthwhile to remark that Theorem 3.1 is valid for any square matrix F , independently of the magnitude of its spectral radius.

The next lemma shows that we can focus on the series of the LU factors of $|F|$.

LEMMA 3.2. *Let F be an $n \times n$ complex matrix, let $\mathcal{L}_k$ and $\mathcal{U}_k$, $k \geq 1$, be the matrices defined by the recurrence relation (2.8)–(2.10), and let $\bar{\mathcal{L}}_k$ and $\bar{\mathcal{U}}_k$ be the matrices defined by the recurrence relation (2.8)–(2.10) applied to the matrix $|F|$, i.e., $\bar{\mathcal{L}}_1 = |F|_L$ and $\bar{\mathcal{U}}_1 = |F|_U$. Then*

$$|\mathcal{L}_k + \mathcal{U}_k| \leq (-1)^{k+1} (\bar{\mathcal{L}}_k + \bar{\mathcal{U}}_k) = |\bar{\mathcal{L}}_k + \bar{\mathcal{U}}_k| \quad \text{for } k \geq 1.$$

Proof. In Theorem 2.2, $(-1)^{k+1} (\mathcal{L}_k + \mathcal{U}_k)$ is expressed as a sum of certain matrices. Apply the triangular inequality of the absolute value to this sum. Notice that for any matrices C and D the inequality $|C_L D_U| \leq |C_L| |D_U| = |C|_L |D|_U$ holds. This is applied to the absolute value of the terms of the sum appearing in Theorem 2.2 to get the same terms for $|F|$ instead of F . $\square$

Before starting with the technical details of the section, we would like to show with an example that the idea of the proof of Theorem 3.1 is very simple.

EXAMPLE 2. *Consider the case $k = 4$ and denote for simplicity $G \equiv |F|$:*

$$\begin{aligned} |F|^4 &= (G_L + G_U)^4 \\ &\geq G_L G_L G_L G_U + G_L G_L G_U G_U + G_L G_U G_L G_U + G_L G_U G_U G_U \\ &= G_L G_L G_L G_U + G_L (G_L G_U)_L G_U + G_L (G_L G_U)_U G_U \\ &\quad + G_L G_U G_L G_U + G_L G_U G_U G_U \\ &\geq G_L (G_L (G_L G_U)_U)_U + G_L ((G_L G_U)_L G_U)_U + (G_L (G_L G_U)_U)_L G_U \\ &\quad + (G_L G_U)_L (G_L G_U)_U + ((G_L G_U)_L G_U)_L G_U \\ &= -(\bar{\mathcal{L}}_4 + \bar{\mathcal{U}}_4) \geq |\mathcal{L}_4 + \mathcal{U}_4|. \end{aligned}$$

3.1. Tree terminology and auxiliary results. To prove Theorem 3.1 it is useful to introduce some tree terminology. Among the terminologies commonly used in tree theory we will follow that used in [20, section 8.1]. Since Theorem 3.1 is trivial for $k = 1$, we will consider only the case $k \geq 2$.

DEFINITION 3.3. *Let the matrix Π be one of the summands in the expression of $(-1)^{k+1} (\mathcal{L}_k + \mathcal{U}_k)$, $k \geq 2$, as the sum defined in Theorem 2.2. $\mathcal{T}(\Pi)$ is the ordered full binary tree defined as follows.*

1. *The root vertex represents $\Pi = C_L D_U$. The left child of the root represents C_L , and the right child of the root represents D_U .*

2. *Any vertex ν of $\mathcal{T}(\Pi)$ different from the root represents a matrix B_L or B_U . If $B = P_L Q_U$, then the left child of ν represents P_L , and the right child of ν represents Q_U . If $B = F$, then ν is a leaf of $\mathcal{T}(\Pi)$.*

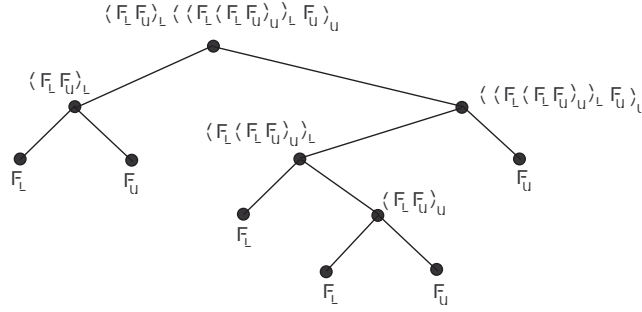
In the previous definition the word “full binary” appears because the internal vertices, i.e., those different from the leaves, have exactly two children. The word “ordered” appears because the children of the internal vertices are ordered from left to right. Moreover, notice that the left children are strictly lower triangular matrices, while the right children are upper triangular matrices. These meanings will be kept in the rest of the paper.

EXAMPLE 3. *Figure 3.1 shows the tree representing the summand*

$$(F_L F_U)_L ((F_L (F_L F_U)_U)_L F_U)_U$$

of $(-\mathcal{L}_6 + \mathcal{U}_6)$.

Based on Definition 3.3, the next definition introduces other trees and some matrices necessary to prove Theorem 3.1. The matrices are important; the trees are

FIG. 3.1. The tree associated with $(F_L F_U)_L ((F_L (F_L F_U)_U)_L F_U)_U$.

used just to define the matrices. In this context, let us remember that the height of a rooted tree is the length of the longest path from the root to any vertex.

DEFINITION 3.4. Let the matrix Π be one of the summands in the expression of $(-1)^{k+1}(\mathcal{L}_k + \mathcal{U}_k)$, $k \geq 2$, as the sum defined in Theorem 2.2. Let h be the height of the tree $\mathcal{T}(\Pi)$. Let us define the following trees and matrices.

1. $\mathcal{T}_h(\Pi) := \mathcal{T}(\Pi)$. The matrix Π_h is the product from left to right of the matrices represented by the leaves of $\mathcal{T}_h(\Pi)$.

2. For $1 \leq j \leq (h-1)$, the tree $\mathcal{T}_{h-j}(\Pi)$ is obtained from the tree $\mathcal{T}_{h-(j-1)}(\Pi)$ by removing all the pairs of leaves that are siblings and the edges incident with the removed leaves. The matrix Π_{h-j} is the product from left to right of the matrices represented by the leaves of $\mathcal{T}_{h-j}(\Pi)$.

The order from left to right for the leaves of the trees in the previous definition is well defined independently of the graphic representation of the tree: given any parent vertex ν , its left child and all the descendants of this left child have to be considered as being to the left of the right child of ν and all the descendants of this right child.

EXAMPLE 4. In the example in Figure 3.1, the matrices in Definition 3.4 are

$$(3.1) \quad \Pi_h = \Pi_4 = F_L F_U F_L F_L F_U F_U,$$

$$(3.2) \quad \Pi_{h-1} = \Pi_3 = (F_L F_U)_L F_L (F_L F_U)_U F_U,$$

$$(3.3) \quad \Pi_{h-2} = \Pi_2 = (F_L F_U)_L (F_L (F_L F_U)_U)_L F_U,$$

$$(3.4) \quad \Pi_{h-3} = \Pi_1 = (F_L F_U)_L ((F_L (F_L F_U)_U)_L F_U)_U.$$

Notice that Π_1 is equal to Π . Moreover, Π_4 is the product of six factors, Π_3 is the product of four factors, Π_2 is the product of three factors, and Π_1 is the product of just two factors.

For the trees and matrices defined in Definition 3.4 the following lemma holds.

LEMMA 3.5. Let $\mathcal{T}_{h-j}(\Pi)$ and Π_{h-j} with $0 \leq j \leq (h-1)$ be, respectively, the trees and matrices defined in Definition 3.4. Then the following results hold.

1. $\mathcal{T}_{h-j}(\Pi)$ is an ordered full binary tree with height $(h-j)$.
2. $\Pi = \Pi_1$.
3. Let $B_1, B_2, \dots, B_q$ be the matrices represented by the leaves of $\mathcal{T}_{h-j}(\Pi)$, i.e., $\Pi_{h-j} = B_1 B_2 \cdots B_q$. Then B_i and B_{i+1} , $1 \leq i \leq (q-1)$, are represented by two leaves that are siblings if and only if B_i is strictly lower triangular and B_{i+1} is upper triangular.

4. Let us use the same notation as in the previous item: $\Pi_{h-j} = B_1 B_2 \cdots B_q$ with $0 \leq j \leq (h-2)$. Consider all the products $B_i B_{i+1}$, $1 \leq i \leq (q-1)$, such

that B_i is strictly lower triangular and B_{i+1} is upper triangular; assume that there are p of these products. Write all these products as the sum of two terms, $B_i B_{i+1} = (B_i B_{i+1})_L + (B_i B_{i+1})_U$, and substitute and expand them in the expression for Π_{h-j} . Thus, Π_{h-j} is written as a sum of 2^p terms. Then one of these terms is $\Pi_{h-(j+1)}$.

Proof. 1. It is true by definition for $j = 0$. Assume that it is true for $(j-1)$. $\mathcal{T}_{h-j}(\Pi)$ is obtained from $\mathcal{T}_{h-(j-1)}(\Pi)$ without changing the order between left and right children, and by removing pairs of leaves that are siblings. Thus, $\mathcal{T}_{h-j}(\Pi)$ is again an ordered full binary tree. Concerning the height, let ν be any leaf of $\mathcal{T}_{h-(j-1)}(\Pi)$ such that the path from the root to ν has length $(h-j+1)$. Then the length of the path from the root to the sibling of ν is also $(h-j+1)$. This implies that the sibling of ν is also a leaf; otherwise the height of $\mathcal{T}_{h-(j-1)}(\Pi)$ will be greater than $(h-j+1)$. Therefore, ν and its sibling are not vertices of $\mathcal{T}_{h-j}(\Pi)$, but their parent is.

2. $\mathcal{T}_1(\Pi)$ is an ordered full binary tree with height 1, and it is also a subgraph of $\mathcal{T}(\Pi)$ containing the root of $\mathcal{T}(\Pi)$. This means that $\mathcal{T}_1(\Pi)$ is the tree formed by the root of $\mathcal{T}(\Pi)$ and its two children. From Definition 3.3 it is trivial that $\Pi = \Pi_1$.

3. If B_i and B_{i+1} are represented by two leaves that are siblings, the ordered property of the tree $\mathcal{T}_{h-j}(\Pi)$ implies that B_i is strictly lower triangular and B_{i+1} is upper triangular.

Assume now that B_i is strictly lower triangular and B_{i+1} is upper triangular. Let β and ν be the leaves of $\mathcal{T}_{h-j}(\Pi)$ representing, respectively, B_i and B_{i+1} . The ordering property of $\mathcal{T}_{h-j}(\Pi)$ implies that β is the left child of a parent β' and that ν is the right child of a parent ν' . The right child of β' and the left child of ν' are vertices of $\mathcal{T}_{h-j}(\Pi)$ because this is a full binary tree. Hence, if $\beta' \neq \nu'$, the matrices represented by the leaves descending from the right child of β' and from the left child of ν' should be between B_i and B_{i+1} in Π_{h-j} , but there is nothing between B_i and B_{i+1} . Therefore, $\beta' = \nu'$, and β and ν are siblings.

4. Let $\{B_{i_1} B_{i_1+1}, B_{i_2} B_{i_2+1}, \dots, B_{i_p} B_{i_p+1}\}$, $1 \leq i_1 < i_2 < \dots < i_p \leq (q-1)$, be the set of all the products $B_i B_{i+1}$ with B_i strictly lower triangular, and B_{i+1} upper triangular. After the proposed expansion the summands of Π_{h-j} are all the 2^p possible products $B_1 B_2 \dots B_q$ in which every $B_{i_t} B_{i_t+1}$, $1 \leq t \leq p$, is replaced by $(B_{i_t} B_{i_t+1})_L$ or $(B_{i_t} B_{i_t+1})_U$. The previous item has shown that the leaves of $\mathcal{T}_{h-j}(\Pi)$ representing $\{B_{i_1}, B_{i_1+1}, B_{i_2}, B_{i_2+1}, \dots, B_{i_p}, B_{i_p+1}\}$ are precisely all the pairs of leaves that are siblings. These are the leaves to be removed to get $\mathcal{T}_{h-(j+1)}(\Pi)$. Therefore, $\mathcal{T}_{h-(j+1)}(\Pi)$ has as leaves those of $\mathcal{T}_{h-j}(\Pi)$ which have not been removed, and the parents of those which have been removed. But by Definition 3.3 these parents represent $(B_{i_t} B_{i_t+1})_L$ or $(B_{i_t} B_{i_t+1})_U$, $1 \leq t \leq p$. $\square$

3.2. Proof of Theorem 3.1. The case $k = 1$ is trivial. We will focus on the case $k \geq 2$. By Lemma 3.2 it is enough to prove the result for the series of the LU factors of $I + |F|$. Remember that the terms of these series have been denoted by $\bar{\mathcal{L}}_k$ and $\bar{\mathcal{U}}_k$ according to Lemma 3.2. Let $\Pi^{(1)}, \dots, \Pi^{(r_k)}$ be the summands of $(-1)^{k+1}(\bar{\mathcal{L}}_k + \bar{\mathcal{U}}_k)$ in the sum defined in Theorem 2.2 applied to $|F|$. Let $h_1, \dots, h_{r_k}$ be the heights of the trees $\mathcal{T}(\Pi^{(1)}), \dots, \mathcal{T}(\Pi^{(r_k)})$ defined in Definition 3.3. We write $|F|^k$ as the following sums of matrices of nonnegative elements.

1. Expand $|F|^k = (|F|_L + |F|_U)^k = \sum_{i=1}^{2^k} S_i^{(0)}$ as the sum of all the 2^k different products of k factors $|F|_L$ or $|F|_U$.

2. For $j \geq 1$,

$$(3.5) \quad |F|^k = \sum_{i=1}^{p_j} S_i^{(j)}$$

is obtained from

$$(3.6) \quad |F|^k = \sum_{i=1}^{p_{j-1}} S_i^{(j-1)}$$

as follows: for every summand $S_i^{(j-1)}$, $1 \leq i \leq p_{j-1}$, which is the product of more than two factors, write every pair of consecutive factors of the type $C_L D_U$ as $C_L D_U = (C_L D_U)_L + (C_L D_U)_U$. Substitute and expand these expressions in $S_i^{(j-1)}$, writing the result as a sum of terms. Substitute the new expressions for $S_i^{(j-1)}$ in (3.6) and reenumerate the summands to get (3.5).

In the first place, notice that the set of matrices $\{\Pi_{h_1}^{(1)}, \dots, \Pi_{h_{r_k}}^{(r_k)}\}$ is a subset of the set of summands $\{S_1^{(0)}, \dots, S_{2^k}^{(0)}\}$, because by Definitions 3.3 and 3.4 the matrices $\Pi_{h_i}^{(i)}$, $1 \leq i \leq r_k$, are products of k factors $|F|_L$ or $|F|_U$.¹

Now, we prove the following inductive step: if, for some index i , $\Pi_{h_i-j}^{(i)}$ with $(h_i - j) > 1$ is an element of the set $\{S_1^{(j)}, \dots, S_{p_j}^{(j)}\}$, then $\Pi_{h_i-(j+1)}^{(i)}$ is an element of the set $\{S_1^{(j+1)}, \dots, S_{p_{j+1}}^{(j+1)}\}$. This follows directly from item 4 in Lemma 3.5 and the way in which (3.5) is obtained from (3.6).

Finally, it is obvious that if, for some index i , $\Pi_1^{(i)}$ is an element of the set $\{S_1^{(j)}, \dots, S_{p_j}^{(j)}\}$, then $\Pi_1^{(i)}$ is also an element of the set $\{S_1^{(j+1)}, \dots, S_{p_{j+1}}^{(j+1)}\}$, because by the second item in Lemma 3.5, $\Pi_1^{(i)} = \Pi^{(i)}$, and it is the product of just two factors $C_L D_U$. Therefore, this summand remains unchanged when the set $\{S_1^{(j+1)}, \dots, S_{p_{j+1}}^{(j+1)}\}$ is obtained from $\{S_1^{(j)}, \dots, S_{p_j}^{(j)}\}$.

As a consequence of the previous argument, the set of summands of $(-1)^{k+1}(\bar{\mathcal{L}}_k + \bar{U}_k)$, i.e., $\{\Pi^{(1)}, \dots, \Pi^{(r_k)}\}$, is a subset of the set of summands $\{S_1^{(m)}, \dots, S_{p_m}^{(m)}\}$, where $m = \max\{h_1 - 1, \dots, h_{r_k} - 1\}$. Taking into account that all the matrices $\{\Pi^{(1)}, \dots, \Pi^{(r_k)}\}$ are different and that the matrices $S_1^{(m)}, \dots, S_{p_m}^{(m)}$ have nonnegative entries, then from (3.5), with $j = m$, $(-1)^{k+1}(\bar{\mathcal{L}}_k + \bar{U}_k) = |\bar{\mathcal{L}}_k + \bar{U}_k| \leq |F|^k$. Theorem 3.1 follows from Lemma 3.2.

3.3. A general version of Theorem 3.1. To study the series and perturbation results corresponding to the Cholesky factorization, the factorization coming from the diagonal pivoting method, and block LU factorizations, a more general version of Theorem 3.1 is needed. This is presented in this subsection.

Consider a given $n \times n$ real matrix $[\lambda_{ij}]$ such that $0 \leq \lambda_{ij} \leq 1$ for any (i, j) . For any $n \times n$ matrix $B = [b_{ij}]$, let us define the two $n \times n$ matrices $B_\Lambda = [(B_\Lambda)_{ij}]$ and $B_\Upsilon = [(B_\Upsilon)_{ij}]$ as follows:

$$(3.7) \quad (B_\Lambda)_{ij} = \lambda_{ij} b_{ij}, \quad (B_\Upsilon)_{ij} = (1 - \lambda_{ij}) b_{ij}.$$

Notice that when $\lambda_{ij} = 1$ if $i > j$ and $\lambda_{ij} = 0$ otherwise, the usual strictly lower and upper triangular parts appearing in (2.7) are obtained. Another interesting case that we will use in the context of the Cholesky factorization, when B is Hermitian, is

¹It is interesting to note that it may happen that $\Pi_{h_i}^{(i)} = \Pi_{h_j}^{(j)}$ with $i \neq j$. This happens, for instance, in the second and fourth summands in the right-hand side of (2.13). However, all the summands $S_i^{(0)}$, $1 \leq i \leq 2^k$, are different. This is consistent with the fact that we are speaking about *sets*, because for sets, it does not matter if an element is listed more than once. Our arguments are independent of these details, but they are cumbersome precisely due to the fact that equalities of the kind $\Pi_{h_i}^{(i)} = \Pi_{h_j}^{(j)}$ with $i \neq j$ may appear.

$\lambda_{ij} = 1$ if $i > j$, $\lambda_{ii} = 1/2$, and $\lambda_{ij} = 0$ otherwise. Notice that in this case $B_{\Upsilon} = B_{\Lambda}^*$. Finally, consider that B is *partitioned into blocks* as follows: $B = [B_{lm}]$, $1 \leq l, m \leq p$, where B_{lm} is an $n_l \times n_m$ matrix, and $\sum_{l=1}^p n_l = n$. If we choose $\lambda_{ij} = 1$ in the positions corresponding to the entries of the blocks B_{lm} with $l > m$ and $\lambda_{ij} = 0$ otherwise, then B_{Λ} and B_{Υ} are, respectively, the *block* strictly lower and block upper triangular parts of B .

For any matrix B , the matrices B_{Λ} and B_{Υ} satisfy the properties (1) $B = B_{\Lambda} + B_{\Upsilon}$; (2) $|B_{\Lambda}| = |B|_{\Lambda}$ and $|B_{\Upsilon}| = |B|_{\Upsilon}$; (3) if $|A| \leq |B|$, then $|A|_{\Lambda} \leq |B|_{\Lambda}$ and $|A|_{\Upsilon} \leq |B|_{\Upsilon}$; (4) $|B_{\Lambda}| \leq |B|$ and $|B_{\Upsilon}| \leq |B|$. With all these we get the following.

THEOREM 3.6. *Let F be an $n \times n$ complex matrix and let us define recursively the following matrices:*

$$(3.8) \quad \mathcal{L}_1 = F_{\Lambda}, \mathcal{U}_1 = F_{\Upsilon},$$

$$(3.9) \quad \mathcal{L}_k = (-\mathcal{L}_1 \mathcal{U}_{k-1} - \mathcal{L}_2 \mathcal{U}_{k-2} - \cdots - \mathcal{L}_{k-1} \mathcal{U}_1)_{\Lambda} \quad \text{for } k \geq 2,$$

$$(3.10) \quad \mathcal{U}_k = (-\mathcal{L}_1 \mathcal{U}_{k-1} - \mathcal{L}_2 \mathcal{U}_{k-2} - \cdots - \mathcal{L}_{k-1} \mathcal{U}_1)_{\Upsilon} \quad \text{for } k \geq 2.$$

Then

$$|\mathcal{L}_k + \mathcal{U}_k| \leq |F|^k \quad \text{for } k \geq 1,$$

and, therefore, $|\mathcal{L}_k| \leq (|F|^k)_{\Lambda}$ and $|\mathcal{U}_k| \leq (|F|^k)_{\Upsilon}$.

The proof of Theorem 3.6 follows step by step the proof of Theorem 3.1, just by replacing $(\cdot)_L$ by $(\cdot)_{\Lambda}$ and $(\cdot)_U$ by $(\cdot)_{\Upsilon}$. With these replacements, an analogue of Theorem 2.2 also holds for the matrices defined by the recurrence relation (3.8)–(3.10).

4. Convergence properties for the series of the LU factors and strict perturbation bounds. In this section, the most relevant properties on the convergence of the series of the LU factors are shown. As an application, previously existing strict norm- and componentwise perturbation bounds are proven. The results appearing in this section are direct consequences of Theorem 3.1. We begin with a result for the LU factorization of a perturbation of the identity matrix. The first two items in this theorem also appear in Theorem 2.1. They are reproduced here for the sake of clarity and completeness. Additionally, we present here a new proof of item 2 using only matrix techniques and without any reference to Taylor series. We think that it is interesting from a mathematical point of view.

THEOREM 4.1. *Let F be an $n \times n$ complex matrix, such that $\rho(|F|) < 1$, let $\mathcal{L}_k$ and $\mathcal{U}_k$, $k \geq 1$, be the matrices defined by the recurrence relation (2.8)–(2.10), and let $\mathcal{L}_0 = \mathcal{U}_0 = I$. Then*

1. $I + F$ has a unique LU factorization: $I + F = \mathcal{L}\mathcal{U}$;

2.

$$(4.1) \quad \mathcal{L} = I + \sum_{k=1}^{\infty} \mathcal{L}_k \quad \text{and} \quad \mathcal{U} = I + \sum_{k=1}^{\infty} \mathcal{U}_k,$$

and these series converge absolutely componentwise; i.e., the numerical series corresponding to each entry of $\mathcal{L}$ and $\mathcal{U}$ converges absolutely;

3.

$$(4.2) \quad \left| \mathcal{L} - \sum_{k=0}^N \mathcal{L}_k \right| \leq (|F|^{N+1}(I - |F|)^{-1})_L,$$

$$(4.3) \quad \left| \mathcal{U} - \sum_{k=0}^N \mathcal{U}_k \right| \leq (|F|^{N+1}(I - |F|)^{-1})_U;$$

4. if moreover $\|\cdot\|$ is an absolute and consistent matrix norm and $\|F\| < 1$, then

$$(4.4) \quad \max \left\{ \left\| \mathcal{L} - \sum_{k=0}^N \mathcal{L}_k \right\|, \left\| \mathcal{U} - \sum_{k=0}^N \mathcal{U}_k \right\| \right\} \leq \left\| \mathcal{L} + \mathcal{U} - \sum_{k=0}^N (\mathcal{L}_k + \mathcal{U}_k) \right\| \leq \frac{\|F\|^{N+1}}{1 - \|F\|}.$$

In [24, Theorem 2.1] Stewart presented a normwise bound for $\|\mathcal{L} + \mathcal{U} - I - \mathcal{L}_1 - I - \mathcal{U}_1\|$, i.e., just in the case $N = 1$. Theorem 4.1 improves the result in [24] in several ways: the remainders of *any order* of the series are also bounded *componentwise*. The normwise bound (4.4) holds for $\|F\| < 1$ while the bound in [24] holds only if $\|F\| \leq 1/4$. Finally the bound (4.4) is smaller than the one in [24, Theorem 2.1].²

Proof of Theorem 4.1. 1. This is a consequence of $\rho(|F|) < 1$ and (2.4), which guarantee that all the leading principal submatrices of $I + F$ are nonsingular.

2. We consider the series of the entrywise absolute values; the convergence follows trivially from Theorem 3.1:

$$I + \sum_{k=1}^{\infty} |\mathcal{L}_k| \leq I + \sum_{k=1}^{\infty} (|F|^k)_L = I + \left(\sum_{k=1}^{\infty} |F|^k \right)_L = I + (|F|(I - |F|)^{-1})_L.$$

The last step is a consequence of [18, Lemma 5.6.10, Corollary 5.6.16]. A similar argument holds for the series of $\mathcal{U}$. We have proven that the series in (4.1) converge, but not that they converge to $\mathcal{L}$ and $\mathcal{U}$. To prove this, it suffices to prove that

$$(4.5) \quad \lim_{N \rightarrow \infty} \left(\sum_{k=0}^N \mathcal{L}_k \right) \left(\sum_{j=0}^N \mathcal{U}_j \right) = I + F,$$

and to use the fact that the LU factorization of $I + F$ is unique. To do this notice that

$$(4.6) \quad \left(\sum_{k=0}^N \mathcal{L}_k \right) \left(\sum_{j=0}^N \mathcal{U}_j \right) = \sum_{j=0}^N \sum_{k=0}^j \mathcal{L}_k \mathcal{U}_{j-k} + \sum_{j=N+1}^{2N} \sum_{\substack{k=0 \\ k, (j-k) \leq N}}^j \mathcal{L}_k \mathcal{U}_{j-k}.$$

By using (2.8)–(2.10), the first summand in the right-hand side is just

$$(4.7) \quad \sum_{j=0}^N \sum_{k=0}^j \mathcal{L}_k \mathcal{U}_{j-k} = I + F.$$

By using Lemma 3.2, (2.8)–(2.10), and Theorem 3.1, the second summand in the right-hand side of (4.6) can be bounded as follows:

$$\begin{aligned} \left| \sum_{j=N+1}^{2N} \sum_{\substack{k=0 \\ k, (j-k) \leq N}}^j \mathcal{L}_k \mathcal{U}_{j-k} \right| &= \left| \sum_{j=N+1}^{2N} \sum_{\substack{k=1 \\ k, (j-k) \leq N}}^{j-1} \mathcal{L}_k \mathcal{U}_{j-k} \right| \leq \sum_{j=N+1}^{2N} \sum_{k=1}^{j-1} |\mathcal{L}_k| |\mathcal{U}_{j-k}| \\ &\leq \sum_{j=N+1}^{2N} (-1)^j \sum_{k=1}^{j-1} \bar{\mathcal{L}}_k \bar{\mathcal{U}}_{j-k} = \sum_{j=N+1}^{2N} (-1)^{j+1} (\bar{\mathcal{L}}_j + \bar{\mathcal{U}}_j) \\ &\leq \sum_{j=N+1}^{2N} |F|^j \leq |F|^{N+1} \sum_{j=0}^{\infty} |F|^j = |F|^{N+1} (I - |F|)^{-1}. \end{aligned}$$

²It should be remarked that there is a minor misprint in the statement of Theorem 2.1 in [24]: the bound appearing there has to be multiplied by 2, as follows from the proof of this theorem. This is necessary to see that the bound (4.4) for $N = 1$ is always smaller than the bound in [24].

This means that the second summand in (4.6) goes to zero as N goes to infinity. This and (4.7) finally prove (4.5).

3. We prove only the result for $\mathcal{L}$, i.e., the bound (4.2). The proof for (4.3) is similar. Again, these results are trivial consequences of the bound in Theorem 3.1. Notice that

$$\left| \mathcal{L} - \sum_{k=0}^N \mathcal{L}_k \right| \leq \sum_{k=N+1}^{\infty} |\mathcal{L}_k| \leq \sum_{k=N+1}^{\infty} (|F|^k)_L = \left(\sum_{k=N+1}^{\infty} |F|^k \right)_L = \left(|F|^{N+1} \sum_{k=0}^{\infty} |F|^k \right)_L,$$

and by summing the series, (4.2) is obtained.

4. By using Theorem 3.1 and the fact that the norm $\|\cdot\|$ is absolute,

$$\left\| \mathcal{L} + \mathcal{U} - \sum_{k=0}^N (\mathcal{L}_k + \mathcal{U}_k) \right\| \leq \sum_{k=N+1}^{\infty} \|F\|^k,$$

and by summing the series, (4.4) is obtained. $\square$

As a direct consequence of Theorem 4.1 we get the following theorem for the series and perturbation bounds of the LU factors of a general matrix A .

THEOREM 4.2. *Let the nonsingular $n \times n$ complex matrix A have the LU factorization $A = LU$. Let us consider the matrix $A + E$ and define $F = L^{-1}EU^{-1}$. If $\rho(|F|) < 1$, then*

1. *the matrix $A + E$ is nonsingular and has a unique LU factorization*

$$A + E = L(I + F)U = L\mathcal{L}\mathcal{U}U = \tilde{L}\tilde{U};$$

2. *if $\mathcal{L}_k$ and $\mathcal{U}_k$ are the terms of the series appearing in (4.1), then*

$$(4.8) \quad \tilde{L} = L\mathcal{L} = L \left(I + \sum_{k=1}^{\infty} \mathcal{L}_k \right) \quad \text{and} \quad \tilde{U} = \mathcal{U}U = \left(I + \sum_{k=1}^{\infty} \mathcal{U}_k \right) U;$$

3.

$$(4.9) \quad \|\tilde{L} - L\| \leq \|L\| (|F|(I - |F|)^{-1})_L,$$

$$(4.10) \quad \|\tilde{U} - U\| \leq (|F|(I - |F|)^{-1})_U \|U\|;$$

4. *if moreover $\|\cdot\|$ is an absolute and consistent matrix norm and $\|F\| < 1$, then*

$$(4.11) \quad \max \left\{ \frac{\|\tilde{L} - L\|}{\|L\|}, \frac{\|\tilde{U} - U\|}{\|U\|} \right\} \leq \frac{\|F\|}{1 - \|F\|}.$$

The componentwise bounds (4.9) and (4.10) are essentially the same as those presented in [28, Theorem 5.1], and the normwise bound (4.11) is of the same kind as the one presented in [3, Theorem 3.1]. However, the proofs appearing in [3, 28] are based on different approaches. For the sake of brevity, we do not present bounds on the remainders of any order for the series (4.8). They can be easily deduced from Theorem 4.1. The results in Theorem 4.2 directly imply weaker bounds, under more restrictive conditions, that may be more useful in practice; see [28, Theorem 5.2].

The normwise bounds (4.11) can systematically overestimate the actual errors [9]; however, some inner diagonal scalings can be introduced in the normwise perturbation theory of the LU factorization to improve the bounds and to estimate the true

normwise condition numbers of the LU factors. We will further discuss these questions after Theorem 5.1.

Proof of Theorem 4.2. 1. The proof of this item follows from the first item in Theorem 4.1, and (2.2). See also the comments following (2.2).

2. This follows from (2.2) and the second item in Theorem 4.1. Obviously, if a convergent series of matrices is multiplied by a constant matrix, then we get a series which converges to the product of the former series times the constant matrix.

3. This is a consequence of (4.8), (4.2), and (4.3) for the case $N = 0$.

4. This follows from (4.8) and (4.4) in the case $N = 0$. $\square$

This theorem can be used in practice in combination with the backward error analysis for the LU factorization. The classical backward error result (see, for instance, [17, Theorem 9.3]) states that the computed LU factors, $\hat{L}$, $\hat{U}$, are the exact LU factors of a matrix $A + E$, $A + E = \hat{L}\hat{U}$ with $|E| \leq (n\epsilon/(1 - n\epsilon))|\hat{L}||\hat{U}|$, where ϵ is the unit roundoff of the computer. Combining this backward error bound with the perturbation bounds appearing in (4.9)–(4.11) (interchanging the roles of A and $A + E$, and the roles of L , U and $\hat{L}$, $\hat{U}$), bounds on the forward error bounds of the computed LU factors can be obtained [7].

5. Perturbation theory for block LU factorization. In this section, we deal with the block LU factorization of a nonsingular square matrix A . The main result in this section is that the perturbation theory for the block LU factorization is completely analogous to the perturbation theory of the usual LU factorization, just by replacing pointwise by block lower and upper triangular matrices.

The stability properties and applications of block LU factorizations appear in [17, Chapter 13] and the references therein. A block LU factorization of an $n \times n$ complex matrix A is as follows:

$$(5.1) \quad \underbrace{\begin{bmatrix} A_{11} & A_{12} & \cdots & A_{1p} \\ A_{21} & A_{22} & \cdots & A_{2p} \\ \vdots & \vdots & \ddots & \vdots \\ A_{p1} & A_{p2} & \cdots & A_{pp} \end{bmatrix}}_A = \underbrace{\begin{bmatrix} I & & & \\ L_{21} & I & & \\ \vdots & & \ddots & \\ L_{p1} & \cdots & L_{p,p-1} & I \end{bmatrix}}_L \underbrace{\begin{bmatrix} U_{11} & U_{12} & \cdots & U_{1p} \\ & U_{22} & & \vdots \\ & & \ddots & U_{p-1,p} \\ & & & U_{pp} \end{bmatrix}}_U,$$

where the matrices L_{ij} , U_{ij} , and A_{ij} have dimensions $n_i \times n_j$, and $\sum_{i=1}^p n_i = n$. These dimensions are a priori given numbers which define the exact meaning of the *block* LU factorization. It is assumed that these dimensions remain unchanged throughout this section. We have chosen, as usual, to keep the same notation for block and pointwise LU factorizations. Therefore, here L and U have a different meaning than that in the previous sections. The reader should be careful about this question.

The block LU factorization and block triangular matrices have analogous properties to those of the usual LU factorization and triangular matrices: if the p leading block principal submatrices of A are nonsingular, then A has a unique block LU factorization; if a block LU factorization of a nonsingular matrix exists, then it is unique (see [17, Theorem 13.2] for a more general result covering the case of singular matrices). Notice that according to the previous properties, it may happen that a matrix has a unique block LU factorization with blocks of given dimensions, but not a usual pointwise LU factorization. However, if a matrix has a unique pointwise LU factorization, then it also has a unique block LU factorization for any possible block dimensions.

The product of two upper (lower) block triangular matrices is an upper (lower) block triangular matrix with the same block dimensions. Finally, the inverse of an upper (lower) block triangular matrix is also an upper (lower) block triangular matrix with the same block dimensions. Moreover, the diagonal blocks of the inverse are the inverse matrices of the diagonal blocks of the matrix. All these properties can easily be proven.

Now, we can proceed with a similar argument to that used to study the series and perturbation theory of the LU factorization. Let $A = LU$ be the unique block LU factorization of the $n \times n$ nonsingular matrix A . Let us consider a perturbation $\tilde{A} = A + E$ of A , such that $\rho(|L^{-1}EU^{-1}|) < 1$. Then

$$(5.2) \quad \tilde{A} = L(I + L^{-1}EU^{-1})U = (L\mathcal{L})(\mathcal{U}U) \equiv \tilde{L}\tilde{U},$$

where $(I + L^{-1}EU^{-1}) = \mathcal{L}\mathcal{U}$ is the unique block LU factorization of $(I + L^{-1}EU^{-1})$ with the same block dimensions as those of $A = LU$. Therefore, $\tilde{A} = (L\mathcal{L})(\mathcal{U}U) \equiv \tilde{L}\tilde{U}$ is the unique block LU factorization of the nonsingular matrix $\tilde{A}$ with the same block dimensions as those of $A = LU$. The condition $\rho(|L^{-1}EU^{-1}|) < 1$, along with (2.4), guarantees that all these block factorizations exist and are unique. These results are formally the same as those appearing in (2.2)–(2.4), but here $L, U, \mathcal{L}$, and $\mathcal{U}$ are block triangular matrices; thus the results and the condition $\rho(|L^{-1}EU^{-1}|) < 1$ depend on the block structure we are considering.

Given a block partitioned matrix A , such as the one appearing in (5.1), let us define its *block* strictly lower triangular and *block* upper triangular parts, A_L and A_U , respectively, in the obvious way:

$$(5.3) \quad \underbrace{\begin{bmatrix} A_{11} & \cdots & A_{1p} \\ A_{21} & \cdots & A_{2p} \\ \vdots & \vdots & \vdots \\ A_{p1} & \cdots & A_{pp} \end{bmatrix}}_A = \underbrace{\begin{bmatrix} 0 & & & \\ A_{21} & 0 & & \\ \vdots & & \ddots & \\ A_{p1} & \cdots & A_{p,p-1} & 0 \end{bmatrix}}_{A_L} + \underbrace{\begin{bmatrix} A_{11} & A_{12} & \cdots & A_{1p} \\ & A_{22} & & \vdots \\ & & \ddots & A_{p-1,p} \\ & & & A_{pp} \end{bmatrix}}_{A_U},$$

where for the sake of simplicity the same notation as in the case of pointwise strictly lower and upper triangular parts is used. Let us remember that for block strictly lower and block upper triangular parts a block analogue of Theorem 3.1 holds, as we have discussed in the general Theorem 3.6. With this, a block version of Theorem 4.1 can be proved in an analogous way. To keep the paper concise, we will not state this block theorem. This theorem allows us to prove the main result of this section.

THEOREM 5.1. *Let the nonsingular $n \times n$ complex matrix A have the block LU factorization $A = LU$ appearing in (5.1). For any partitioned matrix B , let us denote by B_L and B_U , respectively, its block strictly lower triangular and block upper triangular parts, as defined in (5.3). Let us consider the matrix $A + E$ and define $F = L^{-1}EU^{-1}$. If $\rho(|F|) < 1$, then*

1. *the matrix $A + E$ is nonsingular and has a unique block LU factorization, $A + E = \tilde{L}\tilde{U}$, with the same block dimensions as those in $A = LU$;*
2. *let $\mathcal{L}_k$ and $\mathcal{U}_k$ be the matrices defined recursively by*

$$\begin{aligned} \mathcal{L}_1 &= F_L, \quad \mathcal{U}_1 = F_U, \\ \mathcal{L}_k &= (-\mathcal{L}_1\mathcal{U}_{k-1} - \mathcal{L}_2\mathcal{U}_{k-2} - \cdots - \mathcal{L}_{k-1}\mathcal{U}_1)_L \quad \text{for } k \geq 2, \\ \mathcal{U}_k &= (-\mathcal{L}_1\mathcal{U}_{k-1} - \mathcal{L}_2\mathcal{U}_{k-2} - \cdots - \mathcal{L}_{k-1}\mathcal{U}_1)_U \quad \text{for } k \geq 2; \end{aligned}$$

then with $\mathcal{L}$ and $\mathcal{U}$ defined below, giving $\mathcal{LU} = I + F$,

$$(5.4) \quad \tilde{L} = L \mathcal{L} = L \left(I + \sum_{k=1}^{\infty} \mathcal{L}_k \right) \quad \text{and} \quad \tilde{U} = \mathcal{U} U = \left(I + \sum_{k=1}^{\infty} \mathcal{U}_k \right) U;$$

3.

$$(5.5) \quad \left| \tilde{L} - L \right| \leq |L| (|F| (I - |F|)^{-1})_L,$$

$$(5.6) \quad \left| \tilde{U} - U \right| \leq (|F| (I - |F|)^{-1})_U |U|;$$

4. if moreover $\|\cdot\|$ is an absolute and consistent matrix norm and $\|F\| < 1$, then

$$(5.7) \quad \max \left\{ \frac{\|\tilde{L} - L\|}{\|L\|}, \frac{\|\tilde{U} - U\|}{\|U\|} \right\} \leq \frac{\|F\|}{1 - \|F\|}.$$

This theorem and its proof are completely analogous to the proof of Theorem 4.2.

The perturbation theorem above has to be combined with backward error results to estimate the forward errors produced by the algorithms that compute the block LU factorization. If this factorization is computed using conventional matrix operations, then it is possible to prove component- and normwise backward error bounds. However, in the case of block algorithms, fast matrix multiplication techniques are frequently used to improve the performance, and then the backward error bounds are necessarily normwise [11] (see also [17, Theorem 13.6]). In this case, the bound (5.7) can systematically overestimate the actual errors, but, at the cost of discarding second-order terms, this bound can be improved. From (5.4), using again $F = L^{-1}EU^{-1}$,

$$\tilde{L} - L = L(L^{-1}EU^{-1})_L + \|L\| O(\|F\|^2) = (LS)((LS)^{-1}EU^{-1})_L + \|L\| O(\|F\|^2),$$

where S is any block diagonal matrix whose blocks have the same dimensions as the diagonal blocks of A . Let us denote by $\mathcal{S}$ the set of block diagonal matrices with blocks of these dimensions and by $\kappa(B) = \|B\|\|B^{-1}\|$ the condition number, in the norm we are considering, of the matrix B . Then, the previous equation implies

$$\frac{\|\tilde{L} - L\|}{\|L\|} \leq \min_{S \in \mathcal{S}} \kappa(LS) \frac{\|U^{-1}\|\|E\|}{\|L\|} + O(\|F\|^2) \leq \left(\min_{S \in \mathcal{S}} \kappa(LS) \right) \kappa(U) \frac{\|E\|}{\|A\|} + O(\|F\|^2).$$

A similar argument for U implies that

$$\frac{\|\tilde{U} - U\|}{\|U\|} \leq \min_{S \in \mathcal{S}} \kappa(SU) \frac{\|L^{-1}\|\|E\|}{\|U\|} + O(\|F\|^2) \leq \left(\min_{S \in \mathcal{S}} \kappa(SU) \right) \kappa(L) \frac{\|E\|}{\|A\|} + O(\|F\|^2).$$

For the usual LU factorization the set $\mathcal{S}$ is just the set of diagonal matrices, and $\min_{S \in \mathcal{S}} \kappa(LS)$ (resp., $\min_{S \in \mathcal{S}} \kappa(SU)$) can be reliably estimated, for p norms, by choosing S in a way such that all the columns (resp., rows) of LS (resp., SU) have unit norms [29]. Moreover, $\min_{S \in \mathcal{S}} \kappa(LS)\kappa(U)$ and $\min_{S \in \mathcal{S}} \kappa(SU)\kappa(L)$ are, respectively, good approximations of the true normwise condition numbers of the L and U factors [9, 25]. The true normwise condition numbers were obtained in [9], but it is difficult to estimate them efficiently. In the case of block LU factorization, the set $\mathcal{S}$ contains all the diagonal matrices; then $\min_{S \in \mathcal{S}} \kappa(LS)$ and $\min_{S \in \mathcal{S}} \kappa(SU)$ are smaller than for the usual pointwise LU factorization. But unfortunately there is no easy way to estimate these optimally block scaled condition numbers [21].

6. Perturbation theory for block LDL* factorization of general Hermitian matrices. The block LU factorization plays a relevant role in the solution of linear systems of equations with a Hermitian indefinite coefficient matrix B [17, section 11.1]. In this case, the most widely used approach is to perform a block factorization as follows:

$$PBP^T = LDL^*,$$

where P is a permutation matrix, L is unit lower triangular, and D is block diagonal with diagonal blocks of dimension 1×1 or 2×2 . The 2×2 diagonal blocks of D are Hermitian indefinite matrices. Moreover, the diagonal blocks of L corresponding to the 2×2 blocks of D are 2×2 identity matrices. This factorization method is usually called the *diagonal pivoting method*. Three well-known pivoting strategies to choose the permutation matrix P are available: complete pivoting [6], partial pivoting [5], and rook pivoting [2]. The partial pivoting strategy is available in LAPACK [1], and it is used in the LAPACK symmetric indefinite linear equation solver.

The block LDL* factorization of a nonsingular Hermitian matrix is a special case of the block LU factorization, as the following lemma shows.³

LEMMA 6.1. *Let A be a nonsingular Hermitian matrix having a block LU factorization given by (5.1), $A = LU$; then the block diagonal matrix $\Delta = \text{diag}(U_{11}, U_{22}, \dots, U_{pp})$ is Hermitian and $L^* = \Delta^{-1}U$. Therefore, $A = L\Delta L^*$ and this factorization is unique once the block dimensions are fixed.*

Proof. Let us write $A = LU = L\Delta(\Delta^{-1}U)$ and—using the fact that $A = A^*$ — $A = (\Delta^{-1}U)^*(L\Delta)^*$. The block LU factorization of a nonsingular matrix is unique; therefore, $L = (\Delta^{-1}U)^*$ and $A = L\Delta L^*$, which implies that Δ is Hermitian. The uniqueness of $A = L\Delta L^*$ follows from the uniqueness of the block LU factorization. $\square$

Therefore, the conditions for the existence of an LDL* factorization of a nonsingular Hermitian matrix are again that all the corresponding leading principal block submatrices of A are nonsingular. Moreover, to guarantee that the 2×2 diagonal blocks of D are indefinite matrices, the inertias of the leading principal block submatrices of A have to follow a special pattern with respect to the 2×2 blocks. This is the case whenever $A = PBP^T$, and P is obtained by applying some of the previously mentioned pivoting strategies.

The following shorthand notation will be frequently used: let $B = [B_{ij}]$, $1 \leq i, j \leq p$, be a block partitioned matrix, where B_{ij} is an $n_i \times n_j$ matrix. B_D will denote the block diagonal part of B , i.e., $B_D = \text{diag}(B_{11}, \dots, B_{pp})$.

Since the LDL* factorization is a block LU factorization,⁴ its perturbation theory can be obtained from Theorem 5.1 by using that in this case $U = DL^*$.

THEOREM 6.2. *Let the nonsingular $n \times n$ Hermitian matrix A have the block LDL* factorization $A = LDL^*$. Let us denote for any partitioned matrix B by B_L and B_D its block strictly lower and block diagonal parts, respectively. Let us consider the Hermitian matrix $A + E$ and define $F = L^{-1}EL^{-*}D^{-1}$. If $\rho(|F|) < 1$, then the following results hold.*

³In Lemma 6.1, the letter Δ is used instead of D because the dimensions of the blocks in Δ are not necessarily 1×1 or 2×2 .

⁴In this section, we restrict ourselves to LDL* factorizations with blocks of dimensions 1×1 or 2×2 , because this is the most useful situation in numerical analysis. However, the results we present can be extended without effort to general block dimensions.

1. The matrix $A + E$ is nonsingular and has a unique block LDL^* factorization, $A + E = \tilde{L}\tilde{D}\tilde{L}^*$, with the same block dimensions as those in $A = LDL^*$. Moreover, let us denote the block diagonal matrices D and $\tilde{D}$ as follows: $D = \text{diag}(D_{11}, \dots, D_{pp})$ and $\tilde{D} = \text{diag}(\tilde{D}_{11}, \dots, \tilde{D}_{pp})$. Then, if D_{ii} and $\tilde{D}_{ii}$ are 1×1 blocks, both have the same sign, and if D_{ii} and $\tilde{D}_{ii}$ are 2×2 blocks, both are Hermitian indefinite matrices.

2.

$$(6.1) \quad |\tilde{L} - L| \leq |L| (|F| (I - |F|)^{-1})_L,$$

$$(6.2) \quad |\tilde{D} - D| \leq (|F| (I - |F|)^{-1})_D |D|.$$

3. If moreover $\|\cdot\|$ is an absolute and consistent matrix norm and $\|F\| < 1$, then

$$(6.3) \quad \max \left\{ \frac{\|\tilde{L} - L\|}{\|L\|}, \frac{\|\tilde{D} - D\|}{\|D\|} \right\} \leq \frac{\|F\|}{1 - \|F\|}.$$

Proof. 1. We only have to prove the properties of the diagonal blocks of D and $\tilde{D}$; the rest of this item follows from the first item of Theorem 5.1. We will use a continuity argument. Let us consider the one-parametric family of matrices $A(t) = A + tE$, where $0 \leq t \leq 1$. Notice that $\rho(|L^{-1}(tE)L^{-*}D^{-1}|) = t\rho(|L^{-1}EL^{-*}D^{-1}|) < 1$, and therefore, $A(t)$ is nonsingular and has a unique LDL^* factorization for each t with the same block dimensions as those in $A = LDL^*$. Let us denote this factorization $A(t) = L(t)D(t)L(t)^*$ for $0 \leq t \leq 1$. Clearly $L(t)$ and $D(t)$ are continuous functions of $t \in [0, 1]$. This follows, for instance, from (5.5) and (5.6), taking into account that D is just the block diagonal part of U . On the other hand, $\det A(t) = \det D_{11}(t) \cdots \det D_{pp}(t) \neq 0$ for all $t \in [0, 1]$. Therefore, if $D_{ii}(t)$ is a 1×1 block, then its sign remains constant for $t \in [0, 1]$, and if $D_{ii}(t)$ is a 2×2 block, then the sign of $\det D_{ii}(t)$ also remains constant for $t \in [0, 1]$. This implies that if the two eigenvalues of $D_{ii} = D_{ii}(0)$ have opposite signs, the two eigenvalues of $\tilde{D}_{ii} = D_{ii}(1)$ also have opposite signs.

2. The bound (6.1) is just (5.5), and the bound (6.2) follows from considering the block diagonal part of (5.6).

3. The bound for L is the same as in (5.7). The bound for D follows directly from the second equation in (5.4) and, from Theorem 3.6, that $|\mathcal{U}_k)_D| \leq |\mathcal{U}_k| \leq |F|^k$. $\square$

The normwise bound (6.3) can be improved using inner block diagonal scalings in the same way as was done for the block LU factorization after Theorem 5.1.

Notice that the componentwise bound (6.2) is not a symmetric matrix. This is a drawback of the previous theorem because $|\tilde{D} - D|$ is symmetric. The following theorem partially solves this drawback by splitting the bound on $|\tilde{D} - D|$ into two terms: a symmetric first-order term, by far the most relevant in applications, and a second-order term that is not symmetric.

THEOREM 6.3. *Using the same assumptions and notation as those of Theorem 6.2, we obtain*

$$|\tilde{D} - D| \leq |L^{-1}EL^{-*}|_D + (|F|^2(I - |F|)^{-1})_D |D|.$$

Proof. From (5.4), $\tilde{U} - U = \mathcal{U}_1 U + \sum_{k=2}^{\infty} \mathcal{U}_k U$. Its block diagonal part is

$$(6.4) \quad \tilde{D} - D = F_D D + \sum_{k=2}^{\infty} (\mathcal{U}_k)_D D,$$

where we have used the fact that $\mathcal{U}_1 = F_U$, $U_D = D$, and $(\mathcal{U}_k U)_D = (\mathcal{U}_k)_D D$, because $\mathcal{U}_k$ and U are both upper block triangular matrices. Notice that $F_D D = (FD)_D = (L^{-1}EL^{-*}D^{-1}D)_D = (L^{-1}EL^{-*})_D$. Therefore, (6.4) and Theorem 3.6 imply

$$\begin{aligned} |\tilde{D} - D| &\leq |L^{-1}EL^{-*}|_D + \sum_{k=2}^{\infty} |\mathcal{U}_k|_D |D| \leq |L^{-1}EL^{-*}|_D + \sum_{k=2}^{\infty} (|F|^k)_D |D| \\ &= |L^{-1}EL^{-*}|_D + \left(\sum_{k=2}^{\infty} |F|^k \right)_D |D| = |L^{-1}EL^{-*}|_D + (|F|^2(I - |F|)^{-1})_D |D|. \quad \square \end{aligned}$$

Theorems 6.2 and 6.3 can be used in connection with the backward error results for the usual algorithm to compute the LDL^T factorization [16] (see also [17, Theorem 11.3]) to get bounds for the forward errors of the computed L and D . In this context, it is convenient to interchange the roles of A and $A + E$ in the bounds, because the computed LDL^* factorization is the exact factorization of $A + E$.

In some cases, the D factor of a block LDL^* factorization is just a diagonal matrix, i.e., when the chosen pivoting strategy does not find any 2×2 pivot. One trivial instance of this situation is when the Hermitian matrix A is positive definite, but it may also occur for indefinite matrices. The componentwise bound for $|\tilde{D} - D|$ appearing in Theorem 6.2 can be simplified. If D is diagonal, then $|FD| = |F||D|$ and, therefore,

$$(6.5) \quad |\tilde{D} - D| \leq \text{diag}((I - |F|)^{-1}|L^{-1}EL^{-*}|),$$

where $\text{diag}(B)$ is the diagonal matrix whose entries are those of the diagonal of B .

Although Theorem 6.2 is the first perturbation result for the *block* LDL^* factorization, the case when D is diagonal has been studied by other authors. In [28, Theorem 3.1] componentwise perturbation bounds for the LDL^* factorization of positive definite Hermitian matrices are presented. Theorem 6.2, along with (6.5), represents an improvement over Theorem 3.1 in [28], because Theorem 6.2 is valid also for non-positive diagonal matrices D , it requires less restrictive assumptions concerning the magnitude of the perturbation E , and, moreover, the bounds for L are smaller and simpler. In [22, Theorem 4.1, p. 269] some of the results in [27] for positive definite matrices have been extended to indefinite matrices with factorization LDL^* with D still diagonal, i.e., without 2×2 blocks.

7. Perturbation bounds for the Cholesky factorization. In this section, we present a series expansion and component- and normwise perturbation bounds for the Cholesky factorization of a Hermitian positive definite matrix. The perturbation bounds we present are essentially the same as the bounds in [26, 28] with some improvements. The main goal of this section is to see how the series expansions we have developed allow us to get perturbation bounds also for the Cholesky factorization.

Let A be a Hermitian positive definite matrix with the Cholesky factorization $A = R^*R$, and let E be Hermitian such that $\rho(|R^{-*}ER^{-1}|) < 1$. Then

$$\tilde{A} = A + E = R^*(I + R^{-*}ER^{-1})R$$

is positive definite, because $(I + R^{-*}ER^{-1})$ is positive definite. In addition, if $(I + R^{-*}ER^{-1}) = \mathcal{R}^*\mathcal{R}$ is the corresponding Cholesky factorization, then

$$\tilde{A} = (R^*\mathcal{R}^*)(\mathcal{R}R) \equiv \tilde{R}^*\tilde{R}$$

is the Cholesky factorization of $\tilde{A}$. Thus, let us focus on the Cholesky factorization of $(I + R^{-*}ER^{-1}) \equiv I + F$.

Given a Hermitian $n \times n$ matrix $B = [b_{ij}]$, we define B_{Λ_H} and $B_{\Upsilon_H} = B_{\Lambda_H}^*$:

$$(7.1) \quad (B_{\Lambda_H})_{ij} = \begin{cases} b_{ij} & \text{if } i > j, \\ b_{ii}/2 & \text{if } i = j, \\ 0 & \text{otherwise,} \end{cases} \quad (B_{\Upsilon_H})_{ij} = \begin{cases} b_{ij} & \text{if } i < j, \\ b_{ii}/2 & \text{if } i = j, \\ 0 & \text{otherwise.} \end{cases}$$

If Theorem 3.6 is applied to a Hermitian matrix F , for instance, $F = R^{-*}ER^{-1}$, with $(\cdot)_{\Lambda} = (\cdot)_{\Lambda_H}$ and $(\cdot)_{\Upsilon} = (\cdot)_{\Upsilon_H}$, then a simple argument shows that $\mathcal{L}_k + \mathcal{U}_k$ is Hermitian and that $\mathcal{L}_k = \mathcal{U}_k^*$. Using these facts and introducing the more convenient notation $\mathcal{R}_k$ instead of $\mathcal{U}_k$, we get the following Hermitian version of Theorem 3.6.

THEOREM 7.1. *Let F be an $n \times n$ Hermitian matrix and let us define recursively the following upper triangular matrices:*

$$(7.2) \quad \mathcal{R}_1 = F_{\Upsilon_H}; \quad \mathcal{R}_k = (-\mathcal{R}_1^* \mathcal{R}_{k-1} - \mathcal{R}_2^* \mathcal{R}_{k-2} - \cdots - \mathcal{R}_{k-1}^* \mathcal{R}_1)_{\Upsilon_H} \quad \text{for } k \geq 2.$$

Then $|\mathcal{R}_k^* + \mathcal{R}_k| \leq |F|^k$ for $k \geq 1$, and therefore, $|\mathcal{R}_k| \leq (|F|)^k_{\Upsilon_H}$.

The result above allows us to prove the next theorem very easily.

THEOREM 7.2. *Let the Hermitian positive definite matrix A have the Cholesky factorization $A = R^*R$. Let us consider the Hermitian matrix $A + E$ and define $F = R^{-*}ER^{-1}$. If $\rho(|F|) < 1$, then*

1. *the matrix $A + E$ is positive definite and has a unique Cholesky factorization $A + E = \tilde{R}^*\tilde{R}$;*

2. *if $\mathcal{R}_k$ are the matrices defined in (7.2), then*

$$(7.3) \quad \tilde{R} = \left(I + \sum_{k=1}^{\infty} \mathcal{R}_k \right) R;$$

3.

$$(7.4) \quad |\tilde{R} - R| \leq (|F|(I - |F|)^{-1})_{\Upsilon_H} |R|;$$

4. *let us denote by $\|\cdot\|_F$ and by $\|\cdot\|_2$ the Frobenius and the spectral norms, respectively; if $\|F\|_F < 1$, then*

$$(7.5) \quad \frac{\|\tilde{R} - R\|_F}{\|R\|_2} \leq \frac{1}{\sqrt{2}} \frac{\|F\|_F}{1 - \|F\|_F};$$

if, moreover, $\|A^{-1}\|_2 \|E\|_F < 1$ and $\kappa_2(A) = \|A\|_2 \|A^{-1}\|_2$, then

$$(7.6) \quad \frac{\|\tilde{R} - R\|_F}{\|R\|_2} \leq \frac{1}{\sqrt{2}} \frac{\kappa_2(A) \frac{\|E\|_F}{\|A\|_2}}{1 - \kappa_2(A) \frac{\|E\|_F}{\|A\|_2}}.$$

The componentwise bound (7.4) is slightly better than the one in [28, Theorem 2.1], because there the upper triangular part $(|F|(I - |F|)^{-1})_U$ appears, which is greater than $(|F|(I - |F|)^{-1})_{\Upsilon_H}$. However, the normwise bound in [26, Theorem 1.4] is slightly better than (7.6), because the bound in Theorem 7.1 involves the entrywise absolute value, and the spectral norm is not absolute. The bound (7.5) is new. The normwise bound (7.5) can be improved using inner diagonal scalings as was done for

the block LU factorization after Theorem 5.1. Bounds of the type (7.5) involving only the spectral norm can be found in [13, 14] for small perturbations.

Proof of Theorem 7.2. With the exception of the last item, the proof is similar to the proof of Theorem 4.2. From (7.3), (7.5) is easily proved by realizing that for any Hermitian matrix B , $\|B_{\Upsilon_H}\|_F \leq \frac{1}{\sqrt{2}}\|B\|_F$. $\square$

REFERENCES

- [1] E. ANDERSON, Z. BAI, C. BISCHOF, S. BLACKFORD, J. DEMMEL, J. DONGARRA, J. DU CROZ, A. GREENBAUM, S. HAMMARLING, A. MCKENNEY, AND D. SORESENSEN, *LAPACK Users' Guide*, 3rd ed., SIAM, Philadelphia, 1999.
- [2] C. ASHCRAFT, R. G. GRIMES, AND J. G. LEWIS, *Accurate symmetric indefinite linear equation solvers*, SIAM J. Matrix Anal. Appl., 20 (1998), pp. 513–561.
- [3] A. BARRLUND, *Perturbation bounds for the LDL^H and LU decompositions*, BIT, 31 (1991), pp. 358–363.
- [4] M. I. BUENO AND F. M. DOPICO, *Stability and sensitivity of tridiagonal LU factorization without pivoting*, BIT, 44 (2004), pp. 651–673.
- [5] J. R. BUNCH AND L. KAUFMAN, *Some stable methods for calculating inertia and solving symmetric linear systems*, Math. Comp., 31 (1977), pp. 163–179.
- [6] J. R. BUNCH AND B. N. PARLETT, *Direct methods for solving symmetric indefinite systems of linear equations*, SIAM J. Numer. Anal., 8 (1971), pp. 639–655.
- [7] X.-W. CHANG, *Some features of Gaussian elimination with rook pivoting*, BIT, 42 (2002), pp. 66–83.
- [8] X.-W. CHANG, C. C. PAIGE, AND G. W. STEWART, *New perturbation analyses for the Cholesky factorization*, IMA J. Numer. Anal., 16 (1996), pp. 457–484.
- [9] X.-W. CHANG AND C. C. PAIGE, *On the sensitivity of the LU factorization*, BIT, 38 (1998), pp. 486–501.
- [10] X.-W. CHANG AND C. C. PAIGE, *Sensitivity analyses for factorizations of sparse or structured matrices*, Linear Algebra Appl., 284 (1998), pp. 53–71.
- [11] J. W. DEMMEL, N. J. HIGHAM, AND R. S. SCHREIBER, *Stability of block LU factorization*, Numer. Linear Algebra Appl., 2 (1995), pp. 173–190.
- [12] J. DEMMEL, M. GU, S. EISENSTAT, I. SLAPNIČAR, K. VESELIĆ, AND Z. DRMAČ, *Computing the singular value decomposition with high relative accuracy*, Linear Algebra Appl., 299 (1999), pp. 21–80.
- [13] Z. DRMAČ, M. OMLADIĆ, AND K. VESELIĆ, *On the perturbation of the Cholesky factorization*, SIAM J. Matrix Anal. Appl., 15 (1994), pp. 1319–1332.
- [14] A. EDELMAN AND W. F. MASCARENHAS, *On Parlett's matrix norm inequality for the Cholesky decomposition*, Numer. Linear Algebra Appl., 2 (1995), pp. 243–250.
- [15] G. H. GOLUB AND C. F. VAN LOAN, *Matrix Computations*, 3rd ed., The Johns Hopkins University Press, Baltimore, MD, 1996.
- [16] N. J. HIGHAM, *Stability of the diagonal pivoting method with partial pivoting*, SIAM J. Matrix Anal. Appl., 18 (1997), pp. 52–65.
- [17] N. J. HIGHAM, *Accuracy and Stability of Numerical Algorithms*, 2nd ed., SIAM, Philadelphia, 2002.
- [18] R. A. HORN AND C. R. JOHNSON, *Matrix Analysis*, Cambridge University Press, Cambridge, UK, 1990.
- [19] B. N. PARLETT, *The new qd algorithms*, in Acta Numerica 1995, Acta. Numer., Cambridge University Press, Cambridge, UK, 1995, pp. 459–491.
- [20] K. H. ROSEN, *Discrete Mathematics and Its Applications*, 4th ed., McGraw-Hill, Boston, 1999.
- [21] A. SHAPIRO, *Optimal block diagonal l_2 -scaling of matrices*, SIAM J. Numer. Anal., 22 (1985), pp. 81–94.
- [22] I. SLAPNIČAR, *Componentwise analysis of direct factorization of real symmetric and Hermitian matrices*, Linear Algebra Appl., 272 (1998), pp. 227–275.
- [23] G. W. STEWART, *Perturbation bounds for the QR factorization of a matrix*, SIAM J. Numer. Anal., 14 (1977), pp. 509–518.
- [24] G. W. STEWART, *On the perturbation of LU, Cholesky, and QR factorizations*, SIAM J. Matrix Anal. Appl., 14 (1993), pp. 1141–1145.
- [25] G. W. STEWART, *On the perturbation of LU and Cholesky factors*, IMA J. Numer. Anal., 17 (1997), pp. 1–6.

- [26] J.-G. SUN, *Perturbation bounds for the Cholesky and QR factorizations*, BIT, 31 (1991), pp. 341–352.
- [27] J.-G. SUN, *Rounding-error and perturbation bounds for the Cholesky and LDL^T factorizations*, Linear Algebra Appl., 173 (1992), pp. 77–97.
- [28] J.-G. SUN, *Componentwise perturbation bounds for some matrix decompositions*, BIT, 32 (1992), pp. 702–714.
- [29] A. VAN DER SLUIS, *Condition numbers and equilibration of matrices*, Numer. Math., 14 (1969), pp. 14–23.

THE GENERALIZED EIGENVALUE PROBLEM FOR NONSQUARE PENCILS USING A MINIMAL PERTURBATION APPROACH*

GREGORY BOUTRY[†], MICHAEL ELAD[‡], GENE H. GOLUB[†], AND
PEYMAN MILANFAR[§]

Abstract. This work focuses on nonsquare matrix pencils $A - \lambda B$, where $A, B \in \mathcal{M}^{m \times n}$ and $m > n$. Traditional methods for solving such nonsquare generalized eigenvalue problems $(A - \lambda B)\underline{v} = \underline{0}$ are expected to lead to no solutions in most cases. In this paper we propose a different treatment: We search for the minimal perturbation to the pair (A, B) such that these solutions are indeed possible. Two cases are considered and analyzed: (i) the case when $n = 1$ (vector pencils); and (ii) more generally, the $n > 1$ case with the existence of one eigenpair. For both, this paper proposes insight into the characteristics of the described problems along with practical numerical algorithms toward their solution. We also present a simplifying factorization for such nonsquare pencils, and some relations to the notion of pseudospectra.

Key words. nonsquare pencils, generalized eigenvalue, pseudospectra

AMS subject classifications. 15A18, 15A22, 65F10

DOI. 10.1137/S0895479803428795

1. Introduction.

1.1. Pencils and nonsquare pencils. The term “pencil” used in linear algebra is well known and refers to the expression $A - \lambda B$, where both A and B are typically square $n \times n$ matrices, and λ is a complex scalar. Of special interest are the values that reduce the pencil rank, namely, the λ values satisfying $(A - \lambda B)\underline{v} = \underline{0}$ for some nonzero vector $\underline{v}$. Then the problem is defined as the generalized eigenvalue problem, where the scalars λ are the generalized eigenvalues, and their corresponding vectors $\underline{v}$ are the generalized eigenvectors.¹

In recent years several different fields of research have led to the generalized eigenvalue problem involving a nonsquare pencil, where A and B are rectangular $m \times n$ matrices and $m > n$ [2, 12, 22, 25]. In many of these applications, the pencil involved becomes rectangular due to additional measurements—each new set of measurements constructs a new row in the pencil. Clearly, more measurements (and thus more rows) imply better possible estimation, though it is not clear how this additional information can be exploited. Also, in most applications the content of the pencil involves noisy measurements. Theoretically, in the noiseless case it is known that n (or possibly fewer) solutions reducing the pencil rank exist. However, after the perturbation these solutions may be elusive. It is this very property of the noiseless solvable case that leads us to a different perspective in treating the nonsquare generalized eigenvalue problem.

*Received by the editors May 23, 2003; accepted for publication (in revised form) by B. Kågström April 8, 2004; published electronically November 22, 2005.

<http://www.siam.org/journals/simax/27-2/42879.html>

[†]Faculté Libre des Sciences et Technologies, Institut Catholique de Lille, 41 rue du Port, 59046 Lille Cedex, France (gregory.boutry@icl-lille.fr).

[‡]Computer Science Department, The Technion - Israel Institute of Technology, Haifa, 32000 Israel (elad@cs.technion.ac.il).

[§]Department of Electrical Engineering, University of California, Santa Cruz, CA 95064 (milanfar@ee.ucsc.edu).

¹If B is invertible, one immediately sees the relation to the regular eigenvalue problem because $(A - \lambda B)\underline{v} = \underline{0}$ implies $B^{-1}A\underline{v} = \lambda\underline{v}$.

1.2. Problem formulation. Removing the application details, we can cast the eventual problem in the following form: Given the matrix pair $(A, B) \in \mathcal{M}^{m \times n}$ where $m > n$, we assume that these matrices originated from the pair $(A_0, B_0) \in \mathcal{M}^{m \times n}$ by perturbation (additive noise). We further assume that for the original pair there exist n distinct eigenpairs of the form $(A_0 - \lambda_k B_0)\underline{v}_k = 0$, $k = 1, 2, \dots, n$. Therefore, given the measured pair (A, B) we search for both the eigenpairs $\{\lambda_k, \underline{v}_k\}_{k=1}^n$ and the original pair (A_0, B_0) . Clearly, even if $m = n$, there may be some situations when there are not n solutions. Similarly, in the general case of $m > n$, there may not be any solution to the pencil $(A - \lambda B)\underline{v} = 0$, and we are searching for the minimal perturbation such that a rank-reducing solution exists. We will assume hereafter that the perturbed matrices A and B are full rank (i.e., n). More formally, we are interested in solving the following optimization problem:

$$(1) \quad \text{Given } A, B, \text{ find } \min_{A_0, B_0, \{\lambda_k, \underline{v}_k\}_{k=1}^n} \|A_0 - A\|_F^2 + \|B_0 - B\|_F^2$$

$$\text{subject to } \{(A_0 - \lambda_k B_0)\underline{v}_k = 0, \quad \|\underline{v}_k\|_2^2 = 1\}_{k=1}^n.$$

The Frobenius norm is used to measure the perturbation, entrywise, between the original pair and the estimated pair. This proposed formulation coincides with the maximum-likelihood description if the perturbation is assumed to originate from an additive white and zero mean Gaussian noise.

It should be noted that in many cases there is a need to constrain the solution to have a specific structure (e.g., Hankel/Toeplitz). In this work we neglect this option and leave this topic for future research. We note, however, that exploiting such additional information should lead to a more satisfactory solution. Similarly, in the spirit of generalizing the above problem we might be interested in adding weights to the computed Frobenius norm in order to emphasize some of the entries, or to cover cases when some of the entries are exact (e.g., fixed zeros). Again, we leave this weighted form of the problem for future research.

1.3. Relation to previous work. Several previous papers already addressed the treatment of perturbed rectangular pencils. However, all these contributions concentrate on theoretical analysis, proposing extensions based on perturbation theory, and forming bounds on the error measure we defined in (1) as our minimization goal. In contrast, this paper discusses a numerical algorithm for finding the eigenvalues and the matrix pair (A_0, B_0) .

Demmel and Kågström [4] suggested a treatment for nonsquare pencils based on the Kronecker canonical form (KCF). Under the assumption on the regularity of the KCF, the spectrum of the perturbed pencil $(A + E) - \lambda(B + F)$ is considered and compared to the spectrum of the original pencil $A - \lambda B$. Later work by Edelman, Elmroth, and Kågström and Elmroth, Johansson, and Kågström extends this work, suggesting a geometric interpretation of the above perturbation results [6, 8, 9]. Bounds are obtained for the distance from the pencil $A - \lambda B$ to $A_0 - \lambda B_0$. A similar approach and treatment is found in [2, 22]. Again, lower and upper bounds on the distance to a solvable pair of matrices are developed.

Another line of research relevant to this work is the study of pseudospectra for rectangular pencils (see the paper by Wright and Trefethen [25] on this topic). Most of the contributions in this field refer to the square pencil case [23, 11, 24]. When treating a rectangular nonsolvable pencil, we see that its pseudospectra is most likely to be empty for very small distances. As this distance grows, at some point the pseudospectra becomes nonempty. This is exactly the distance to which our solution

should point. More on the relation of pencils to pseudospectra will be mentioned in section 5.

In this work we focus on the minimal perturbation that guarantees a solution. However, we concentrate on a practical numerical scheme that finds the optimal (in the maximum-likelihood sense) eigenpairs, and also leads to the perturbed pair that guarantees a solution. We find a set of eigenvalues that minimizes the distance and we find the matrices that are the closest.

1.4. This paper's contribution and content. Solving the optimization problem in (1) is the challenge of this paper. We open the discussion with an effective method for visualizing the problem at hand. Then we present an analysis and insight for the following two simplified variations of the problem posed in (1):

1. We assume $n = 1$, so that $A = \underline{a}$ and $B = \underline{b}$. For this case we show that the defined problem leads to a closed form solution. We also show equivalence to the total least squares (TLS) method (cf. [13, p. 595]).
2. We assume $n > 1$ and search for a single eigenpair $\{\lambda, \underline{v}\}$. For this more complicated problem, a stable and convergent numerical algorithm is proposed.

This paper is organized as follows. In the next section we present a method for visualizing the behavior of such nonsquare pencils as a function of the scalar λ , and this way get a better understanding of the features of the problem at hand. In section 3 we show how the nonsquare pencil is solved analytically for the case of $n = 1$; i.e., the matrices A and B are single vectors. Through this solution we obtain a better understanding of the solution mechanism for the more general case. Section 4 treats the more general case with $n > 1$, but with the simplifying assumption that only one eigenpair is sought. A highly effective numerical method is proposed and analyzed. In section 5, we discuss the relationship between our work and the notion of pseudospectra, showing the similarities and the differences between these two approaches toward the issue of analyzing rectangular pencils. In section 6, we present a new factorization for the rectangular pencil, and through it an analysis of our initialization choice in the numerical algorithm. We conclude this paper in section 7 and summarize its contribution, list some open questions we have not treated, and outline future research directions.

The problem is wide and complicated. We regard this contribution as a first stage in a sequence of activities made to encompass its various aspects.

2. Visualizing nonsquare pencils. In order to gain insight into the treatment of such nonsquare pencils, let us look at a related function

$$(2) \quad f(\lambda) = \sigma_{\min}(A - \lambda B) = \min_{\underline{v} \mid \|\underline{v}\|^2=1} \|(A - \lambda B)\underline{v}\|_2.$$

The notation $\sigma(X)$ designates the singular values of the matrix X and $\sigma_{\min}$ is the smallest singular value. It is interesting to note that this function is used often for the construction of the pseudospectra of square matrices [25, 23]. This function describes the constraints in (1) as a penalty term, and it has several interesting properties that may help us visualize the behavior of the nonsquare pencil. This function is a mapping from the complex plane (values of λ) to the real and nonnegative numbers, because $f(\lambda)$ is nonnegative by the definition of the singular values. For a solvable pair of matrices A and B (i.e., a pair that has a rank-reducing solution λ_0), the value of the function at this λ_0 value is exactly zero. This is trivial since, if the pencil $A - \lambda_0 B$ is rank deficient, it must have zero as its smallest singular value (and may have more than one zero singular value). For such a solvable pair the eigenvalues are the local

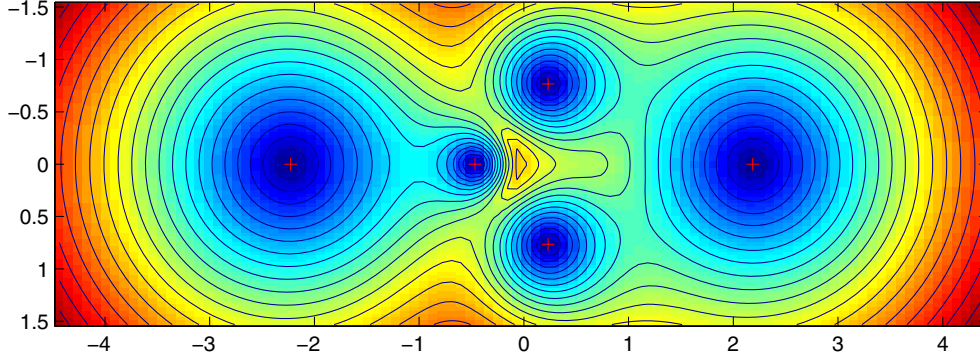


FIG. 1. A plot of the level curves of $f_0 = \sigma_{\min}(A_0 - \lambda B_0)$, built using the unperturbed pair (A_0, B_0) .

minimum points of this function. This is easily seen since, if λ_0 is perturbed, we get a full rank pencil that leads to a strictly positive minimal singular value and, in the case of a nonsolvable pair A and B , the function is strictly positive for all λ . However, the local minimum of this function may still help in finding a solution to (1).

Due to the first property, it is quite easy for us to visualize this function and learn about its behavior. Let us describe a simple example: We start by creating a pair of 5×5 random matrices, denoted as $\tilde{A}$ and $\tilde{B}$. The entries are drawn from a zero mean Gaussian distribution with standard deviation $\sigma = 1$. We verify that this pair has five distinct generalized eigenvalues denoted as $\{\alpha_k\}_{k=1}^5$. Let Q be an arbitrary full rank matrix of size 50×5 ; we create the pair $A_0 = Q\tilde{A}$ and $B_0 = Q\tilde{B}$. This pair of matrices is made up of rectangular 50×5 matrices. It is clear that they represent a solvable pair since, for all k ,

$$\begin{aligned} \text{Rank}(\tilde{A} - \alpha_k \tilde{B}) &= 4 \\ \implies \text{Rank}(A_0 - \alpha_k B_0) &= \text{Rank}(Q\tilde{A} - \alpha_k Q\tilde{B}) = \text{Rank}(Q(\tilde{A} - \alpha_k \tilde{B})) = 4. \end{aligned}$$

Given this rectangular pair, Figure 1 shows the 2D function $f_0(\lambda) \equiv \sigma_{\min}(A_0 - \lambda B_0)$. The exact eigenvalues $\{\alpha_k\}_{k=1}^5$ are overlaid (represented by “+” signs), and the minimum points of f_0 match exactly the location of the eigenvalues.

We proceed and define $A = A_0 + N_A$ and $B = B_0 + N_B$, where both N_A and N_B are of size 50×5 with entries drawn from a Gaussian white noise distribution with $\sigma = 0.1$. In Figure 2 we describe the function $f(\lambda) = \sigma_{\min}(A - \lambda B)$ for the noisy version of the original pair. As can be seen, there are still five local minima. However, relative to the location of the true eigenvalues, the local minima points are slightly deviated.

To conclude this experiment, we present in Figure 3 yet another function $f(\lambda)$, built using a perturbed pair A and B but with much stronger noise $\sigma = 5$. We see that all the local minima points vanished and there is a single minimum point near the origin. This demonstrates the difficulty we will be facing when the perturbation is relatively great.

Figure 4 shows an intersection of these functions with the real axis. As can be seen, for f_0 the values at the minimum are exactly zero, as expected.² As we go to the

²Since this function was sampled uniformly, near zero values are obtained.

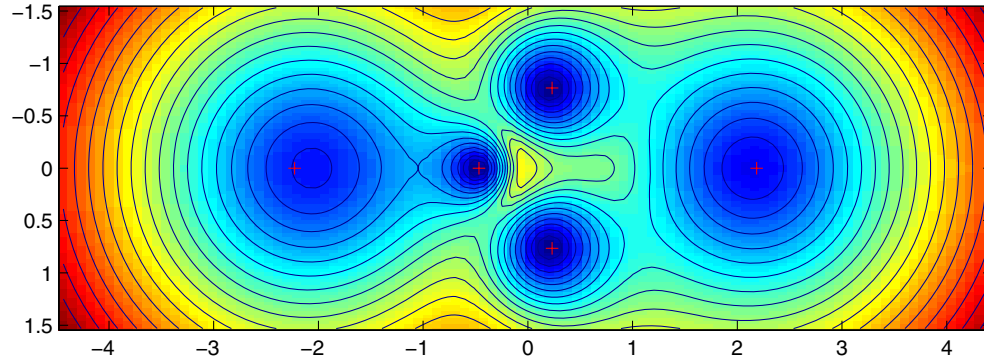


FIG. 2. A plot of the level curves of $f = \sigma_{\min}(A - \lambda B)$, built using a moderately perturbed pair (A, B) ($\sigma = 0.1$).

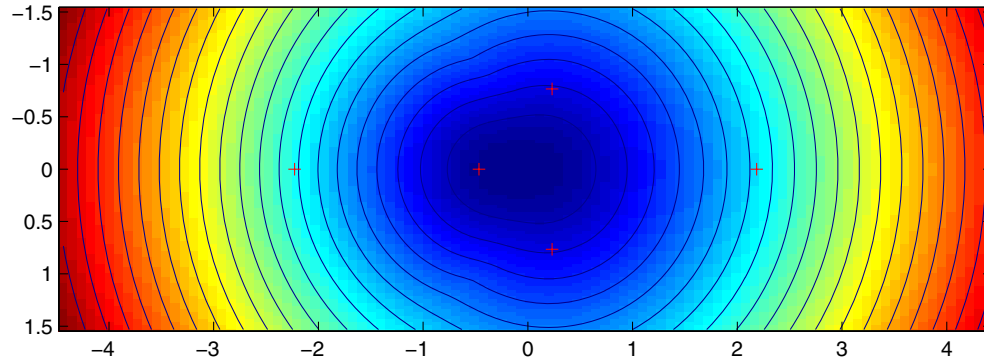


FIG. 3. A plot of the level curves of $f = \sigma_{\min}(A - \lambda B)$, built using a strongly perturbed pair (A, B) ($\sigma = 5$).

function f with the moderate noise, we get that the minimum points' height is above zero, though they exist and are close to the true points. The case of strong perturbation seems to appear as a unimodal function with one minimal point at the origin.

From this experiment it is clear that the function $f(\lambda) = \sigma_{\min}(A - \lambda B)$ could be beneficial in obtaining a good guess for the required eigenvalues. In the small perturbation case this idea seems to work quite well. We also see that in solving the problem presented in (1) we actually create a new pair of matrices that reduce the height of the function $f(\lambda)$ to zero at the local minima. It is not clear how we can refine the eigenvalues to compensate for the induced shift due to the perturbation. Lastly, we see that strong perturbations may lead to numerical problems if n distinct eigenvalues are sought.

3. Solving the $n = 1$ case. In this section, we redefine the problem described in (1) while assuming $n = 1$, and show that this case is analytically solvable.

3.1. Analysis.

THEOREM 1. *Given any pair of vectors $\underline{a}$ and $\underline{b}$, a unique global solution of the minimization problem given (1) always exists, and is given by*

$$(3) \quad \underline{a}_0 = \underline{a} - \frac{(\underline{a} - \lambda \underline{b})}{1 + |\lambda|^2}, \quad \underline{b}_0 = \underline{b} + \frac{\lambda^*(\underline{a} - \lambda \underline{b})}{1 + |\lambda|^2},$$

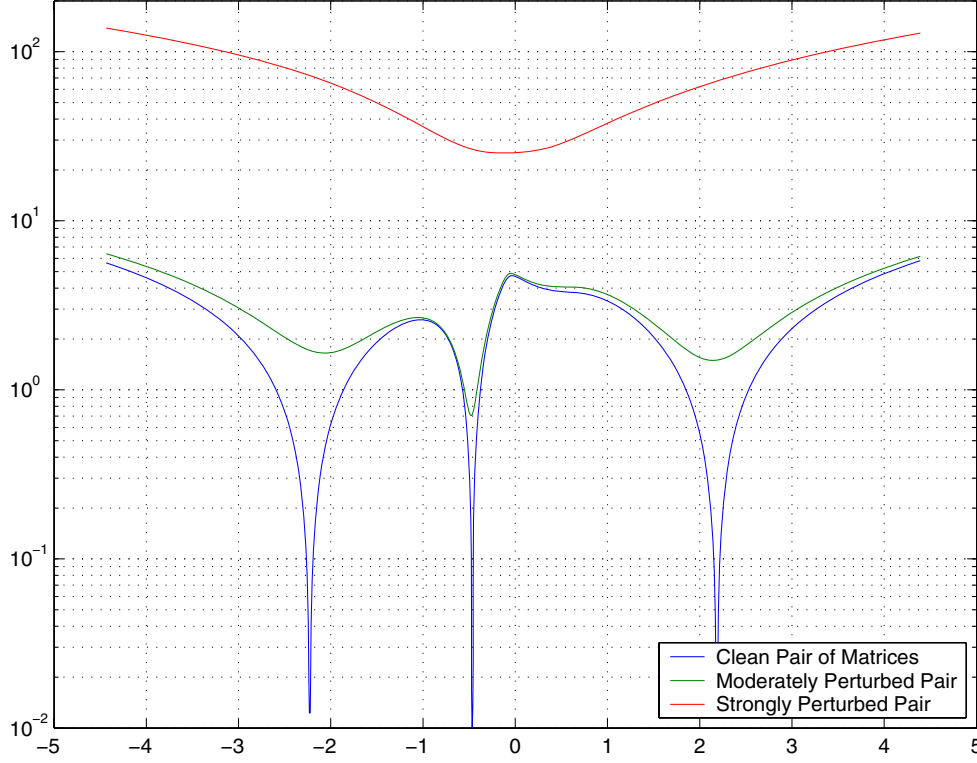


FIG. 4. A 1D intersection along the center horizontal axis (the real values) of the 2D functions described in Figures 1, 2, and 3.

where λ is the (+)-root³ of the quadratic equation

$$(4) \quad 0 = \lambda^2 \underline{a}^H \underline{b} + \lambda (\underline{b}^H \underline{b} - \underline{a}^H \underline{a}) - \underline{b}^H \underline{a}.$$

Proof. As this theorem is a particular case of a more general theorem presented in the next section, the proof will be given in section 4. $\square$

Next, we find an equivalent constraint-free optimization problem that eases the overall solution. Notice the close resemblance to the objective function used in the previous section for visualization.

THEOREM 2. *The optimization problem posed in (1) is equivalent to the optimization problem*

$$(5) \quad \min_{\{\lambda\}} = \frac{\|\underline{a} - \lambda \underline{b}\|_2^2}{1 + |\lambda|^2}.$$

The optimal solutions are given by the former theorem.

Proof. This theorem is also a particular case of a more general theorem presented in the next section, and again, the proof will be given later in section 4. $\square$

It is interesting to note that the observation made above, about the equivalence between the original optimization problem and the one posed in (5), implies that a function closely related to the one presented in the previous section (see (2)) is

³For a quadratic equation $\alpha x^2 + \beta x + \gamma = 0$, the (+)-root is given by $(-\beta + \sqrt{\beta^2 - 4\alpha\gamma})/(2\alpha)$.

minimized, but there is an additional term that moves the eigenvalue solution away from the origin. From Figures 2 and 3, it is clear that such a force is desired to better allocate the eigenvalues as the local minimum points of the treated function. As it turns out, this term in the denominator is quite commonly seen with respect to TLS (cf. [13, p. 595]). We next show that there is a close relation between our problem and TLS.

3.2. Relation to TLS. Here we show that the case of $n = 1$ actually forms a classic TLS problem. This strengthens our claim about the existence of a closed form solution for the $n = 1$ nonsquare pencil. However, as we shall see, this equivalence to TLS does not generalize to the $n > 1$ pencils.

THEOREM 3. *Given any pair of nonzero vectors $\underline{a}$ and $\underline{b}$, a unique global solution of the minimization problem posed in (1) is given by the singular value decomposition (SVD) of the matrix $[\underline{a} \ \underline{b}]$ by replacing the smaller, second singular value by zero.*

Proof. Writing (1) differently, we obtain a classic TLS formation

$$(6) \quad \min_{\underline{a}_0, \underline{b}_0} \|\underline{a}_0, \underline{b}_0\|_F^2 \quad \text{subject to} \quad \text{Rank}([\underline{a}_0, \underline{b}_0]) = 1.$$

The optimal solution for this problem can be obtained by the SVD on $[\underline{a}, \underline{b}]$ (cf. [13, p. 70]). Writing $UDV^T = [\underline{a}, \underline{b}]$, we have that U is an $m \times 2$ unitary matrix, V is a 2×2 unitary matrix, and D is a diagonal 2×2 matrix containing the singular values of $[\underline{a}, \underline{b}]$ in descending order. The optimal solution is obtained by replacing the second singular value by zero, and the solution becomes $[\underline{a}_0, \underline{b}_0] = d_{1,1} \underline{U}_1 \underline{V}_1^H$, where $\underline{U}_1$ and $\underline{V}_1$ are the first columns of U and V , respectively, and $d_{1,1}$ is the larger singular value. $\square$

This analogy between the pencil problem and the TLS holds only for $n = 1$ and cannot be employed for larger n . An attempt to rewrite (1) as a TLS problem appears as

$$\min_{A_0, B_0, V} \| [A_0, B_0] - [A, B] \|_F^2 \quad \text{subject to} \quad \begin{cases} \text{Rank}([A_0 V, B_0 V]) = n, \\ \text{Rank}(V) = n \end{cases},$$

where V is a matrix containing all the eigenvectors. Clearly, this is not a classic TLS formulation and its solution can no longer be applied using the SVD since V is involved. This explains our interest in the simpler solution for $n = 1$.

3.3. Squaring effect. Looking back at (3) and (4), it can be shown that

$$(7) \quad \underline{b}_0^H (\underline{a} - \lambda \underline{b}) = \left(\underline{b} + \frac{\lambda^* (\underline{a} - \lambda \underline{b})}{1 + |\lambda|^2} \right)^H (\underline{a} - \lambda \underline{b}) = \frac{(\underline{b}^H + \lambda \underline{a}^H)(\underline{a} - \lambda \underline{b})}{1 + |\lambda|^2} = 0.$$

This implies that

$$(8) \quad \lambda = \frac{\underline{b}_0^H \underline{a}}{\underline{b}_0^H \underline{b}}.$$

Similarly, if we repeat all the above analysis starting with (1) and using the constraint $\underline{b}_0 - \alpha \underline{a}_0 = 0$, where α plays the role of $1/\lambda$, we get a dual solution

$$(9) \quad \underline{a}_0^H \frac{\underline{b} - \alpha \underline{a}}{1 + |\alpha|^2} = 0 \implies \lambda = \frac{1}{\alpha} = \frac{\underline{a}_0^H \underline{a}}{\underline{a}_0^H \underline{b}}.$$

These two results imply that to find the optimal eigenvalue we square the rectangular pencil by multiplication on both sides by either $\underline{a}_0$ or $\underline{b}_0$. Of course, these vectors are not known and we have to go through a process as described above to find them. However, an approximate alternative to the above process could be an application of these squaring processes using the measured matrices instead of the estimated ones, that is,

$$(10) \quad \hat{\lambda} = \frac{\underline{b}^H \underline{a}}{\underline{b}^H \underline{b}} \quad \text{or} \quad \hat{\lambda} = \frac{\underline{a}^H \underline{a}}{\underline{a}^H \underline{b}}.$$

Then two different candidate solutions may be obtained as before. To choose between them, the corresponding $\underline{a}_0$ and $\underline{b}_0$ need to be computed. This could be done by solving the optimization problem given by (1), while assuming that λ is known. It is not hard to see that the solution is the one proposed in (3). As before, after computing $\underline{a}_0$ and $\underline{b}_0$ for the solutions of λ we may choose the one which leads to the smallest perturbation.⁴ Note that this approximate solution is expected to be close to the optimal solution if a small perturbation is assumed. We shall use this squaring idea for the general ($n > 1$) case.

3.4. Examples. We now give several examples to see the behavior of the above results.

Example 1. Assume $\underline{a} = \underline{b} \neq \underline{0}$. This case is solvable without any perturbation and it is clear that the proper eigenvalue should be 1.

The quadratic equation (4) becomes $\lambda^2 \underline{a}^H \underline{a} - \underline{a}^H \underline{a} = 0$, leading to the possible solutions $\{\lambda_1 = 1, \lambda_2 = -1\}$. Using (3) we get, for $\lambda_1 = 1$, $\underline{a}_0 = \underline{a}$ and $\underline{b}_0 = \underline{b}$, and thus the perturbation is $\underline{0}$. For the other solution $\lambda_1 = -1$, we obtain $\underline{a}_0 = \underline{0}$ and $\underline{b}_0 = \underline{0}$, and the perturbation is $\|\underline{a}\|^2 + \|\underline{b}\|^2$, and so clearly the (+)-root is the desired solution.

Using the approximate method in (10) we get that the two solutions are equal: $\lambda = \underline{b}^H \underline{a} / \underline{b}^H \underline{b} = \underline{a}^H \underline{a} / \underline{a}^H \underline{b} = 1$. Using (3), we obtain the optimal results that lead to no perturbation.

Example 2. Assume that $\underline{a}^H \underline{b} = \epsilon^2 \rightarrow 0$, $\|\underline{a}\|_2^2 = 1$, and $\|\underline{b}\|_2^2 = 0.5$. In this case the minimal perturbation zeros one of the vectors, and replaces the vector of smallest norm by zero. This leads to the optimal result (in our case it is $\underline{b}$ that should be nulled).

The quadratic equation in (4) becomes $0 = \lambda^2 \epsilon^2 - 0.5\lambda - \epsilon^2$, and the solutions are $\{\lambda_1 = \infty, \lambda_2 = 0\}$ for $\epsilon \rightarrow 0$. The (+)-solution leads to $\underline{a}_0 = \underline{a}$ and $\underline{b}_0 = \underline{0}$ as required, and the second solution is similar but nulls $\underline{a}$ instead, which is incorrect.

Approximating the solution by $\lambda = \underline{b}^H \underline{a} / \underline{b}^H \underline{b} = 0$ gives the wrong solution, and using $\lambda = \underline{a}^H \underline{a} / \underline{a}^H \underline{b} = \infty$ leads to the optimal solution.

Example 3. Assume that for $\epsilon \rightarrow 0$ we have $\underline{a} = [1, 0]^T$ and $\underline{b} = [1, \epsilon]^T$. Clearly, if we replace ϵ by zero, we get the case described in Example 1. Thus, a perturbation of ϵ^2 in the ℓ^2 norm could lead to a possible solution. However, moving an $\epsilon/2$ from $\underline{b}$ to $\underline{a}$ also causes $\underline{a} = \underline{b}$ and leads to a better solution since the perturbation in this case is $0.5\epsilon^2$.

From (4) we get $0 = \lambda^2 + \lambda\epsilon^2 - 1$. The solutions are $\{\lambda_1 \approx 1, \lambda_2 \approx -1\}$. The first result is the correct one and the perturbed solution is given by (3):

⁴In this approximate approach we found that there is no solution that was consistently better.

$\underline{a}_0 = [1, \epsilon/2]^T$ and $\underline{b}_0 = [1, \epsilon/2]^T$, as required, and the obtained perturbation is $\epsilon^2/2$.

Using the squaring approximation, one solution is given by λ being $\underline{b}^H \underline{a} / \underline{b}^H \underline{b} = 1/(1 + \epsilon^2)$. Then the minimal perturbation is roughly $3\epsilon^2$. The other solution $\lambda = \underline{a}^H \underline{a} / \underline{a}^H \underline{b} = 1$ leads to the optimal solution as before.

Example 4. Since in all the previous examples the approximate method gave the optimal solution, this might lead to a wrong conjecture that it is always equivalent to the optimal procedure. We give a counterexample, where this approximation fails. Assume $\underline{a} = [0.5, 0.5]^T$ and $\underline{b} = [0.5, -0.25]^T$. In this case, by using the exact solver (or the TLS approach) it is easily verified that the optimal solution is $\lambda_1 = 2$, which is the (+)-root $\underline{a}_0 = [0.6, 0.3]^T$ and $\underline{b}_0 = [0.3, 0.15]^T$. Thus, the minimal perturbation is 0.25. The second solution leads to $\lambda_2 = -1/2$ and the implied perturbation turns out to be bigger.

Using the approximation method, we get two possible solutions $\lambda_1 = 0.4$, $\lambda_2 = 4$, and clearly, both are wrong. It turns out that using the first, the perturbation is 0.3879, and using the second, the perturbation is 0.2647, and so we see that for both solutions the result is indeed suboptimal.

4. $n > 1$. The one-eigenvalue case.

4.1. Analysis. We analyze the $n > 1$ case by discussing a simplified problem, where indeed $n > 1$ but only one eigenvalue is known to exist. The optimization problem given in (1) simplifies in this case to

$$(11) \quad \min_{A_0, B_0, \lambda, \underline{v}} \|A_0 - A\|_F^2 + \|B_0 - B\|_F^2$$

$$\text{subject to } (A_0 - \lambda B_0)\underline{v} = 0 \quad \text{and} \quad \|\underline{v}\|_2^2 = 1,$$

where A, B are given.

As we shall see, this case corresponds to the assumption that the perturbation on the pair (A, B) is rank-one. Similar to the analysis for $n = 1$, we find an equivalent constraint-free optimization problem that simplifies the overall solution.

THEOREM 4. *The optimization problem posed in (11) is equivalent to the optimization problem*

$$(12) \quad \min_{\{\lambda, \underline{v}\}} \frac{\|(A - \lambda B)\underline{v}\|_2^2}{1 + |\lambda|^2} \quad \text{subject to} \quad \|\underline{v}\|_2^2 = 1.$$

The optimal solution for A_0 and B_0 is obtained by a rank-one perturbation to the original pair A and B :

$$(13) \quad A_0 = A - \frac{A - \lambda B}{1 + |\lambda|^2} \underline{v} \underline{v}^H \quad \text{and} \quad B_0 = B + \lambda^* \frac{A - \lambda B}{1 + |\lambda|^2} \underline{v} \underline{v}^H.$$

Proof. To solve the optimization problem in (11), we express it by distinguishing the real part (x-part) and the imaginary part (y-part) of each variable. We solve the problem by the Lagrange multipliers method. We define a Lagrangian function

$$(14) \quad \mathcal{L}\{A_x^0, A_y^0, B_x^0, B_y^0\}$$

$$= \frac{1}{2}(\|A_x^0 - A_x\|_F^2 + \|A_y^0 - A_y\|_F^2 + \|B_x^0 - B_x\|_F^2 + \|B_y^0 - B_y\|_F^2)$$

$$+ \gamma_1^T [(A_x^0 - \lambda_x B_x^0 + \lambda_y B_y^0)\underline{v}_x - (A_y^0 - \lambda_y B_x^0 - \lambda_x B_y^0)\underline{v}_y]$$

$$+ \gamma_2^T [(A_x^0 - \lambda_x B_x^0 + \lambda_y B_y^0)\underline{v}_y + (A_y^0 - \lambda_y B_x^0 - \lambda_x B_y^0)\underline{v}_x]$$

$$+ \delta(\|\underline{v}_x\|_2^2 + \|\underline{v}_y\|_2^2 - 1).$$

The vectors $\underline{\gamma}_x, \underline{\gamma}_y$ and the scalar δ are the Lagrange multipliers of the constraints. Taking derivatives with respect to A_x^0, A_y^0, B_x^0 , and B_y^0 , we get

$$\begin{aligned}\frac{\partial \mathcal{L}}{\partial A_x^0} &= A_x^0 - A_x + \underline{\gamma}_1 v_x^T + \underline{\gamma}_2 v_y^T = 0, \\ \frac{\partial \mathcal{L}}{\partial A_y^0} &= A_y^0 - A_y - \underline{\gamma}_1 v_y^T + \underline{\gamma}_2 v_x^T = 0, \\ \frac{\partial \mathcal{L}}{\partial B_x^0} &= B_x^0 - B_x - \lambda_x \underline{\gamma}_1 v_x^T + \lambda_y \underline{\gamma}_1 v_y^T - \lambda_x \underline{\gamma}_2 v_y^T - \lambda_y \underline{\gamma}_2 v_x^T = 0, \\ \frac{\partial \mathcal{L}}{\partial B_y^0} &= B_y^0 - B_y + \lambda_y \underline{\gamma}_1 v_x^T + \lambda_x \underline{\gamma}_1 v_y^T + \lambda_y \underline{\gamma}_2 v_y^T - \lambda_x \underline{\gamma}_2 v_x^T = 0.\end{aligned}$$

From this equation, we can deduce that

$$(15) \quad A_0 = A - \underline{\gamma} \underline{v}^H, \quad B_0 = B + \lambda^* \underline{\gamma} \underline{v}^H,$$

with $\gamma = \gamma_1 + i\gamma_2$. These expressions validate our claim regarding the rank-one update. Using the constraint $(A_0 - \lambda B_0) \underline{v} = \underline{0}$, we obtain

$$\begin{aligned}0 &= (A_0 - \lambda B_0) \underline{v} = (A - \lambda B) \underline{v} - (\underline{\gamma} \underline{v}^H + |\lambda|^2 \underline{\gamma} \underline{v}^H) \underline{v} \\ &= (A - \lambda B) \underline{v} - (1 + |\lambda|^2) \underline{\gamma},\end{aligned}$$

where we have used the constraint $\|\underline{v}\|^2 = 1$. Thus,

$$(16) \quad \underline{\gamma} = \frac{(A - \lambda B) \underline{v}}{1 + |\lambda|^2}.$$

Using this in (15), we get

$$(17) \quad A_0 = A - \frac{A - \lambda B}{1 + |\lambda|^2} \underline{v} \underline{v}^H \quad \text{and} \quad B_0 = B + \lambda^* \frac{A - \lambda B}{1 + |\lambda|^2} \underline{v} \underline{v}^H.$$

For an eigenpair $\{\lambda, \underline{v}\}$ we may compute A_0 and B_0 using this equation, and the perturbation can be derived easily, leading to

$$\begin{aligned}(18) \quad \|A_0 - A\|_F^2 + \|B_0 - B\|_F^2 &= \left\| \frac{A - \lambda B}{1 + |\lambda|^2} \underline{v} \underline{v}^H \right\|_F^2 + \left\| \lambda^* \frac{A - \lambda B}{1 + |\lambda|^2} \underline{v} \underline{v}^H \right\|_F^2 \\ &= \frac{\|(A - \lambda B) \underline{v} \underline{v}^H\|_F^2}{1 + |\lambda|^2} = \frac{\|(A - \lambda B) \underline{v}\|_2^2}{1 + |\lambda|^2}.\end{aligned}$$

Here we used the property $\|\underline{u} \underline{z}^H\|_F^2 = \|\underline{u}\|_2^2 \cdot \|\underline{z}\|_2^2$ and the fact that $\|\underline{v}\|_2^2 = 1$. Note that, as in the $n = 1$ case, we get a similar relation to the heuristic function $\|(A - \lambda B) \underline{v}\|_2$ described in section 2, but there is an additional force that moves the eigenvalue away from the origin.

Finally, it is clear that minimizing this function with respect to $\lambda_x, \lambda_y, \underline{v}_x, \underline{v}_y$ leads to the optimal solution of both problems posed in (11) and (12), as claimed.

For $n = 1$ we have that $\underline{v}$ is a scalar of unit norm. Thus $\underline{v} \underline{v}^H = \underline{v}^H \underline{v} = 1$, and from (13) and (12) we obtained the results stated in (3) and (5). $\square$

This suggests that, instead of solving the problem given by (11), finding optimal A_0, B_0, λ , and $\underline{v}$, one can solve the problem presented in (12) and find optimal values

for λ and $\underline{v}$ only. These optimal values can be used in (13) to obtain optimal A_0 and B_0 as well.

An interesting consequence of the above result is that evaluation of pseudospectra for nonsquare pencils can be performed using the function $\sigma_{\min}(A - \lambda B)/\sqrt{1 + |\lambda|^2}$. This is a generalization of the function $\sigma_{\min}(A - \lambda I)$ used for the square pseudospectra case.

In case there are several such candidate eigenpair solutions, one could check each by computing the perturbation obtained and choose the solution that leads to the smallest value. This perturbation is given simply by $\|(A - \lambda B)\underline{v}\|_2/\sqrt{1 + |\lambda|^2}$. In fact, multiple solutions for (12) may lead to the solution of the more general case of having more than one eigenpair.

The problem of finding the optimal eigenpair remains. We next show the necessary conditions on $\{\lambda, \underline{v}\}$ for both optimization problems.

THEOREM 5. *The necessary conditions on $\{\lambda, \underline{v}\}$ to solve both optimization problems posed in (11) and (12) are given by*

$$(19) \quad \underline{v}^H (B^H + \lambda A^H) (A - \lambda B) \underline{v} = 0,$$

$$(20) \quad (A - \lambda B)^H (A - \lambda B) \underline{v} = \alpha \underline{v}$$

for the real and positive scalar $\alpha = \underline{v}^H (A - \lambda B)^H (A - \lambda B) \underline{v}$.

Proof. Returning to the Lagrangian function in (14) and taking derivatives with respect to λ_x, λ_y , we get

$$\begin{aligned} \frac{\partial \mathcal{L}}{\partial \lambda_x} &= -\gamma_1^T B_x^0 \underline{v}_x + \gamma_1^T B_y^0 \underline{v}_y - \gamma_2^T B_x^0 \underline{v}_y - \gamma_2^T B_y^0 \underline{v}_x = 0, \\ \frac{\partial \mathcal{L}}{\partial \lambda_y} &= \gamma_1^T B_y^0 \underline{v}_x + \gamma_1^T B_x^0 \underline{v}_y + \gamma_2^T B_y^0 \underline{v}_y - \gamma_2^T B_x^0 \underline{v}_x = 0. \end{aligned}$$

In fact, we have

$$\begin{aligned} \underline{v}^H B_0^H \underline{\gamma} = 0 \text{ or } 0 &= \underline{v}^H \left(B^H + \underline{v} \underline{v}^H \lambda \frac{A^H - \lambda^* B^H}{1 + |\lambda|^2} \right) \frac{(A - \lambda B) \underline{v}}{1 + |\lambda|^2} \\ (21) \quad &= \underline{v}^H \frac{(B^H + \lambda A^H) (A - \lambda B)}{(1 + |\lambda|^2)^2} \underline{v}. \end{aligned}$$

The denominator may be discarded since $1 + |\lambda|^2 \geq 1$. The resemblance to (4) is evident. Note also that for $n = 1$ this immediately leads to (4). To validate our claim in Theorem 1 that the (+)-root is the proper solution, we take the second derivative of (5) with respect to λ (done properly, with real and imaginary parts separated) and show that this derivative is negative for the (+)-root. The function we refer to is

$$(22) \quad f(\lambda) = \frac{\|\underline{a} - \lambda \underline{b}\|_2^2}{1 + |\lambda|^2} = \frac{\|\underline{a}_x - \lambda_x \underline{b}_x + \lambda_y \underline{b}_y\|_2^2 + \|\underline{a}_y - \lambda_x \underline{b}_y - \lambda_y \underline{b}_x\|_2^2}{1 + |\lambda_x|^2 + |\lambda_y|^2}.$$

We skip the algebraic steps of the derivative, as those are elementary (and tedious). It can be shown that the second derivative is indeed negative. Equation (4) will be shown as a consequence of the next theorem.

Returning to the $n > 1$ case, we see that any nontrivial solution (i.e., $\|\underline{v}\|_2^2 = 1$) for the equation

$$(23) \quad \underline{v}^H [\lambda^2 A^H B + \lambda (B^H B - A^H A) - B^H A] \underline{v} = \underline{v}^H (B^H + \lambda A^H) (A - \lambda B) \underline{v} = 0$$

is a candidate for the minimum point of the problem in (11). Another constraint on the solution is given by taking a derivative of the Lagrangian in (14) with respect to $\underline{v}_x, \underline{v}_y$. In this way we get

$$\begin{aligned}\frac{\partial \mathcal{L}}{\partial \underline{v}_x} &= (A_x^0 - \lambda_x B_x^0 + \lambda_y B_y^0)^T \underline{\gamma}_1 + (A_x^0 - \lambda_y B_x^0 - \lambda_x B_y^0)^T \underline{\gamma}_2 + 2\delta \underline{v}_x = \underline{0}, \\ \frac{\partial \mathcal{L}}{\partial \underline{v}_y} &= -(A_y^0 - \lambda_y B_x^0 - \lambda_x B_y^0)^T \underline{\gamma}_1 + (A_x^0 - \lambda_x B_x^0 + \lambda_y B_y^0)^T \underline{\gamma}_2 + 2\delta \underline{v}_y = \underline{0}.\end{aligned}$$

Joining these together, we obtain

$$\begin{aligned}(24) \quad \underline{0} &= (A_0 - \lambda B_0)^H \underline{\gamma} + 2\delta \underline{v} \\ &= \left[(A - \lambda B) - (1 + |\lambda|^2) \frac{(A - \lambda B)}{1 + |\lambda|^2} \underline{v} \underline{v}^H \right]^H \frac{(A - \lambda B) \underline{v}}{1 + |\lambda|^2} + 2\delta \underline{v} \\ &= \frac{(I - \underline{v} \underline{v}^H) (A - \lambda B)^H (A - \lambda B) \underline{v}}{1 + |\lambda|^2} + 2\delta \underline{v}.\end{aligned}$$

Multiplying both sides by $\underline{v}^H$ from the left and exploiting again the unit norm of $\underline{v}$ yields

$$(25) \quad 0 = (\underline{v}^H - \underline{v}^H) (A - \lambda B)^H (A - \lambda B) \underline{v} + 2(1 + |\lambda|^2) \delta = 2(1 + |\lambda|^2) \delta,$$

implying that $\delta = 0$. Thus (24) becomes

$$0 = (I - \underline{v} \underline{v}^H) (A - \lambda B)^H (A - \lambda B) \underline{v}.$$

The matrix $(I - \underline{v} \underline{v}^H)$ is a projection matrix with a null space spanned by the vector $\underline{v}$. Thus, this leads to the requirement

$$(26) \quad (A - \lambda B)^H (A - \lambda B) \underline{v} = \alpha \underline{v}$$

for the real and positive scalar $\alpha = \underline{v}^H (A - \lambda B)^H (A - \lambda B) \underline{v}$. To conclude this part, the optimal eigenpair $\lambda, \underline{v}$ should satisfy (23) and (26), as claimed by Theorem 5. Note that these two requirements could be derived directly from (12). $\square$

The two proposed requirements on the optimal λ and $\underline{v}$ cannot be solved analytically. Instead, we propose a numerical algorithm for their solution, leading to the minimization of the function given in (12).

4.2. Numerical algorithm. From our understanding gained in Theorem 4, and the necessary conditions on λ and $\underline{v}$ in Theorem 5, we propose a numerical algorithm to minimize (12). The algorithm requires a proper initialization guess for λ . Using (21), we obtain $\underline{v}^H B_0^H (A - \lambda B) \underline{v} = \underline{0}$. This equation can be approximated by the solution of $B^H (A - \lambda B) \underline{v} = \underline{0}$, which is a square generalized eigenvalue problem and equivalent to the problem $(B^\dagger A - \lambda I) \underline{v} = \underline{0}$, and this can serve for the computation of the initial values.

Given an initial λ , the optimal $\underline{v}$ is obtained by the right singular vector corresponding to the smallest singular value of $(A - \lambda B)^H (A - \lambda B)$, as suggested above. Using this updated vector, from (19) we obtain a quadratic equation with respect to λ . In the spirit of our discussion in section 3, we choose the (+)-root, leading to the minimum of the function with respect to λ . This process is repeated, alternating between an updated λ and an updated $\underline{v}$ until convergence. Algorithm A is a pseudocode description of this algorithm.

ALGORITHM A. *Numerical algorithm for the minimization of the $n > 1$ case with one eigenpair.*

Task: Minimize $\|(A - \lambda B)\underline{v}\|_2^2 / (1 + |\lambda|^2)$ with respect to λ and $\underline{v}$.

Initialization: Choose initial λ from $(B^H A - \lambda B^H B)\underline{v} = 0$.

Repeat until convergence:

- Update $\underline{v}$: Compute $\underline{v}$ as the right singular vector corresponding to the smallest singular value of the matrix $A - \lambda B$.
- Update λ : Choose λ as the (+)-root of the scalar quadratic equation $\underline{v}^H (B^H + \lambda A^H) (A - \lambda B) \underline{v}$.

Finalize: With λ and $\underline{v}$ compute A_0 and B_0 using (17).

THEOREM 6. *Algorithm A is guaranteed to converge to a local minimum of the function $f(\lambda, \underline{v}) = \|(A - \lambda B)\underline{v}\|_2^2 / (1 + |\lambda|^2)$.*

Proof. For the case $n = 1$, the convergence is established in one step, since the update of $\underline{v}$ is unnecessary, and the solution is given by solving the quadratic equation for λ .

Let us look at the function $f(\lambda, \underline{v})$ at the k th iteration and treat the general $n > 1$ case. For a current solution $\underline{v}_k$, we have two candidate solutions for λ_{k+1} by solving the quadratic equation (23). The solutions are

$$\lambda_{k+1}^{\pm} = \frac{-\underline{v}_k B^H B \underline{v}_k + \underline{v}_k A^H A \underline{v}_k \pm \sqrt{(\underline{v}_k B^H B \underline{v}_k - \underline{v}_k A^H A \underline{v}_k)^2 + 4(\underline{v}_k A^H B \underline{v}_k)^2}}{2\underline{v}_k A^H B \underline{v}_k}.$$

It can be shown that the second derivative of $f(\lambda, \underline{v})$ for fixed $\underline{v}_k$ and the two solutions for λ_{k+1} are proportional to

$$\mp \sqrt{(\underline{v}_k B^H B \underline{v}_k - \underline{v}_k A^H A \underline{v}_k)^2 + 4(\underline{v}_k A^H B \underline{v}_k)^2}.$$

Thus the (+)-root is the minimizer of $f(\lambda, \underline{v})$, and we get that

$$f(\lambda_{k+1}, \underline{v}_k) \leq f(\lambda_k, \underline{v}_k).$$

Given λ_{k+1} , we update $\underline{v}_{k+1}$ by choosing the rightmost singular vector of the matrix $A - \lambda_{k+1}B$, and it immediately follows that

$$\|(A - \lambda_{k+1}B)\underline{v}_{k+1}\|_2^2 \leq \|(A - \lambda_{k+1}B)\underline{v}_k\|_2^2.$$

Thus we obtain

$$0 \leq f(\lambda_{k+1}, \underline{v}_{k+1}) \leq f(\lambda_{k+1}, \underline{v}_k) \leq f(\lambda_k, \underline{v}_k),$$

which guarantees convergence. $\square$

4.3. Example. Theoretical analysis of the rate of convergence is difficult. Extensive experiments consistently show successful convergence at a rate similar to that shown in the examples below. In those experiments we see a linear rate of convergence. We define the rate of convergence as

$$r(k) = \frac{f(\lambda_{k+1}, \underline{v}_{k+1}) - f(\lambda_{\infty}, \underline{v}_{\infty})}{f(\lambda_k, \underline{v}_k) - f(\lambda_{\infty}, \underline{v}_{\infty})}.$$

Figure 5 presents a graph of $r(k)$ as a function of k for a specific yet representative

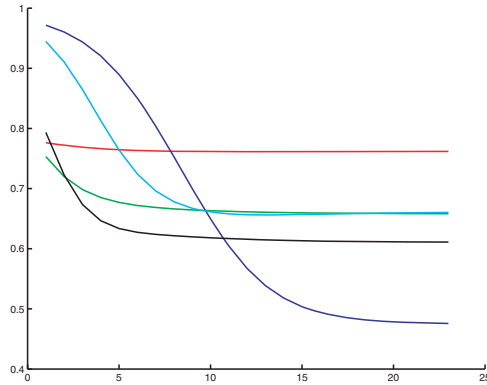


FIG. 5. Convergence of $r(k)$ as a function of k for a 15×5 matrix with $\sigma = 0.1$.

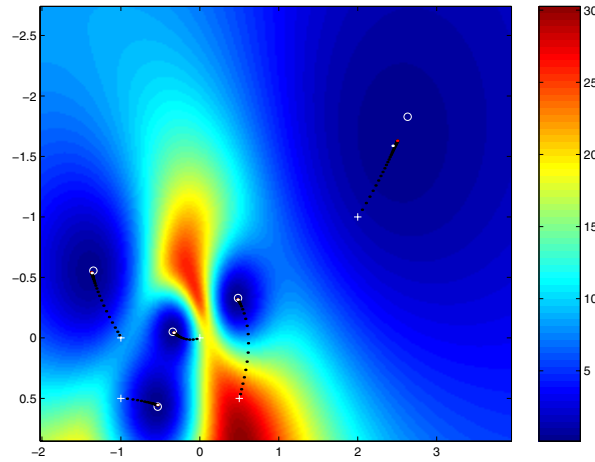


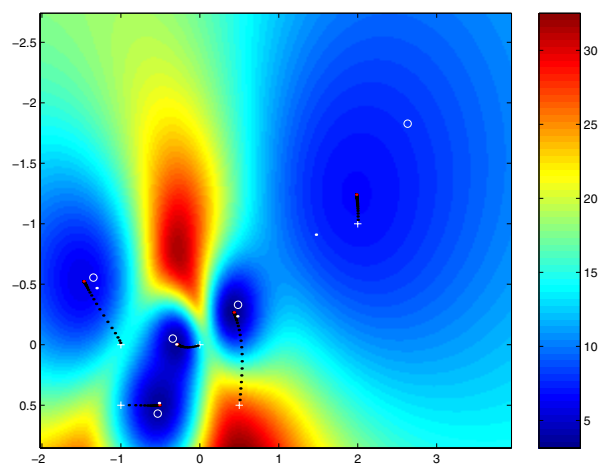
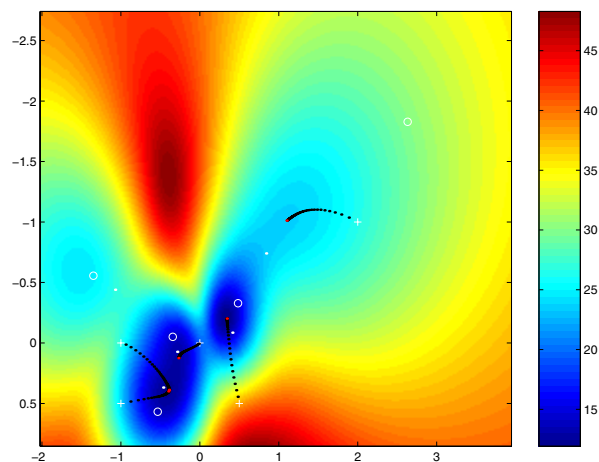
FIG. 6. The function $f(\lambda) = \sigma_{\min}(A - \lambda B)$ using $\sigma = 0.1$.

example. The pencil was chosen to be of size 15×5 , generated similarly to those described in section 2 and with $\sigma = 0.1$. We plotted the five different curves obtained with different initial values chosen manually (such that they are far worse than the ones found by the proposed squaring approach). As can be seen, $r(k)$ tends to stabilize to a constant value, indicating linear convergence.

In Figure 6, the function

$$f(\lambda) = \min_{\underline{v}} \frac{\|(A - \lambda B)\underline{v}\|_2^2}{1 + |\lambda|^2}$$

is visualized, along with the iterative process using the same experiment mentioned above with $\sigma = 0.1$ (note that this time we used complex entries so that the function is no longer symmetric). The +’s mark the manually chosen initial points, and the white points are the initial points that could have been obtained by the squaring method (those points are much closer to the steady-state solution). The o’s mark the true eigenvalues of the clear matrix pair (A_0, B_0) . The black dots present the iterative

FIG. 7. The function $f(\lambda) = \sigma_{\min}(A - \lambda B)$ using $\sigma = 0.5$.FIG. 8. The function $f(\lambda) = \sigma_{\min}(A - \lambda B)$ using $\sigma = 1.0$.

algorithm results. We can see that convergence to near optimal values is obtained, especially for those eigenvalues near the origin.

Figures 7 and 8 present a similar simulation for stronger noise. Figure 7 corresponds to $\sigma = 0.5$ and Figure 8 to $\sigma = 1$. As can be seen, the analyzed function's local minima points deviate gradually from the true values. As the noise gets stronger, some local minima may vanish altogether.

The simulations of the proposed algorithm and the creation of all the figures were performed using MATLAB. A typical run-time of the algorithm for the matrix sizes involved takes approximately 1–2 seconds for 20 iterations.

5. Relation to pseudospectra. For the regular eigenvalue problem with a square matrix A , we define λ as the eigenvalues of A satisfying the condition $(A - \lambda I)\underline{v} = \underline{0}$ for some nonzero n -vector $\underline{v}$. We write also that $\lambda \in \Lambda(A)$. Similarly, $\lambda \in \Lambda(A, B)$ refers to the generalized eigenvalues of the (possibly nonsquare) pencil $A - \lambda B$. A pseudospectra of a matrix extends the concept of eigenvalues and

generalized eigenvalues and considers their location due to perturbation of the matrices involved [23] (see also [3, 15, 19, 18] for a relation of this concept to stability). The notion of pseudospectra in general, and for rectangular pencils in particular, is a relatively recent subject of research. The definition of pseudospectra for the square case is given in [23]:

$$\Lambda_\epsilon(A) = \{z \in \Lambda(A + E) : \|E\|_F \leq \epsilon\}.$$

Many papers treat the square case [2, 11, 10, 24, 25]. For a comprehensive survey see [23]. A treatment of the nonsquare pencil is addressed in [25, 11] by generalizing the definition of Van Dorsselaer for pencils to⁵

$$\Lambda_\epsilon(A, B) = \{z : \exists \underline{u} \neq 0, (A + E)\underline{u} = z(B + F)\underline{u} \quad \forall \|E\|_F^2 + \|F\|_F^2 \leq \epsilon\}.$$

Clearly, the resemblance to our problem is evident. Analysis of the pseudospectra is primarily concerned with the patterns of eigenvalue location due to perturbation of the involved matrices. The above definition creates a cloud of points that correspond to all possible perturbation matrices E and F in the permissible range.

Referring to the above definition, our approach is similar and closely related. If we apply this definition to a nonsquare pencil with more rows than columns, it is most likely that this pseudospectra set will be empty. As we increase the perturbation energy ϵ , at some point we hit a solvable pair for which there are eigenvalues and the pseudospectra set is no longer empty. This is the solution of our proposed approach, since we seek the closest solvable pair. Thus, for a given starting pair of matrices A, B our method finds the minimal ϵ that leads to a nonempty pseudospectra. Moreover, our approach yields the exact perturbation that gets such a set of eigenvalues, and the eigenvalues themselves. In a way, we may refer to our approach as an *inverse pseudospectra* analysis of the treated pencil.

As a last comment in this discussion, we note that pseudospectra for a square matrix A is practically computed by studying the function $f(\lambda) = \sigma_{\min}(A - \lambda I)$ [23]. When we are working with matrix pencils, and allowing perturbation in both A and B , this function needs to be changed to $f(\lambda) = \sigma_{\min}(A - \lambda B) / \sqrt{1 + |\lambda|^2}$. The close relation between these two choices can be easily explained if one returns to Theorem 3 and discusses the maximum likelihood estimation of A_0 in the case when B is chosen to be fixed. Then, with the constraint $B_0 = B$, we obtain a target function without the denominator $\sqrt{1 + |\lambda|^2}$. Indeed, in the case of $B = I$, we return to the original function used by the pseudospectra.

6. A simplifying factorization. There are several ways to prefactorize the involved matrices A and B to simplify their structure, prior to the application of the numerical algorithm proposed here (see, for example, [21]). In this section we present one such way, which serves two different objectives: (i) requiring less computation in the iterative algorithm, and (ii) obtaining a bound on the error of the squaring initialization method. The proposed algorithm is a combination of the QR transformation and the GSVD due to Van Loan (cf. page 465 in [13] and [20]). A similar decomposition was given in [16, 17].

⁵The definition of pseudospectra used in [25] is actually slightly different, using separate and independent bounds on the perturbation of each of the two matrices involved. We chose to describe here a variant of their definition that aligns well with our point of view.

6.1. The factorization. We start our description with the assumption that $m \geq 2n$ (there are at least twice as many rows as there are columns in the pencil). The general case will be treated later.

THEOREM 7. *Consider a rectangular matrix pencil, defined by the two matrices $A, B \in \mathcal{M}^{m,n}$. An $m \times m$ unitary matrix W and an $n \times n$ invertible matrix X can be found such that*

$$A = W \begin{pmatrix} A_n \\ 0 \\ 0 \end{pmatrix} X^H, \quad B = W \begin{pmatrix} G \\ B_n \\ 0 \end{pmatrix} X^H,$$

with A_n, B_n two diagonal matrices with nonnegative terms such that $A_n^T A_n + B_n^T B_n = I_n$, and G is a full square matrix.⁶

Proof. First, apply a QR decomposition to the matrix $[A \ B]$. Consider the obtained upper triangular matrices $A_1 = QA$ and $B_1 = QB$. Applying the GSVD to the matrices $A_1(1:n, :)$ and $B_1(n+1:2n, :)$, we obtain two $n \times n$ unitary matrices U, V and an $n \times n$ invertible matrix X such that

$$A_1 = UA_n X^H, \quad B_1 = VB_n X^H,$$

with $A_n^T A_n + B_n^T B_n = I_n$. Defining the $m \times m$ unitary matrix $W := Q(U \oplus V \oplus I_{m-2n})$, we obtain the desired result. $\square$

The next theorem extends the above result for the case $n < m < 2n$.

THEOREM 8. *Consider a rectangular matrix pencil, built of the two matrices $A, B \in \mathcal{M}^{m,n}$. An $m \times m$ unitary matrix U and an $n \times n$ invertible matrix X can be found such that*

$$A = Q \begin{pmatrix} A_{m-n} \oplus I_{2n-m} \\ 0_{m-n,n} \end{pmatrix} X^H, \quad B = Q \begin{pmatrix} G \\ [B_{m-n} \ O_{m-n,2n-m}] \end{pmatrix} X^H,$$

where A_{m-n}, B_{m-n} are two $(m-n) \times (m-n)$ diagonal matrices with nonnegative terms such that $A_{m-n}^T A_{m-n} + B_{m-n}^T B_{m-n} = I_{m-n}$, and G is a full square matrix.

The proof of this theorem is omitted, as it immediately follows from the application of the GSVD to two matrices with different number of rows. We should note that we have found a speed-up factor of 2–3 for large matrices when using the factorization before iterating.

6.2. Squaring method for initialization. As a last point in our discussions in this section, we show that based on the factorization algorithm we are able to bound the error induced by the squaring initialization algorithm, as presented in previous sections.

THEOREM 9. *Consider a rectangular matrix pencil, built of the two full rank matrices $A, B \in \mathcal{M}^{m,n}$, and assume $m > 2n$. We denote the pair $\{\lambda_s, \underline{v}_s\}$ as the initialization pair, obtained by solving the square generalized eigenvalue problem $B^H(A - \lambda B)\underline{v} = \underline{0}$. Then, our target function*

$$f(\lambda, \underline{v}) = \frac{\|(A - \lambda B)\underline{v}\|_2^2}{1 + |\lambda|^2}$$

⁶Note that X is the product of an upper triangular matrix with a unitary matrix, which is also easily invertible.

satisfies

$$f(\lambda_s, \underline{v}_s) \leq \|A_n X\|_2^2,$$

where A_n and X are matrices obtained by the factorization given by applying it on the pair $[B, A]$.

Proof. Using the factorization (note that we replaced the order of the treated matrices) we get

$$A = W \begin{pmatrix} G \\ A_n \\ 0 \end{pmatrix} X^H, \quad B = W \begin{pmatrix} B_n \\ 0 \\ 0 \end{pmatrix} X^H.$$

Thus,

$$(27) \quad f(\lambda, \underline{v}) = \frac{\|(A - \lambda B)\underline{v}\|_2^2}{1 + |\lambda|^2} = \frac{\|(G - \lambda B_n)X^H \underline{v}\|_2^2 + \|A_n X^H \underline{v}\|_2^2}{1 + |\lambda|^2}.$$

On the other hand, when using the squaring method, we solve $B^H(A - \lambda B)\underline{v} = 0$. Using again the factorization, we get

$$(28) \quad B^H(A - \lambda B)\underline{v} = X B_n^H (G - \lambda B_n) X^H \underline{v} = 0.$$

Since X is invertible, and since B is full rank, it implies that the multiplication by $X B_n^H$ can be omitted from the above equation without affecting the solution. Thus, the squaring method finds the pair $\{\lambda_s, \underline{v}_s\}$ that solves $(G - \lambda_s B_n) X^H \underline{v}_s = 0$. Substituting this relation in (27), we obtain

$$(29) \quad f(\lambda_s, \underline{v}_s) = \frac{\|(G - \lambda_s B_n)X^H \underline{v}_s\|_2^2 + \|A_n X^H \underline{v}_s\|_2^2}{1 + |\lambda_s|^2} = \frac{\|A_n X^H \underline{v}_s\|_2^2}{1 + |\lambda_s|^2} \\ \leq \|A_n X^H \underline{v}_s\|_2^2 \leq \|A_n X^H\|_2^2 \leq \|A_n\|_2^2 \|X\|_2^2 \leq \|X\|_2^2,$$

where we have used $\|\underline{v}_s\|_2 = 1$ and $A_n^T A_n + B_n^T B_n = I_n$. $\square$

Based on the above theorem we have that for the function $f(\lambda, \underline{v}) = \|A - \lambda B\|_2^2 / (1 + |\lambda|^2)$, we get the relation

$$(30) \quad 0 \leq f(\lambda_{opt}, \underline{v}_{opt}) \leq f(\lambda_s, \underline{v}_s) \leq \|A_n X^H\|_2^2.$$

This implies that from the computable bound we are able to evaluate the proximity of our pencil pair (A, B) to a solvable pair. This result also implies that, while the squaring method does not minimize our penalty function, it performs near-optimal evaluation of the solution by concentrating on the easier part of the penalty, as can be seen in (27).

7. Conclusions and future work. This work has addressed the nonsquare generalized eigenvalue problem. In its general form, the problem we aimed to solve was

$$\min_{A_0, B_0, \{\lambda_k, \underline{v}_k\}_{k=1}^n} \|A_0 - A\|_F^2 + \|B_0 - B\|_F^2 \\ \text{subject to } \{(A_0 - \lambda_k B_0)\underline{v}_k = 0, \quad \|\underline{v}_k\|_2^2 = 1\}_{k=1}^n.$$

In this paper we treated two simplified variations of this problem. Starting with the $n = 1$ case, we showed that it leads to a closed form solution. Then we treated the $n > 1$ case with one eigenpair. For this more general problem we proposed a numerical algorithm and demonstrated its behavior. We presented a new factorization of such rectangular pencils, leading to a simplified implementation.

Work is underway to extend these results and treat the general problem with n eigenpairs as proposed above. Our final goal is to propose a reliable numerical method that can be extended to treat the structured pencil problem.

Acknowledgments. The authors would like to thank Professor Zhong-Zhi Bai (Chinese Academy of Sciences, China), Professor Mark Embree (Rice University, USA), and Professor Bo Kågström (Umeå University, Sweden) for helpful comments and discussions.

REFERENCES

- [1] Z. BAI, J. DEMMEL, J. DONGARRA, A. RUHE, AND H. VAN DER VORST, *Templates for the Solution of Algebraic Eigenvalue Problems: A Practical Guide*, SIAM, Philadelphia, 2000.
- [2] D. BOLEY, *Estimating the sensitivity of the algebraic structure of pencils with simple eigenvalue estimates*, SIAM J. Matrix Anal. Appl., 11 (1990), pp. 632–643.
- [3] J. V. BURKE, A. S. LEWIS, AND M. L. OVERTON, *Optimization and pseudospectra, with applications to robust stability*, SIAM J. Matrix Anal. Appl., 25 (2003), pp. 80–104.
- [4] J. W. DEMMEL AND B. KÅGSTRÖM, *Computing stable eigendecompositions of matrix pencils*, Linear Algebra Appl., 88 (1987), pp. 139–186.
- [5] J. W. DEMMEL AND B. KÅGSTRÖM, *The generalized Schür decomposition of an arbitrary pencil $A - \lambda B$: Robust software with error bounds and applications*. I. Theory and algorithms, ACM Trans. Math. Software, 19 (1993), pp. 160–174.
- [6] A. EDELMAN, E. ELMROTH, AND B. KÅGSTRÖM, *A geometric approach to perturbation theory of matrices and matrix pencils. Part I: Versal deformations*, SIAM J. Matrix Anal. Appl., 18 (1997), pp. 653–692.
- [7] A. EDELMAN, E. ELMROTH, AND B. KÅGSTRÖM, *A geometric approach to perturbation theory of matrices and matrix pencils. Part II: A stratification-enhanced staircase algorithm*, SIAM J. Matrix Anal. Appl., 20 (1999), pp. 667–699.
- [8] E. ELMROTH, P. JOHANSSON, AND B. KÅGSTRÖM, *Bounds for the distance between nearby Jordan and Kronecker structures in a closure hierarchy*, J. Math. Sci. (N. Y.), 112 (2003), pp. 1765–1779.
- [9] E. ELMROTH, P. JOHANSSON, AND B. KÅGSTRÖM, *Computation and presentation of graphs displaying closure hierarchies of Jordan and Kronecker structures*, Numer. Linear Algebra Appl., 8 (2001), pp. 381–399.
- [10] M. EMBREE AND L. N. TREFETHEN, *Generalizing eigenvalue theorems to pseudospectra theorems*, SIAM J. Sci. Comput., 23 (2001), pp. 583–590.
- [11] V. FRAYSSÉ, M. GUEURY, F. NICOU, AND V. TOUMAZOU, *Spectral Portraits for Matrix Pencils*, Technical report TR/PA/96/19, CERFACS, Toulouse, France, 1996.
- [12] M. ELAD, P. MILANFAR, AND G. H. GOLUB, *Shape from moments—An estimation theory perspective*, IEEE Trans. Signal Process., 52 (2004), pp. 1814–1829.
- [13] G. H. GOLUB AND C. F. VAN LOAN, *Matrix Computations*, 3rd ed., The Johns Hopkins University Press, Baltimore, MD, 1996.
- [14] G. H. GOLUB AND H. VAN DER VORST, *Eigenvalue computation in the 20th century*, J. Comput. Appl. Math., 123 (2000), pp. 35–65.
- [15] M. GU, *New methods for estimating the distance to uncontrollability*, SIAM J. Matrix Anal. Appl., 21 (2000), pp. 989–1003.
- [16] B. KÅGSTRÖM, *The generalized singular value decomposition and the general $A - \lambda B$ problem*, BIT, 24 (1984), pp. 568–583.
- [17] B. KÅGSTRÖM, *RGSVD—An algorithm for computing the Kronecker structure and reducing subspaces of singular $A - \lambda B$ pencils*, SIAM J. Sci. Statist. Comput., 7 (1986), pp. 185–211.
- [18] R. LIPPERT AND A. EDELMAN, *The computation and sensitivity of double eigenvalues*, in Advances in Computational Mathematics, Lecture Notes in Pure and Appl. Math. 202, Dekker, New York, 1999, pp. 353–393.
- [19] A. MALYSHEV, *On Wilkinson's Problem*, Tech. report 140, Department of Informatics, University of Bergen, Norway, 1997.

- [20] C. C. PAIGE AND M. A. SAUNDERS, *Towards a generalized singular value decomposition*, SIAM J. Numer. Anal., 18 (1981), pp. 398–405.
- [21] P. SPELLUCI AND W. M. HARTMANN, *A QR decomposition for matrix pencils*, BIT, 40 (1999), pp. 183–189.
- [22] G. W. STEWART, *Perturbation theory for rectangular matrix pencils*, Linear Algebra Appl., 208 (1994), pp. 297–301.
- [23] L. N. TREFETHEN, *Computation of pseudospectra*, Acta Numer., 8 (1999), pp. 247–295.
- [24] J. L. M. VAN DORSSELAER, *Pseudospectra for matrix pencils and stability of equilibria*, BIT, 37 (1997), pp. 833–845.
- [25] T. G. WRIGHT AND L. N. TREFETHEN, *Pseudospectra of rectangular matrices*, IMA J. Numer. Anal., 22 (2002), pp. 501–519.
- [26] T. G. WRIGHT AND L. N. TREFETHEN, *Large-scale computation of pseudospectra using ARPACK and eigs*, SIAM J. Sci. Comput., 23 (2001), pp. 591–605.

CONDITION NUMBERS OF GAUSSIAN RANDOM MATRICES*

ZIZHONG CHEN[†] AND JACK J. DONGARRA[†]

Abstract. Let $G_{m \times n}$ be an $m \times n$ real random matrix whose elements are independent and identically distributed standard normal random variables, and let $\kappa_2(G_{m \times n})$ be the 2-norm condition number of $G_{m \times n}$. We prove that, for any $m \geq 2$, $n \geq 2$, and $x \geq |n - m| + 1$, $\kappa_2(G_{m \times n})$ satisfies $\frac{1}{\sqrt{2\pi}}(c/x)^{|n-m|+1} < P(\frac{\kappa_2(G_{m \times n})}{n/(|n-m|+1)} > x) < \frac{1}{\sqrt{2\pi}}(C/x)^{|n-m|+1}$, where $0.245 \leq c \leq 2.000$ and $5.013 \leq C \leq 6.414$ are universal positive constants independent of m , n , and x . Moreover, for any $m \geq 2$ and $n \geq 2$, $E(\log \kappa_2(G_{m \times n})) < \log \frac{n}{|n-m|+1} + 2.258$. A similar pair of results for complex Gaussian random matrices is also established.

Key words. condition number, eigenvalues, random matrices, singular values, Wishart distribution

AMS subject classifications. 15A52, 15A12

DOI. 10.1137/040616413

1. Introduction. In [5], Edelman obtained the limiting distributions and the limiting expected logarithms of the condition numbers of random rectangular matrices whose elements are independent and identically distributed standard normal random variables. The exact distributions of the condition numbers of $2 \times n$ matrices are also given in [5] by Edelman.

However, in the study of real-number and complex-number error correction codes based on random matrices [3] and their applications in fault tolerant high performance computing [4], in order to estimate the numerical stability and reliability of our coding schemes, we need to estimate the probabilities that the condition numbers of small random rectangular matrices are large. For example, what is the probability that the condition number of a 10×5 random matrix is larger than 10^2 ?

In this paper, we investigate the tails of the condition number distributions of random rectangular matrices whose elements are independent and identically distributed standard normal real or complex random variables. We establish upper and lower bounds for the tails of the condition number distributions of these matrices. Upper bounds for the expected logarithms of the condition numbers of these matrices are also given.

Based on our results, for random rectangular matrices whose elements are independent and identically distributed standard normal real or complex random variables, we are able to estimate the probabilities that their condition numbers are large. For example, based on our results, we are able to tell, for a 10×5 real random matrix whose elements are independent and identically distributed standard normal random

*Received by the editors October 5, 2004; accepted for publication (in revised form) by P. C. Hansen April 20, 2005; published electronically November 23, 2005. This research was supported in part by the Applied Mathematical Sciences Research Program of the Office of Mathematical, Information, and Computational Sciences, U.S. Department of Energy under contract DE-AC05-00OR22725 with UT-Battelle, LLC. The U.S. Government retains a nonexclusive, royalty-free license to publish or reproduce the published form of this contribution, or allow others to do so, for U.S. Government purposes. Copyright is owned by SIAM to the extent not limited by these rights.

<http://www.siam.org/journals/simax/27-3/61641.html>

[†]Department of Computer Science, The University of Tennessee, 203 Claxton Complex, 1122 Volunteer Boulevard, Knoxville, TN 37996-3450 (zchen@cs.utk.edu, dongarra@cs.utk.edu).

variables, the probability that the condition number is larger than 10^2 is less than 6×10^{-7} .

Our main results for the 2-norm condition number κ of an $m \times n$ real random matrix whose elements are independent and identically distributed standard normal random variables are

$$\frac{1}{\sqrt{2\pi}} \left(\frac{c}{x}\right)^{|n-m|+1} < P\left(\frac{\kappa}{n/(|n-m|+1)} > x\right) < \frac{1}{\sqrt{2\pi}} \left(\frac{C}{x}\right)^{|n-m|+1}$$

and

$$E(\log \kappa) < \log \frac{n}{|n-m|+1} + 2.258,$$

where $0.245 \leq c \leq 2.000$ and $5.013 \leq C \leq 6.414$ are universal positive constants independent of m , n , and x , and $m \geq 2$, $n \geq 2$, and $x \geq |n-m|+1$.

For an $m \times n$ complex random matrix whose elements are independent and identically distributed standard normal random variables, our main results for the 2-norm condition number κ are

$$\frac{1}{2\pi} \left(\frac{c}{x}\right)^{2(|n-m|+1)} < P\left(\frac{\kappa}{n/(|n-m|+1)} > x\right) < \frac{1}{2\pi} \left(\frac{C}{x}\right)^{2(|n-m|+1)}$$

and

$$E(\log \kappa) < \log \frac{n}{|n-m|+1} + 2.240,$$

where $0.319 \leq c \leq 2.000$ and $5.013 \leq C \leq 6.298$ are universal positive constants independent of m , n , and x , and $m \geq 2$, $n \geq 2$, and $x \geq |n-m|+1$.

After finishing the manuscript of this paper, we communicated with Edelman and learned that a similar problem was also being studied independently by Edelman and Sutton [7]. After simple formatting, the upper bounds in both papers actually can be unified into the same format,

$$P(\kappa > x) \leq C(m, n, \beta) \left(\frac{1}{x}\right)^{\beta(|n-m|+1)},$$

where $\beta = 1$ for real random matrices, $\beta = 2$ for complex random matrices, and $C(m, n, \beta)$ is a function of m , n , and β . However, the function $C(m, n, \beta)$ in the two papers do take very different forms and imply very different meanings.

On one hand, the bounds in [7] are asymptotically tight as $x \rightarrow \infty$, while the bounds in this paper are not. On the other hand, the bounds in this paper involve only elementary functions. Hence they are much simpler than the asymptotically tight bounds in [7] which involve high order moments of the largest eigenvalues of Wishart matrices. Although for the special case of large square random matrices, simple estimations for $C(m, n, \beta)$ are given in [7], for general rectangular matrices, no simple estimation is available.

It is well known that the joint eigenvalue density function of a Wishart matrix has a closed form expression [9]. Therefore, $P(\kappa > x)$ can actually be expressed *accurately* as a high-dimensional integration of this joint eigenvalue density function. One of the key aspects to estimate $P(\kappa > x)$ is to find a simple-to-use estimation of this accurate (but not simple-to-use) high-dimensional integral expression. This paper

is meaningful in that it finds out such a simple-to-use estimation by giving out simple upper and lower bounds which involve only elementary functions. We refer interested readers to [7] for more accurate asymptotically tight bounds and other related bounds for the tails of the condition numbers of general β -Laguerre ensembles.

Above and in what follows in this paper, the constants C and c denote universal positive constants independent of m , n , and x ; however, identical symbols may represent different numbers in different place.

2. Preliminaries and basic facts. Let X be an $m \times n$ matrix. If $\sigma_1 \geq \sigma_2 \geq \dots \geq \sigma_p$, where $p = \min\{m, n\}$, are the p singular values of X , then the 2-norm condition number of X is

$$\kappa_2(X) = \frac{\sigma_1}{\sigma_p}.$$

For any $m \times n$ matrix X , X^T is an $n \times m$ matrix and $\kappa_2(X) = \kappa_2(X^T)$. So, without loss of generality, in discussing the condition numbers of random matrices, it is enough to consider only random matrices with no more rows than columns. Therefore, when we speak of an $m \times n$ matrix, we will assume $m \leq n$ in the rest of this paper.

Let $G_{m \times n}$ be an $m \times n$ real random matrix whose elements are independent and identically distributed standard normal random variables. Let $W_{m,n}$ denote the $m \times m$ random matrix $G_{m \times n} G_{m \times n}^T$. $W_{m,n}$ is the well-known Wishart matrix named after John Wishart, who had first studied its distribution.

Similar to [5], in this paper, we will study the condition number of $G_{m \times n}$ through investigating the eigenvalues of the Wishart matrix $W_{m,n}$. The following lemma establishes a simple relationship between the condition number of $G_{m \times n}$ and the eigenvalues of $W_{m,n}$.

PROPOSITION 2.1. *If λ_{max} is the largest eigenvalue of $W_{m,n}$, and λ_{min} is the smallest eigenvalue of $W_{m,n}$, then the 2-norm condition number of $G_{m \times n}$ satisfies*

$$\kappa_2(G_{m \times n}) = \sqrt{\frac{\lambda_{max}}{\lambda_{min}}}.$$

Remarkably enough, the exact joint probability density function for the m eigenvalues of the Wishart matrix $W_{m,n}$ can be written down in a closed form [9].

LEMMA 2.2. *If $\lambda_1 \geq \dots \geq \lambda_m$ are the m eigenvalues of $W_{m,n}$, then the joint probability density function of $\lambda_1 \geq \dots \geq \lambda_m$ is*

$$(2.1) \quad f(x_1, \dots, x_m) = K_{m,n} e^{-\frac{1}{2} \sum_{i=1}^m x_i} \prod_{i=1}^m x_i^{\frac{1}{2}(n-m-1)} \prod_{i=1}^{m-1} \prod_{j=i+1}^m (x_i - x_j),$$

where

$$(2.2) \quad K_{m,n}^{-1} = \left(\frac{2^n}{\pi}\right)^{m/2} \prod_{i=1}^m \Gamma\left(\frac{n-m+i}{2}\right) \Gamma\left(\frac{i}{2}\right).$$

Let $N(0, 1)$ denote the standard normal distribution. Let $\tilde{N}(0, 1)$ denote the distribution of $u + iv$, where u and v are independent and identically distributed $N(0, 1)$ random variables, and $i = \sqrt{-1}$. Let $\tilde{G}_{m \times n}$ be an $m \times n$ complex random matrix whose elements are independent and identically distributed $\tilde{N}(0, 1)$ random

variables. Let $\widetilde{W}_{m,n}$ denote the $m \times m$ random matrix $\widetilde{G}_{m \times n} \widetilde{G}_{m \times n}^H$. In the literature, $\widetilde{W}_{m,n}$ is called the complex Wishart matrix.

Similar to the real case, there is also a simple relationship between the condition number of $\widetilde{G}_{m \times n}$ and the eigenvalues of $\widetilde{W}_{m,n}$.

PROPOSITION 2.3. *If $\widetilde{\lambda}_{max}$ is the largest eigenvalue of $\widetilde{W}_{m,n}$, and $\widetilde{\lambda}_{min}$ is the smallest eigenvalue of $\widetilde{W}_{m,n}$, then the 2-norm condition number of $\widetilde{G}_{m \times n}$ satisfies*

$$\kappa_2(\widetilde{G}_{m \times n}) = \sqrt{\frac{\widetilde{\lambda}_{max}}{\widetilde{\lambda}_{min}}}.$$

Like the real case, the exact joint probability density function for the m eigenvalues of the complex Wishart matrix $\widetilde{W}_{m,n}$ can also be written down in a closed form [9].

LEMMA 2.4. *If $\widetilde{\lambda}_1 \geq \cdots \geq \widetilde{\lambda}_m$ are the m eigenvalues of $\widetilde{W}_{m,n}$, then the joint probability density function of $\widetilde{\lambda}_1 \geq \cdots \geq \widetilde{\lambda}_m$ is*

$$(2.3) \quad \widetilde{f}(x_1, \dots, x_m) = \widetilde{K}_{m,n} e^{-\frac{1}{2} \sum_{i=1}^m x_i} \prod_{i=1}^m x_i^{n-m} \prod_{i=1}^{m-1} \prod_{j=i+1}^m (x_i - x_j)^2,$$

where

$$(2.4) \quad \widetilde{K}_{m,n}^{-1} = 2^{mn} \prod_{i=1}^m \Gamma(n - m + i) \Gamma(i).$$

In the process of deriving our upper and lower bounds for the tails of the condition number distributions, some bounds for Gamma and incomplete Gamma functions are very useful.

LEMMA 2.5. *Assume $a > 0$ and $b > 0$. If $t \leq \frac{b}{a}$, then*

$$\int_0^t e^{-ax} x^b dx \leq e^{-at} t^{b+1}.$$

Proof. Let $f(t) = \int_0^t e^{-ax} x^b dx - e^{-at} t^{b+1}$. Then $f'(t) = e^{-at} t^b (1 + at - (b+1))$. So $f(t)$ decreases on $[0, \frac{b}{a}]$ and increases on $[\frac{b}{a}, \infty)$. Since $f(0) = 0$ and $f(\infty) = \int_0^\infty e^{-ax} x^b dx > 0$, if $t \leq \frac{b}{a}$, then $f(t) < 0$. Therefore, if $t \leq \frac{b}{a}$, then $\int_0^t e^{-ax} x^b dx \leq e^{-at} t^{b+1}$. $\square$

LEMMA 2.6. *Assume $a > 0$, $b > 0$, and $k > \frac{1}{a}$. If $t \geq \frac{kb}{ka-1}$, then*

$$\int_t^\infty e^{-ax} x^b dx \leq k e^{-at} t^b.$$

Proof. Let $f(t) = \int_t^\infty e^{-ax} x^b dx - k e^{-at} t^b$. Then $f'(t) = e^{-at} t^b (-1 + ka - \frac{kb}{t})$. So $f(t)$ decreases on $[0, \frac{kb}{ka-1}]$ and increases on $[\frac{kb}{ka-1}, \infty)$, since $f(0) = \int_0^\infty e^{-ax} x^b dx > 0$ and $f(\infty) = 0$. So, if $t \geq \frac{kb}{ka-1}$, then $f(t) < 0$. Therefore, if $t \geq \frac{kb}{ka-1}$, then $\int_t^\infty e^{-ax} x^b dx \leq k e^{-at} t^b$. $\square$

LEMMA 2.7. *If $\Gamma(x) = \int_0^\infty e^{-t} t^{x-1} dt$, where $x > 0$, then*

$$(2.5) \quad \sqrt{2\pi} x^{x+\frac{1}{2}} e^{-x} < \Gamma(x+1) < \sqrt{2\pi} x^{x+\frac{1}{2}} e^{-x+\frac{1}{12x}}$$

and

$$(2.6) \quad \Gamma\left(x + \frac{1}{2}\right) < \Gamma(x)\sqrt{x}.$$

Proof. Inequality (2.5) follows straightforwardly from 6.1.38 in [1], and (2.6) can be obtained from the answer to Problem 9.60 in [8]. $\square$

3. Bounds for eigenvalue densities of Wishart matrices. In this section, we will prove some bounds for the probability density functions of the eigenvalues of Wishart matrices. These bounds are very useful in the derivation of the bounds for the tails of the condition number distributions.

Let λ_{max} denote the largest eigenvalue of $W_{m,n}$, and let λ_{min} denote the smallest eigenvalue of $W_{m,n}$. In the following lemma, we prove an upper bound for the joint probability density function of λ_{max} and λ_{min} .

LEMMA 3.1. *Let $f_{\lambda_{max}, \lambda_{min}}(x, y)$ denote the joint probability density function of λ_{max} and λ_{min} . Then $f_{\lambda_{max}, \lambda_{min}}(x, y)$ satisfies*

$$(3.1) \quad f_{\lambda_{max}, \lambda_{min}}(x, y) \leq C_{m,n} e^{-\frac{1}{2}(x+y)} x^{\frac{1}{2}(n+m-3)} y^{\frac{1}{2}(n-m-1)},$$

where

$$(3.2) \quad C_{m,n} = \frac{1}{4\Gamma(m-1)\Gamma(n-m+1)}.$$

Proof. Let $R_{x,y} = \{(x_2, x_3, \dots, x_{m-1}) : x \geq x_2 \geq \dots \geq x_{m-1} \geq y\} \subseteq R^{m-2}$. From the joint probability density function of the m eigenvalues of $W_{m,n}$ in Lemma 2.2, we have

$$(3.3) \quad \begin{aligned} f_{\lambda_{max}, \lambda_{min}}(x, y) &= \int_{R_{x,y}} f(x, x_2, \dots, x_{m-1}, y) dx_2 dx_3 \cdots dx_{m-1} \\ &= K_{m,n} e^{-\frac{1}{2}(x+y)} x^{\frac{1}{2}(n-m-1)} y^{\frac{1}{2}(n-m-1)} \\ &\quad \times \int_{R_{x,y}} e^{-\frac{1}{2} \sum_{i=2}^{m-1} x_i} \prod_{i=2}^{m-1} x_i^{\frac{1}{2}(n-m-1)} \\ &\quad \times (x-y) \prod_{i=2}^{m-1} (x-x_i)(x_i-y) \prod_{i=2}^{m-2} \prod_{j=i+1}^{m-1} (x_i-x_j) \prod_{i=2}^{m-1} dx_i. \end{aligned}$$

Let $R_{m-2} = \{(x_2, x_3, \dots, x_{m-1}) : x_2 \geq \dots \geq x_{m-1} \geq 0\}$. Then $R_{m-2} \subseteq R_{x,y}$. Note that, in (3.3), $x \geq x_i \geq y$ for $i = 2, 3, \dots, m-1$. Replacing $x-y$ and $x-x_i$ by x , x_i-y by x_i for $i = 2, 3, \dots, m-1$, and $R_{x,y}$ by R_{m-2} , we get

$$(3.4) \quad \begin{aligned} f_{\lambda_{max}, \lambda_{min}}(x, y) &\leq K_{m,n} e^{-\frac{1}{2}(x+y)} x^{\frac{1}{2}(n+m-3)} y^{\frac{1}{2}(n-m-1)} \\ &\quad \int_{R_{m-2}} e^{-\frac{1}{2} \sum_{i=2}^{m-1} x_i} \prod_{i=2}^{m-1} x_i^{\frac{1}{2}(n-m+1)} \prod_{i=2}^{m-2} \prod_{j=i+1}^{m-1} (x_i-x_j) \prod_{i=2}^{m-1} dx_i. \end{aligned}$$

Note that $f(x_1, x_2, \dots, x_m)$ in (2.1) is a probability density function. Therefore, for any $m \leq n$, we have

$$\int_{R_m} e^{-\frac{1}{2} \sum_{i=1}^m x_i} \prod_{i=1}^m x_i^{\frac{1}{2}(n-m-1)} \prod_{i=1}^{m-1} \prod_{j=i+1}^m (x_i-x_j) \prod_{i=1}^m dx_i = K_{m,n}^{-1},$$

where $R_m = \{x_1 \geq x_2 \geq \cdots \geq x_m \geq 0\} \subseteq R^m$. Therefore, we have

$$(3.5) \quad \int_{R_{m-2}} e^{-\frac{1}{2} \sum_{i=2}^{m-1} x_i} \prod_{i=2}^{m-1} x_i^{\frac{1}{2}(n-m+1)} \prod_{i=2}^{m-2} \prod_{j=i+1}^{m-1} (x_i - x_j) \prod_{i=2}^{m-1} dx_i = K_{m-2,n}^{-1}.$$

Substituting (3.5) into (3.4), we obtain

$$(3.6) \quad f_{\lambda_{max}, \lambda_{min}}(x, y) \leq \frac{K_{m,n}}{K_{m-2,n}} e^{-\frac{1}{2}(x+y)} x^{\frac{1}{2}(n+m-3)} y^{\frac{1}{2}(n-m-1)}.$$

From (2.2), we have

$$(3.7) \quad \begin{aligned} \frac{K_{m,n}}{K_{m-2,n}} &= \frac{\pi}{2^n} \frac{1}{\Gamma\left(\frac{m-1}{2}\right) \Gamma\left(\frac{m}{2}\right) \Gamma\left(\frac{n-m+1}{2}\right) \Gamma\left(\frac{n-m+2}{2}\right)} \\ &= \frac{1}{4\Gamma(m-1)\Gamma(n-m+1)}. \end{aligned}$$

Substituting (3.6) into (3.5), we get (3.1) and (3.2). $\square$

Let $\tilde{\lambda}_{max}$ denote the largest eigenvalue of $\tilde{W}_{m,n}$, and let $\tilde{\lambda}_{min}$ denote the smallest eigenvalue of $\tilde{W}_{m,n}$. Similar to the real case, in the following lemma, we give an upper bound for the joint probability density function of $\tilde{\lambda}_{max}$ and $\tilde{\lambda}_{min}$. The upper bound in the complex case can be proved using the same techniques used in the real case. Therefore, we omit the proof and only give the result here.

LEMMA 3.2. *Let $\tilde{f}_{\tilde{\lambda}_{max}, \tilde{\lambda}_{min}}(x, y)$ denote the joint probability density function of $\tilde{\lambda}_{max}$ and $\tilde{\lambda}_{min}$. Then $\tilde{f}_{\tilde{\lambda}_{max}, \tilde{\lambda}_{min}}(x, y)$ satisfies*

$$(3.8) \quad \tilde{f}_{\tilde{\lambda}_{max}, \tilde{\lambda}_{min}}(x, y) \leq \tilde{C}_{m,n} e^{-\frac{1}{2}(x+y)} x^{n+m-2} y^{n-m},$$

where

$$(3.9) \quad \tilde{C}_{m,n} = \frac{1}{2^{2n} \Gamma(m-1) \Gamma(m) \Gamma(n-m+1) \Gamma(n-m+2)}.$$

Bounds for the probability density functions of the smallest eigenvalues are also very useful in the derivation of the bounds for the tails of the condition number distributions. In the following lemma, we prove upper and lower bounds for the probability density function of the smallest eigenvalue of a real Wishart matrix.

LEMMA 3.3. *Let $f_{\lambda_{min}}(x)$ denote the probability density function of the smallest eigenvalue of $W_{m,n}$. Then $f_{\lambda_{min}}(x)$ satisfies*

$$(3.10) \quad L_{m,n} e^{-\frac{m}{2}x} x^{\frac{1}{2}(n-m-1)} \leq f_{\lambda_{min}}(x) \leq L_{m,n} e^{-\frac{1}{2}x} x^{\frac{1}{2}(n-m-1)},$$

where

$$(3.11) \quad L_{m,n} = \frac{2^{\frac{n-m-1}{2}} \Gamma\left(\frac{n+1}{2}\right)}{\Gamma\left(\frac{m}{2}\right) \Gamma(n-m+1)}.$$

Proof. Let $R_x = \{(x_1, x_2, \dots, x_{m-1}) : x_1 \geq \cdots \geq x_{m-1} \geq x\} \subseteq R^{m-1}$. From the joint probability density function of the eigenvalues of $W_{m,n}$ in Lemma 2.2, we have

$$f_{\lambda_{min}}(x) = \int_{R_x} f(x_1, x_2, \dots, x_{m-1}, x) dx_1 dx_2 \cdots dx_{m-1}$$

$$\begin{aligned}
 &= K_{m,n} e^{-\frac{1}{2}x} x^{\frac{1}{2}(n-m-1)} \int_{R_x} e^{-\frac{1}{2} \sum_{i=1}^{m-1} x_i} \prod_{i=1}^{m-1} x_i^{\frac{1}{2}(n-m-1)} \\
 &\quad \times \prod_{i=1}^{m-1} (x_i - x) \prod_{i=1}^{m-2} \prod_{j=i+1}^{m-1} (x_i - x_j) \prod_{i=1}^{m-1} dx_i.
 \end{aligned}$$

For the lower bound part, taking the transformation $y_i = x_i - x$, where $i = 1, 2, \dots, m-1$, we have

$$\begin{aligned}
 f_{\lambda_{min}}(x) &= K_{m,n} e^{-\frac{m}{2}x} x^{\frac{1}{2}(n-m-1)} \int_{R_y} e^{-\frac{1}{2} \sum_{i=1}^{m-1} y_i} \prod_{i=1}^{m-1} (y_i + x)^{\frac{1}{2}(n-m-1)} \\
 &\quad \times \prod_{i=1}^{m-1} y_i \prod_{i=1}^{m-2} \prod_{j=i+1}^{m-1} (y_i - y_j) \prod_{i=1}^{m-1} dy_i,
 \end{aligned}$$

where $R_y = \{y_1 \geq y_2 \geq \dots \geq y_{m-1} \geq 0\} \subseteq R^{m-1}$.

Replacing $y_i + x$ by y_i for $i = 1, 2, \dots, m-1$, we obtain

$$\begin{aligned}
 f_{\lambda_{min}}(x) &\geq K_{m,n} e^{-\frac{m}{2}x} x^{\frac{1}{2}(n-m-1)} \int_{R_y} e^{-\frac{1}{2} \sum_{i=1}^{m-1} y_i} \prod_{i=1}^{m-1} y_i^{\frac{1}{2}(n-m+1)} \\
 &\quad \times \prod_{i=1}^{m-2} \prod_{j=i+1}^{m-1} (y_i - y_j) \prod_{i=1}^{m-1} dy_i.
 \end{aligned}$$

Note that

$$\int_{R_y} e^{-\frac{1}{2} \sum_{i=1}^{m-1} y_i} \prod_{i=1}^{m-1} y_i^{\frac{1}{2}(n-m+1)} \prod_{i=1}^{m-2} \prod_{j=i+1}^{m-1} (y_i - y_j) \prod_{i=1}^{m-1} dy_i = K_{m-1,n+1}^{-1}.$$

Therefore, we obtain

$$(3.12) \quad f_{\lambda_{min}}(x) \geq \frac{K_{m,n}}{K_{m-1,n+1}} e^{-\frac{m}{2}x} x^{\frac{1}{2}(n-m-1)}.$$

For the upper bound part, from [6], we have

$$(3.13) \quad f_{\lambda_{min}}(x) \leq \frac{K_{m,n}}{K_{m-1,n+1}} e^{-\frac{1}{2}x} x^{\frac{1}{2}(n-m-1)}.$$

From (2.2), we have

$$\begin{aligned}
 (3.14) \quad \frac{K_{m,n}}{K_{m-1,n+1}} &= \frac{\sqrt{\pi} \left(\frac{1}{2}\right)^{\frac{n-m+1}{2}} \Gamma\left(\frac{n+1}{2}\right)}{\Gamma\left(\frac{m}{2}\right) \Gamma\left(\frac{n-m+1}{2}\right) \Gamma\left(\frac{n-m+2}{2}\right)} \\
 &= \frac{2^{\frac{n-m-1}{2}} \Gamma\left(\frac{n+1}{2}\right)}{\Gamma\left(\frac{m}{2}\right) \Gamma(n-m+1)}.
 \end{aligned}$$

Substituting (3.14) into (3.13) and (3.12), we get (3.10) and (3.11). $\square$

Similar to the real case, in the following lemma, we give upper and lower bounds for the probability density function of the smallest eigenvalue $\tilde{\lambda}_{min}$ of a complex

Wishart matrix. These bounds can be proved using the same techniques used in the real case. Therefore, we omit the proof and only give the result here.

LEMMA 3.4. *Let $\tilde{f}_{\lambda_{\min}}(x)$ denote the probability density function of the smallest eigenvalue of $\tilde{W}_{m,n}$. Then $\tilde{f}_{\lambda_{\min}}(x)$ satisfies*

$$(3.15) \quad \tilde{L}_{m,n} e^{-\frac{m}{2}x} x^{n-m} \leq \tilde{f}_{\lambda_{\min}}(x) \leq \tilde{L}_{m,n} e^{-\frac{1}{2}x} x^{n-m},$$

where

$$(3.16) \quad \tilde{L}_{m,n} = \frac{\Gamma(n+1)}{2^{n-m+1} \Gamma(m) \Gamma(n-m+1) \Gamma(n-m+2)}.$$

4. The upper bounds for the distribution tails. In this section, we will derive the upper bounds for the tails of the condition number distributions of random rectangular matrices whose elements are independent and identically distributed standard normal random variables. Our main results are Theorem 4.5 for real random matrices and Theorem 4.6 for complex random matrices.

LEMMA 4.1. *For any $A > 0$, $x > 0$, and $n \geq m \geq 2$, the largest eigenvalue $\lambda_{\max}$ and the smallest eigenvalue $\lambda_{\min}$ of $W_{m,n}$ satisfy*

$$P\left(\frac{\lambda_{\max}}{\lambda_{\min}} > x^2, \lambda_{\min} \leq \frac{A^2 n}{x^2}\right) < \frac{1}{\Gamma(n-m+2)} \left(\frac{An}{x}\right)^{n-m+1}.$$

Proof. From the upper bound for the probability density function of $\lambda_{\min}$ in Lemma 3.2, we have

$$\begin{aligned} P\left(\frac{\lambda_{\max}}{\lambda_{\min}} > x^2, \lambda_{\min} \leq \frac{A^2 n}{x^2}\right) &< P\left(\lambda_{\min} \leq \frac{A^2 n}{x^2}\right) \\ &= \int_0^{\frac{A^2 n}{x^2}} f_{\lambda_{\min}}(t) dt < L_{m,n} \int_0^{\frac{A^2 n}{x^2}} t^{\frac{1}{2}(n-m-1)} dt \\ &= \frac{\Gamma\left(\frac{n+1}{2}\right)}{\Gamma\left(\frac{m}{2}\right) \left(\frac{n}{2}\right)^{\frac{n-m+1}{2}}} \frac{1}{\Gamma(n-m+2)} \left(\frac{An}{x}\right)^{n-m+1}. \end{aligned}$$

Since $m \leq n$, by applying (2.6) repeatedly, we can prove

$$\Gamma\left(\frac{m}{2}\right) \left(\frac{n}{2}\right)^{\frac{n-m+1}{2}} > \Gamma\left(\frac{n+1}{2}\right).$$

Therefore, we have

$$P\left(\frac{\lambda_{\max}}{\lambda_{\min}} > x^2, \lambda_{\min} \leq \frac{A^2 n}{x^2}\right) < \frac{1}{\Gamma(n-m+2)} \left(\frac{An}{x}\right)^{n-m+1}. \quad \square$$

Similar to real random matrices, for complex random matrices, we have the following lemma, which can be proved using the same techniques as those for Lemma 4.1, so we will omit the proof and only give the result.

LEMMA 4.2. *For any $A > 0$, $x > 0$, and $n \geq m \geq 2$, the largest eigenvalue $\tilde{\lambda}_{\max}$ and the smallest eigenvalue $\tilde{\lambda}_{\min}$ of $\tilde{W}_{m,n}$ satisfy*

$$P\left(\frac{\tilde{\lambda}_{\max}}{\tilde{\lambda}_{\min}} > x^2, \tilde{\lambda}_{\min} \leq \frac{A^2 n^2}{x^2}\right) < \frac{1}{\Gamma(n-m+2)^2} \left(\frac{A^2 n^2}{2x^2}\right)^{n-m+1}.$$

The proof of the following lemma is based on the upper bound for the joint probability density function of λ_{max} and λ_{min} in Lemma 3.1 and the upper bound of the incomplete Gamma function in Lemma 2.6.

LEMMA 4.3. *For any $A \geq 2.32$, $x > 0$, and $n \geq m \geq 2$, the largest eigenvalue λ_{max} and the smallest eigenvalue λ_{min} of $W_{m,n}$ satisfy*

$$P\left(\frac{\lambda_{max}}{\lambda_{min}} > x^2, \lambda_{min} > \frac{A^2 n}{x^2}\right) < 0.017 \frac{1}{\Gamma(n-m+2)} \left(\frac{An}{x}\right)^{n-m+1}.$$

Proof. From the upper bound for the joint probability density function of λ_{max} and λ_{min} in Lemma 3.1, we have

$$\begin{aligned} P\left(\frac{\lambda_{max}}{\lambda_{min}} > x^2, \lambda_{min} > \frac{A^2 n}{x^2}\right) &= \int_{\frac{A^2 n}{x^2}}^{\infty} \int_{tx^2}^{\infty} f_{\lambda_{max}, \lambda_{min}}(s, t) ds dt \\ &< \int_{\frac{A^2 n}{x^2}}^{\infty} \int_{tx^2}^{\infty} C_{m,n} e^{-\frac{1}{2}t} t^{\frac{1}{2}(n-m-1)} e^{-\frac{1}{2}s} s^{\frac{1}{2}(n+m-3)} ds dt. \end{aligned}$$

Taking the transform $u = tx^2$, we have

$$\begin{aligned} P\left(\frac{\lambda_{max}}{\lambda_{min}} > x^2, \lambda_{min} > \frac{A^2 n}{x^2}\right) &= C_{m,n} \left(\frac{1}{x}\right)^{n-m+1} \int_{A^2 n}^{\infty} e^{-\frac{u}{2x^2}} u^{\frac{1}{2}(n-m-1)} \\ &\quad \times \left(\int_u^{\infty} e^{-\frac{1}{2}s} s^{\frac{1}{2}(n+m-3)} ds\right) du. \end{aligned}$$

According to Lemma 2.6, with $k = 4$, if $u \geq 2(n+m-3)$, then

$$\int_u^{\infty} e^{-\frac{1}{2}s} s^{\frac{1}{2}(n+m-3)} ds \leq 4e^{-\frac{1}{2}u} u^{\frac{1}{2}(n+m-3)}.$$

Since $A \geq 2.32$ and $n \geq m$, hence, $u \geq A^2 n \geq 2(n+m-3)$. Therefore, we have

$$\begin{aligned} P\left(\frac{\lambda_{max}}{\lambda_{min}} > x^2, \lambda_{min} > \frac{A^2 n}{x^2}\right) &\leq 4C_{m,n} \left(\frac{1}{x}\right)^{n-m+1} \int_{A^2 n}^{\infty} e^{-\frac{u}{2x^2} - \frac{1}{2}u} u^{n-2} du \\ &\leq 4C_{m,n} \left(\frac{1}{x}\right)^{n-m+1} \int_{A^2 n}^{\infty} e^{-\frac{1}{2}u} u^{n-2} du. \end{aligned}$$

Since $A \geq 2.32$, so $A^2 n \geq 4(n-2)$. Applying Lemma 2.6 again, we have

$$\begin{aligned} P\left(\frac{\lambda_{max}}{\lambda_{min}} > x^2, \lambda_{min} > \frac{A^2 n}{x^2}\right) &\leq 16C_{m,n} e^{-\frac{1}{2}A^2 n} A^{2n-4} n^{n-2} \left(\frac{1}{x}\right)^{n-m+1} \\ &= \frac{4e^{-\frac{A^2 n}{2}} A^{2n-4} n^{m-3}}{\Gamma(m-1)\Gamma(n-m+1)} \left(\frac{n}{x}\right)^{n-m+1} \\ (4.1) \quad &\leq \frac{4e^{(2\ln A - \frac{A^2}{2})n}}{A^4} \frac{n^{m-2}}{\Gamma(m-1)} \frac{1}{\Gamma(n-m+2)} \left(\frac{n}{x}\right)^{n-m+1}. \end{aligned}$$

Note that, for any $2 \leq m \leq n$, it can be proved that

$$(4.2) \quad \frac{n^{m-2}}{\Gamma(m-1)} < \frac{e^n}{\sqrt{4\pi}}.$$

Substituting (4.2) into (4.1), we have

$$P\left(\frac{\lambda_{max}}{\lambda_{min}} > x^2, \lambda_{min} > \frac{A^2 n}{x^2}\right) \leq \frac{4e^{(2 \ln A - \frac{A^2}{2} + 1)n}}{\sqrt{4\pi}A^4} \frac{1}{\Gamma(n-m+2)} \left(\frac{n}{x}\right)^{n-m+1}.$$

Since $A \geq 2.32$, we have

$$e^{(2 \ln A - \frac{A^2}{2} + 1)n} < 1.$$

Therefore, when $A \geq 2.32$, we have

$$\begin{aligned} P\left(\frac{\lambda_{max}}{\lambda_{min}} > x^2, \lambda_{min} > \frac{A^2 n}{x^2}\right) &\leq \frac{4}{\sqrt{4\pi}A^4} \frac{1}{\Gamma(n-m+2)} \left(\frac{n}{x}\right)^{n-m+1} \\ &\leq 0.017 \frac{1}{\Gamma(n-m+2)} \left(\frac{An}{x}\right)^{n-m+1}. \quad \square \end{aligned}$$

Similar to real random matrices, for complex random matrices, we have the following lemma, which can be proved using the same techniques as those for Lemma 4.3, so we will omit the proof and only give the result.

LEMMA 4.4. *For any $A \geq 3.2735$, $x > 0$, and $n \geq m \geq 2$, the largest eigenvalue $\tilde{\lambda}_{max}$ and the smallest eigenvalue $\tilde{\lambda}_{min}$ of $\tilde{W}_{m,n}$ satisfy*

$$P\left(\frac{\tilde{\lambda}_{max}}{\tilde{\lambda}_{min}} > x^2, \tilde{\lambda}_{min} > \frac{A^2 n}{x^2}\right) < 0.0016 \frac{1}{\Gamma(n-m+2)^2} \left(\frac{A^2 n^2}{2x}\right)^{n-m+1}.$$

We are now prepared to prove our first main result about the condition numbers of real random matrices whose elements are independent and identically distributed standard normal random variables.

THEOREM 4.5. *For any $n \geq m \geq 2$ and $x \geq n - m + 1$, the 2-norm condition number of $G_{m \times n}$ satisfies*

$$(4.3) \quad P\left(\frac{\kappa_2(G_{m \times n})}{n/(n-m+1)} > x\right) < \frac{1}{\sqrt{2\pi}} \left(\frac{C}{x}\right)^{n-m+1},$$

where $C \leq 6.414$ is a universal positive constant independent of m , n , and x .

Proof. For any $L > 0$, inspired by [2], we first break down $P(\kappa_2(G_{m \times n}) > x)$ into two parts.

$$\begin{aligned} P(\kappa_2(G_{m \times n}) > x) &= P\left(\frac{\lambda_{max}}{\lambda_{min}} > x^2\right) \\ &= P\left(\frac{\lambda_{max}}{\lambda_{min}} > x^2, \lambda_{min} \leq \frac{L^2 n}{x^2}\right) + P\left(\frac{\lambda_{max}}{\lambda_{min}} > x^2, \lambda_{min} > \frac{L^2 n}{x^2}\right). \end{aligned}$$

Let $L = 2.32$. Then based on Lemmas 4.1 and 4.3, we can get

$$\begin{aligned} P(\kappa_2(G_{m \times n}) > x) &< \frac{1}{\Gamma(n-m+2)} \left(\frac{Ln}{x}\right)^{n-m+1} \\ &\quad + 0.017 \frac{1}{\Gamma(n-m+2)} \left(\frac{Ln}{x}\right)^{n-m+1} \\ &< \frac{1}{\Gamma(n-m+2)} \left(\frac{1.017Ln}{x}\right)^{n-m+1}. \end{aligned}$$

Note that, from Lemma 2.7, we have

$$\Gamma(n - m + 2) > \sqrt{2\pi(n - m + 1)}(n - m + 1)^{n-m+1}e^{-(n-m+1)}.$$

Therefore, we have

$$P(\kappa_2(G_{m \times n}) > x) < \frac{1}{\sqrt{2\pi(n - m + 1)}} \left(\frac{1.017eL \frac{n}{n-m+1}}{x} \right)^{n-m+1}.$$

Therefore,

$$\begin{aligned} P\left(\frac{\kappa_2(G_{m \times n})}{n/(n - m + 1)} > x\right) &< \frac{1}{\sqrt{2\pi(n - m + 1)}} \left(\frac{1.017eL}{x} \right)^{n-m+1} \\ &< \frac{1}{\sqrt{2\pi}} \left(\frac{6.414}{x} \right)^{n-m+1}. \end{aligned}$$

Let $C = 6.414$. Then we get (4.3). $\square$

Remark. 1. The upper bound in Theorem 4.5 is for arbitrary $n \geq m \geq 2$ and $x \geq n - m + 1$. For some special case of m and n , a more precise upper bound can be obtained. For example, for the special case of real random $2 \times n$ matrices, based on the exact probability density function of $\kappa_2(G_{2 \times n})$ in [5], we can get

$$P(\kappa_2(G_{2 \times n}) > x) = \left(\frac{2x}{x^2 + 1} \right)^{n-1} < \left(\frac{2}{x} \right)^{n-1}.$$

2. For the special case of real random $m \times m$ matrices, where $m \geq 3$, it has been proved in [2] that

$$(4.4) \quad P(\kappa_2(G_{m \times m}) > m \cdot x) < \frac{C'}{x},$$

where $C' \leq 5.60$ is a universal positive constant independent of x and m .

In Theorem 4.5, if we take $m = n$, then we have

$$P(\kappa_2(G_{m \times m}) > m \cdot x) < \frac{2.60}{x},$$

which is consistent with (4.4) except that we improved the upper bound for the constant C' from 5.60 to 2.60. From the following (4.5), we know that the constant C' in (4.4) actually must at least be 2.

3. For the special case of large real random $m \times m$ matrices, it has been proved in [5] that

$$\lim_{m \rightarrow \infty} P\left(\frac{\kappa_2(G_{m \times m})}{m} < x\right) = e^{-\frac{2}{x} - \frac{2}{x^2}}.$$

Therefore, we have

$$(4.5) \quad \lim_{m \rightarrow \infty} P\left(\frac{\kappa_2(G_{m \times m})}{m} > x\right) = 1 - e^{-\frac{2}{x} - \frac{2}{x^2}} \sim \frac{2}{x}$$

as $x \rightarrow \infty$. Hence, the smallest possible universal constant C in Theorem 4.5 must be no smaller than $2\sqrt{2\pi}$. Therefore, the universal constant C in Theorem 4.5 actually must satisfy

$$(4.6) \quad C \geq 2\sqrt{2\pi} \approx 5.013.$$

Similar to real random matrices, for complex random matrices, we have the following theorem, which can be proved using the same techniques as those for Theorem 4.5, so we will omit the proof and only give the result.

THEOREM 4.6. *For any $n \geq m \geq 2$ and $x \geq n - m + 1$, the 2-norm condition number of $\tilde{G}_{m \times n}$ satisfies*

$$P\left(\frac{\kappa_2(\tilde{G}_{m \times n})}{n/(n-m+1)} > x\right) < \frac{1}{2\pi} \left(\frac{\tilde{C}}{x}\right)^{2(n-m+1)},$$

where $\tilde{C} \leq 6.298$ is a universal positive constant independent of x, m , and n .

5. The lower bounds for the distribution tails. In this section, we will prove the lower bounds for the tails of the condition number distributions of random rectangular matrices whose elements are independent and identically distributed standard normal random variables. Our main results are Theorem 5.5 for real random matrices and Theorem 5.6 for complex random matrices.

LEMMA 5.1. *For any $B > 0$, $x > 0$, and $n \geq m \geq 2$, the smallest eigenvalue $\lambda_{\min}$ of $W_{m,n}$ satisfies*

$$P\left(\lambda_{\min} \leq \frac{B^2 n}{x^2}\right) > \sqrt{\frac{2e^{\frac{5}{6}}}{3}} e^{-\frac{B^2 m n}{2x^2}} \frac{1}{\Gamma(n-m+2)} \left(\frac{e^{-\frac{1}{2}} B n}{x}\right)^{n-m+1}.$$

Proof. From the lower bound for the probability density function of $\lambda_{\min}$ in Lemma 3.3, we have

$$\begin{aligned} P\left(\lambda_m \leq \frac{B^2 n}{x^2}\right) &= \int_0^{\frac{B^2 n}{x^2}} f(\lambda_m) d\lambda_m \\ &> \int_0^{\frac{B^2 n}{x^2}} L_{m,n} e^{-\frac{m}{2} \lambda_m} \lambda_m^{\frac{1}{2}(n-m-1)} d\lambda_m \\ &> L_{m,n} e^{-\frac{B^2 m n}{2x^2}} \int_0^{\frac{B^2 n}{x^2}} \lambda_m^{\frac{1}{2}(n-m-1)} d\lambda_m \\ &= L_{m,n} e^{-\frac{B^2 m n}{2x^2}} \frac{2n^{\frac{n-m+1}{2}}}{n-m+1} \left(\frac{B}{x}\right)^{n-m+1} \\ &= e^{-\frac{B^2 m n}{2x^2}} \frac{\Gamma\left(\frac{n+1}{2}\right)}{\Gamma\left(\frac{m}{2}\right) \left(\frac{n}{2}\right)^{\frac{n-m+1}{2}}} \frac{1}{\Gamma(n-m+2)} \left(\frac{Bn}{x}\right)^{n-m+1}. \end{aligned}$$

Note that

$$\frac{n+1}{2} \Gamma\left(\frac{n+1}{2}\right) > \sqrt{2\pi} \left(\frac{n+1}{2}\right)^{\frac{n+2}{2}} e^{-\frac{n+1}{2}}$$

and

$$\frac{m}{2} \Gamma\left(\frac{m}{2}\right) < \sqrt{2\pi} \left(\frac{m}{2}\right)^{\frac{m+1}{2}} e^{-\frac{m}{2} + \frac{1}{6m}}.$$

Therefore,

$$\begin{aligned} \frac{\Gamma\left(\frac{n+1}{2}\right)}{\Gamma\left(\frac{m}{2}\right)\left(\frac{n}{2}\right)^{\frac{n-m+1}{2}}} &> e^{-\frac{n-m+1}{2}-\frac{1}{6m}} \sqrt{\frac{(n+1)^n}{m^{m-1}n^{n-m+1}}} \\ &= e^{-\frac{n-m+1}{2}-\frac{1}{6m}} \sqrt{\frac{n^{n+1}(1+1/n)^{n+1}}{(n+1)m^{m-1}n^{n-m+1}}} \\ &> e^{-\frac{n-m+1}{2}-\frac{1}{6m}} \sqrt{\frac{ne}{n+1}}. \end{aligned}$$

Since $2 \leq m \leq n$, therefore, we have

$$\frac{\Gamma\left(\frac{n+1}{2}\right)}{\Gamma\left(\frac{m}{2}\right)\left(\frac{n}{2}\right)^{\frac{n-m+1}{2}}} > \sqrt{\frac{2e^{\frac{5}{6}}}{3}} e^{-\frac{n-m+1}{2}}.$$

Therefore, we have

$$P\left(\lambda_m \leq \frac{B^2 n}{x^2}\right) > \sqrt{\frac{2e^{\frac{5}{6}}}{3}} e^{-\frac{B^2 m n}{2x^2}} \frac{1}{\Gamma(n-m+2)} \left(\frac{e^{-\frac{1}{2}} B n}{x}\right)^{n-m+1}. \quad \square$$

Similar to real random matrices, we have the following lemma for complex random matrices. Lemma 5.2 can be proved using the same techniques as those for Lemma 5.1, so we will omit the proof and only give the result.

LEMMA 5.2. *For any $B > 0$, $x > 0$, and $2 \leq m \leq n$, the smallest eigenvalue $\tilde{\lambda}_{min}$ of $\tilde{W}_{m,n}$ satisfies*

$$P\left(\tilde{\lambda}_{min} \leq \frac{B^2 n}{x^2}\right) > e^{1-\frac{1}{12m}} e^{-\frac{B^2 m n}{2x^2}} \frac{1}{\Gamma(n-m+2)^2} \left(\frac{e^{-1} B^2 n^2}{2x^2}\right)^{n-m+1}.$$

The proof of the following lemma is based on the upper bound of the joint probability density function of λ_{max} and λ_{min} in Lemma 3.1 and the upper bound of the incomplete Gamma function in Lemma 2.5.

LEMMA 5.3. *For any $B \leq e^{-1.7}$, $x > 0$, and $2 \leq m \leq n$, the largest eigenvalue λ_{max} and the smallest eigenvalue λ_{min} of $W_{m,n}$ satisfy*

$$P\left(\lambda_{min} \leq \frac{B^2 n}{x^2}, \frac{\lambda_{max}}{\lambda_{min}} \leq x^2\right) < \frac{11B^{m-1}}{4\sqrt{4\pi}} \frac{1}{\Gamma(n-m+2)} \left(\frac{e^{-\frac{1}{2}} B n}{x}\right)^{n-m+1}.$$

Proof. From the upper bound for the joint probability density function of λ_{max} and λ_{min} in Lemma 3.1, we have

$$\begin{aligned} P\left(\lambda_{min} \leq \frac{B^2 n}{x^2}, \frac{\lambda_{max}}{\lambda_{min}} \leq x^2\right) &= \int_0^{\frac{B^2 n}{x^2}} \int_0^{tx^2} f_{\lambda_{max}, \lambda_{min}}(s, t) ds dt \\ &< C_{m,n} \int_0^{\frac{B^2 n}{x^2}} \int_0^{tx^2} e^{-\frac{1}{2}t} t^{\frac{1}{2}(n-m-1)} e^{-\frac{1}{2}s} s^{\frac{1}{2}(n+m-3)} ds dt. \end{aligned}$$

Taking the transform $u = tx^2$, we have

$$P\left(\lambda_{\min} \leq \frac{B^2 n}{x^2}, \frac{\lambda_{\max}}{\lambda_{\min}} \leq x^2\right) = C_{m,n} \left(\frac{1}{x}\right)^{n-m+1} \int_0^{B^2 n} e^{-\frac{u}{2x^2}} u^{\frac{1}{2}(n-m-1)} \\ \times \left(\int_0^u e^{-\frac{1}{2}s} s^{\frac{1}{2}(n+m-3)} ds\right) du.$$

According to Lemma 2.5, if $u \leq n + m - 3$, then

$$\int_0^u e^{-\frac{1}{2}s} s^{\frac{1}{2}(n+m-3)} ds \leq e^{-\frac{1}{2}u} u^{\frac{1}{2}(n+m-1)}.$$

Therefore, when $B \leq e^{-1.7}$, we have

$$P\left(\lambda_{\min} \leq \frac{B^2 n}{x^2}, \frac{\lambda_{\max}}{\lambda_{\min}} \leq x^2\right) \leq C_{m,n} \left(\frac{1}{x}\right)^{n-m+1} \int_0^{B^2 n} e^{-\frac{u}{2x^2} - \frac{1}{2}u} u^{n-1} du \\ \leq C_{m,n} \left(\frac{1}{x}\right)^{n-m+1} \int_0^{B^2 n} e^{-\frac{1}{2}u} u^{n-1} du.$$

Since $B \leq e^{-1.7}$, so $B^2 n \leq 2(n-1)$. Applying Lemma 2.5 again, we have

$$P\left(\lambda_{\min} \leq \frac{B^2 n}{x^2}, \frac{\lambda_{\max}}{\lambda_{\min}} \leq x^2\right) \leq C_{m,n} \left(\frac{1}{x}\right)^{n-m+1} e^{-\frac{B^2 n}{2}} B^{2n} n^n \\ = \frac{e^{-\frac{B^2 n}{2}} B^{n+m-1} n^{m-1}}{4\Gamma(m-1)\Gamma(n-m+1)} \left(\frac{Bn}{x}\right)^{n-m+1}.$$

From (4.2), we have

$$\frac{n^{m-2}}{\Gamma(m-1)} < \frac{e^n}{\sqrt{4\pi}}.$$

Therefore, we have

$$P\left(\lambda_{\min} \leq \frac{B^2 n}{x^2}, \frac{\lambda_{\max}}{\lambda_{\min}} \leq x^2\right) \leq \frac{e^n e^{-\frac{B^2 n}{2}} B^{n+m-1} n}{4\sqrt{4\pi}\Gamma(n-m+1)} \left(\frac{Bn}{x}\right)^{n-m+1} \\ \leq \frac{B^{m-1} n(n-m+1) e^{\frac{3}{2}n} e^{-\frac{B^2 n}{2}} B^n}{4\sqrt{4\pi}} \\ \times \frac{1}{\Gamma(n-m+2)} \left(\frac{e^{-\frac{1}{2}} Bn}{x}\right)^{n-m+1}.$$

When $B \leq e^{-1.7}$, for all $n \geq m \geq 2$, we have

$$n(n-m+1) e^{\frac{3}{2}n} e^{-\frac{B^2 n}{2}} B^n < 11.$$

Therefore, when $B \leq e^{-1.7}$, we have

$$P\left(\lambda_{\min} \leq \frac{Bn}{x^2}, \frac{\lambda_{\max}}{\lambda_{\min}} \leq x^2\right) < \frac{11B^{m-1}}{4\sqrt{4\pi}} \frac{1}{\Gamma(n-m+2)} \left(\frac{e^{-\frac{1}{2}} Bn}{x}\right)^{n-m+1}. \quad \square$$

Similar to real random matrices, we have the following lemma for complex random matrices. Lemma 5.4 can be proved using the same techniques as those for Lemma 5.3, so we will omit the proof and only give the result.

LEMMA 5.4. *For any $B^2 \leq e^{-1.2}$, $x > 0$, and $2 \leq m \leq n$, the largest eigenvalue $\tilde{\lambda}_{max}$ and the smallest eigenvalue $\tilde{\lambda}_{min}$ of $\tilde{W}_{m,n}$ satisfy*

$$P\left(\tilde{\lambda}_{min} \leq \frac{Bn}{x^2}, \frac{\tilde{\lambda}_{max}}{\tilde{\lambda}_{min}} \leq x^2\right) < 0.0352 \frac{1}{\Gamma(n-m+2)^2} \left(\frac{e^{-1}B^2n^2}{2x^2}\right)^{n-m+1}.$$

We are now prepared to derive the lower bounds for the tails of the condition number distributions of random matrices whose elements are independent and identically distributed standard normal random variables

THEOREM 5.5. *For any $x \geq n - m + 1$ and $n \geq m \geq 2$, the 2-norm condition number of $G_{m \times n}$ satisfies*

$$(5.1) \quad P\left(\frac{\kappa_2(G_{m \times n})}{n/(n-m+1)} > x\right) > \frac{1}{\sqrt{2\pi}} \left(\frac{c}{x}\right)^{n-m+1},$$

where $c \geq 0.245$ is a universal positive constant independent of x , m , and n .

Proof. For any positive constant H , we have

$$\begin{aligned} P(\kappa_2(G_{m \times n}) > x) &= P\left(\frac{\lambda_1}{\lambda_m} > x^2\right) \\ &> P\left(\lambda_m \leq \frac{H^2n}{x^2}, \frac{\lambda_1}{\lambda_m} > x^2\right) \\ &= P\left(\lambda_m \leq \frac{H^2n}{x^2}\right) - P\left(\lambda_m \leq \frac{H^2n}{x^2}, \frac{\lambda_1}{\lambda_m} \leq x^2\right). \end{aligned}$$

Let $H = e^{-1.7}$. Then based on Lemmas 5.1 and 5.3, we have

$$P(\kappa > x) > \left(\sqrt{\frac{2e^{\frac{5}{6}}}{3}} e^{-\frac{H^2mn}{2x^2}} - \frac{11H^{m-1}}{4\sqrt{4\pi}}\right) \frac{1}{\Gamma(n-m+2)} \left(\frac{e^{-\frac{1}{2}}Hn}{x}\right)^{n-m+1}.$$

From Lemma 2.7, we have

$$\Gamma(n-m+2) < \sqrt{2\pi(n-m+1)}(n-m+1)^{n-m+1} e^{-(n-m+1) + \frac{1}{12(n-m+1)}}.$$

Note that, for $2 \leq m \leq n$, we have

$$\sqrt{n-m+1} < 1.21^{n-m+1} \quad \text{and} \quad \frac{1}{12(n-m+1)} \leq \frac{1}{12}.$$

Therefore, we have

$$P(\kappa_2(G_{m,n}) > x) > \left(\sqrt{\frac{2e^{\frac{5}{6}}}{3}} e^{-\frac{H^2mn}{2x^2}} - \frac{11H^{m-1}}{4\sqrt{4\pi}}\right) \frac{e^{-\frac{1}{12}}}{\sqrt{2\pi}} \left(\frac{\frac{e}{1.21(n-m+1)} e^{-\frac{1}{2}}Hn}{x}\right)^{n-m+1}.$$

Since $H = e^{-1.7}$, $x \geq 1$, and $2 \leq m \leq n$, we have

$$\left(\sqrt{\frac{2e^{\frac{5}{6}}}{3}} e^{-\frac{H^2mn}{2x^2}} - \frac{11H^{m-1}}{4\sqrt{4\pi}}\right) e^{-\frac{1}{12}} > 0.99.$$

Therefore, we have

$$\begin{aligned} P(\kappa_2(G_{m,n}) > x) &> \frac{0.99}{\sqrt{2\pi}} \left(\frac{0.248 \frac{n}{n-m+1}}{x} \right)^{n-m+1} \\ &> \frac{1}{\sqrt{2\pi}} \left(\frac{0.245 \frac{n}{n-m+1}}{x} \right)^{n-m+1}. \end{aligned}$$

Therefore,

$$P\left(\frac{\kappa_2(G_{m,n})}{n/(n-m+1)} > x\right) > \frac{1}{\sqrt{2\pi}} \left(\frac{0.245}{x} \right)^{n-m+1}.$$

Let $c = 0.245$. Then we get (5.1). $\square$

Remark. 1. The lower bound in Theorem 5.5 is for arbitrary $n \geq m \geq 2$ and $x \geq n - m + 1$. For some special case of m and n , a more precise lower bound can be obtained. For example, for the special case of real random $m \times m$ matrices, where $m \geq 3$, it has been proved in [2] that

$$P(\kappa_2(G_{m \times m}) > m \cdot x) > \frac{c}{x},$$

where $c \geq 0.13$ is a universal positive constant independent of x and m .

In Theorem 5.5, however, if we take $m = n$, then we can only get

$$P(\kappa_2(G_{m \times m}) > m \cdot x) > \frac{0.097}{x}.$$

2. For the special case of real random $2 \times n$ matrices, based on the exact probability density function of $\kappa_2(G_{2 \times n})$ in [5], we can get

$$P(\kappa_2(G_{2 \times n}) > x) = \left(\frac{2x}{x^2 + 1} \right)^{n-1} \sim \left(\frac{2}{x} \right)^{n-1}$$

as $x \rightarrow \infty$. Hence, the constant c in Theorem 5.5 is no larger than 2. Therefore, the constant c in Theorem 5.5 actually satisfies

$$(5.2) \quad 0.245 \leq c \leq 2.$$

Similar to real random matrices, we have the following theorem for complex random matrices. Theorem 5.6 can be proved using the same techniques as those for Theorem 5.5, so we will omit the proof and only give the result.

THEOREM 5.6. *For any $x \geq n - m + 1$ and $n \geq m \geq 2$, the 2-norm condition number of $G_{m \times n}$ satisfies*

$$P\left(\frac{\kappa_2(\tilde{G}_{m \times n})}{n/(n-m+1)} > x\right) > \frac{1}{2\pi} \left(\frac{c}{x} \right)^{2(n-m+1)},$$

where $c \geq 0.319$ is a universal positive constant independent of x , m , and n .

6. The upper bounds for the expected logarithms. For square Gaussian random matrix $G_{n \times n}$, in [10], Smale asked for $E(\log \kappa_2(G_{n \times n}))$. Similarly, for rectangular Gaussian random matrix $G_{m \times n}$, it is also interesting to investigate $E(\log \kappa_2(G_{m \times n}))$. In this section, we will derive upper bounds for $E(\log \kappa_2(G_{m \times n}))$ and $E(\log \tilde{\kappa}_2(G_{m \times n}))$. Our main results are Theorems 6.1 and 6.2.

THEOREM 6.1. *For any $n \geq m \geq 2$, the 2-norm condition number of $G_{m \times n}$ satisfies*

$$(6.1) \quad E(\log \kappa_2(G_{m \times n})) < \log \frac{n}{n-m+1} + 2.258.$$

Proof. Let $f_\kappa(x)$ be the probability density function of $\kappa_2(G_{m \times n})$. Then

$$\begin{aligned} E \log \left(\frac{\kappa_2(G_{m \times n})}{6.414 \frac{n}{n-m+1}} \right) &= \int_1^\infty \log \left(\frac{x}{6.414 \frac{n}{n-m+1}} \right) f_\kappa(x) dx \\ &< \int_{6.414 \frac{n}{n-m+1}}^\infty \log \left(\frac{x}{6.414 \frac{n}{n-m+1}} \right) f_\kappa(x) dx \\ &= \int_{6.414 \frac{n}{n-m+1}}^\infty P(\kappa_2(G_{m \times n}) > x) \frac{1}{x} dx. \end{aligned}$$

From Theorem 4.3, we have

$$P(\kappa_2(G_{m \times n}) > x) < \frac{1}{\sqrt{2\pi}} \left(\frac{6.414 \frac{n}{n-m+1}}{x} \right)^{n-m+1}.$$

Therefore, we have

$$\begin{aligned} E \log \left(\frac{\kappa_2(G_{m \times n})}{6.414 \frac{n}{n-m+1}} \right) &< \frac{1}{\sqrt{2\pi}} \int_{6.414 \frac{n}{n-m+1}}^\infty \left(\frac{6.414 \frac{n}{n-m+1}}{x} \right)^{n-m+1} \frac{1}{x} dx \\ &= \frac{1}{(n-m+1)\sqrt{2\pi}} \\ &< 0.399. \end{aligned}$$

Therefore, we have

$$\begin{aligned} E \log(\kappa_2(G_{m \times n})) &< \log \frac{n}{n-m+1} + \log 6.414 + 0.399 \\ &< \log \frac{n}{n-m+1} + 2.258. \quad \square \end{aligned}$$

Remark. 1. For the special case of real random $m \times m$ matrices, from the results in [11], we can get

$$(6.2) \quad E \log(\kappa_2(G_{m \times m})) \leq \log m + \frac{3+3\log 2}{2} \approx 2.54.$$

In Theorem 6.1, if we take $m = n$, then we have

$$E \log(\kappa_2(G_{m \times n})) < \log n + 2.258,$$

which is a slightly improved version of (6.2).

2. The upper bound in Theorem 6.1 is for arbitrary $n \geq m \geq 2$. For some special case of m and n or large m and n , more precise results exist.

For the special case of real random $2 \times n$ matrices, it was shown in [6] that

$$E \log(\kappa_2(G_{2 \times n})) = \frac{1}{2} \sqrt{\pi} \frac{\Gamma\left(\frac{n-1}{2}\right)}{\Gamma\left(\frac{n}{2}\right)}.$$

For real random $m \times m$ matrices, it has been proved in [6] that

$$E \log(\kappa_2(G_{m \times m})) = \log m + c + o(1)$$

as $m \rightarrow \infty$, where $c \approx 1.537$.

For rectangular matrix $G_{m_n \times n}$, if $\lim_{n \rightarrow \infty} m_n/n = y$ and $0 < y < 1$, then it has been proved in [6] that

$$E \log(\kappa_2(G_{m_n \times n})) = \log \frac{1 + \sqrt{y}}{1 - \sqrt{y}} + o(1)$$

as $n \rightarrow \infty$.

Similar to real random matrices, we have the following theorem for complex random matrices. Theorem 6.2 can be proved using the same techniques as those for Theorem 6.1, so we will omit the proof and only give the result.

THEOREM 6.2. *For any $n \geq m \geq 2$, the 2-norm condition number of $G_{m \times n}$ satisfies*

$$E(\log \kappa_2(\tilde{G}_{m \times n})) < \log \frac{n}{n - m + 1} + 2.240.$$

Acknowledgment. We would like to thank Julien Langou for precious discussions. We are very grateful to referees for their helpful corrections, valuable comments, and constructive suggestions which have improved the presentation of this paper.

REFERENCES

- [1] M. ABRAMOWITZ AND I. A. STEGUN, EDS., *Handbook of Mathematical Functions*, Dover, New York, 1970.
- [2] J. M. AZAIS AND M. WSCEBOR, *Upper and lower bounds for the tails of the distribution of the condition number of a Gaussian matrix*, SIAM J. Matrix Anal. Appl., 26 (2005), pp. 426–440.
- [3] Z. CHEN AND J. DONGARRA, *Numerically Stable Real Number Codes Based on Random Matrices*, Technical report ut-cs-04-526, Department of Computer Science, The University of Tennessee, Knoxville, 2004.
- [4] Z. CHEN, G. E. FAGG, E. GABRIEL, J. LANGOU, T. ANGSKUN, G. BOSILCA, AND J. DONGARRA, *Fault tolerant high performance computing by a coding approach*, in Proceedings of ACM SIGPLAN Symposium on Principles and Practice of Parallel Programming, Chicago, IL, 2005.
- [5] A. EDELMAN, *Eigenvalues and Condition Numbers of Random Matrices*, Ph.D. thesis, Department of Mathematics, MIT, Cambridge, MA, 1989.
- [6] A. EDELMAN, *Eigenvalues and condition numbers of random matrices*, SIAM J. Matrix Anal. Appl., 9 (1988), pp. 543–560.
- [7] A. EDELMAN AND B. D. SUTTON, *Tails of condition number distributions*, SIAM J. Matrix Anal. Appl., 27 (2005), pp. 547–560.
- [8] R. L. GRAHAM, D. E. KNUTH, AND O. PATASHNIK, *Concrete Mathematics: A Foundation for Computer Science*, 2nd ed., Addison-Wesley, Reading, MA, 1994.
- [9] A. T. JAMES, *Distributions of matrix variates and latent roots derived from normal samples*, Ann. Math. Statist., 35 (1964), pp. 475–501.
- [10] S. SMALE, *On the efficiency of algorithms of analysis*, Bull. Amer. Math. Soc., 13 (1985), pp. 87–121.
- [11] S. J. SZAREK, *Condition numbers of random matrices*, J. Complexity, 7 (1991), pp. 131–149.

GENERALIZING CRAMER'S RULE: SOLVING UNIFORMLY LINEAR SYSTEMS OF EQUATIONS*

GEMA M. DIAZ-TOCA[†], LAUREANO GONZALEZ-VEGA[‡], AND HENRI LOMBARDI[§]

Abstract. Following Mulmuley's lemma, this paper presents a generalization of the Moore–Penrose inverse for a matrix over an arbitrary field. This generalization yields a way to uniformly solve linear systems of equations which depend on some parameters.

Key words. Mulmuley's method, generalized inverses, Gram coefficients, Moore–Penrose inverse

AMS subject classifications. 15A09, 65F05, 65F20

DOI. 10.1137/S0895479802418860

Introduction. Given a system of linear equations over an arbitrary field $\mathbb{K}$ with the coefficient matrix $A \in \mathbb{K}^{m \times n}$, $Ax = v$, the existence of solution depends only on the ranks of the matrices A and $A|v$. These ranks can be easily obtained by using the Gaussian elimination, but this method is not uniform. When the entries of A and v depend on parameters, the resolution of the considered linear system may involve discussions on too many cases (see, for example, [18]).

Solving parametric linear systems of equations is an interesting computational issue encountered, for example, when computing the characteristic solutions for differential equations (see [5]), when dealing with colored Petri nets (see [13]) or in linear prediction problems, and in different applications of operation research and engineering (see, for example, [6], [8], [12], [15], or [19] where the elements of the matrix have a linear affine dependence on a set of parameters). More general situations where the elements are polynomials in a given set of parameters are encountered in computer algebra problems (see [1] and [18]).

In this paper, we give a generalization of Mulmuley's algorithm (see [16]). If $A \in \mathbb{K}^{m \times n}$, we introduce the following matrix $A^\circ \in \mathbb{K}(t)^{n \times m}$:

$$A^\circ = \begin{pmatrix} 1 & 0 & \cdots & 0 \\ 0 & t^{-1} & \cdots & 0 \\ \vdots & \vdots & & \vdots \\ 0 & 0 & \cdots & t^{-(n-1)} \end{pmatrix} A^t \begin{pmatrix} 1 & 0 & \cdots & 0 \\ 0 & t & \cdots & 0 \\ \vdots & \vdots & & \vdots \\ 0 & 0 & \cdots & t^{m-1} \end{pmatrix},$$

where t is a new variable. Next we define the generalized Gram polynomials of A as the coefficients of the characteristic polynomial of AA° , which are Laurent polynomials in the variable t . From the generalized Gram polynomials it is possible in a very

*Received by the editors December 2, 2002; accepted for publication (in revised form) by D. A. Bini May 3, 2005; published electronically November 23, 2005. This research was partially supported by the European Union funded project RAAG (HPRN-CT-2001-00271) and by the Spanish grant BFM2002-04402-C02-0.

<http://www.siam.org/journals/simax/27-3/41886.html>

[†]Departamento de Matemática Aplicada, Universidad de Murcia, Murcia, Spain (gemadiaz@um.es).

[‡]Departamento de Matemáticas, Estadística y Computación, Universidad de Cantabria, Santander 39005, Spain (laureano.gonzalez@unican.es).

[§]Équipe de Mathématiques, UMR CNRS 6623, Université de Franche-Comté, Besançon, France (henri.lombardi@univ-fcomte.fr).

uniform and compact way

1. to characterize the rank of A , and
2. once the rank of A is fixed, to determine
 - (a) matrices associated with precise projections of $\mathbb{K}(t)^m$ onto $\text{Im } A$ and of $\mathbb{K}(t)^n$ onto $\text{Ker } A$, respectively;
 - (b) the equations of $\text{Im}(A)$;
 - (c) a generalized Moore–Penrose inverse of A ;
 - (d) given $v \in \text{Im}(A)$, a solution of $Ax = v$ generalizing the classical Cramer’s rule to the underdetermined and overdetermined cases via our generalized Moore–Penrose inverse.

This provides a generalization of the classical Moore–Penrose inverse and Gram coefficients in the real or complex case. The surprising fact is that for nonreal fields (e.g., fields of characteristic p) the rank of any matrix and a uniform generalized inverse of the matrix can also be controlled by a small number of sums of squares of minors.

In fact our generalized Gram polynomials are only slight variants of the polynomials that appear in Mulmuley’s algorithm.

A consequence of this method is, as in [16], that the characterization of a full rank matrix in $\mathbb{K}^{m \times n}$ (for any field $\mathbb{K}$) can be done by using less than $\min(n, m)|n - m| + 1$ polynomials in the entries of A : each one of these polynomials is a sum of squares of maximal minors of A . If $n = 2m$, then $\min(n, m)|n - m| + 1 = m^2 + 1$, which is clearly far from the consideration of the $\binom{n}{m}$ maximal minors of A .

In general, the total number of sums of squares of minors to be used for characterizing the rank is smaller than $\frac{1}{2}p(p+1)p'$, where $p = \min(m, n)$ and $p' = \max(m, n)$, so it is bounded by $\mathcal{O}(p'p^2)$; note that the total number of minors is clearly bigger than $2^p \left(\frac{p'}{p}\right)^{\frac{p}{2}}$.

Compared to Mulmuley’s method, an important (and new) characteristic of the proposed method is the generalization of Moore–Penrose inverses for each possible rank r . This allows us to obtain a uniform solution to linear systems of equations once the rank is known. Moreover, this is particularly useful when linear systems of equations depend on some parameters (thus avoiding any discussion of when the pivots to be used vanish, or fail to; see the last section for enlightening examples).

Our method requires a complexity similar to the one given in [11], where the parameterized Gauss elimination is analyzed, and it is proved that with k parameters, the complexity of parametric solution to $Ax = v$ is in $n^{O(k)}$.

This paper is organized as follows. The first section recalls the classical theory about generalized inverse matrices. In the second section, we introduce a generalization of both Gram coefficients and Moore–Penrose inverse matrices for an arbitrary field. Finally, the third section presents three different examples, where we apply the different results described in the second section.

1. Gram coefficients and Moore–Penrose inverse in the real or complex case. This section introduces some important theoretical points of Gram coefficients and the Moore–Penrose inverse in the real or complex case.

In [2], [3], [10], [14], [17], one can find different approaches to studying generalized inverses. In [7], one can find precise proofs of the results appearing in this section.

1.1. General theory. Let $\mathbb{K}$ be either the real or the complex field, $E = \mathbb{K}^n$, $F = \mathbb{K}^m$, and let $\mathcal{L}(E, F)$ denote the set of all linear transformations of E into F . If H is a linear subspace of E , we denote by $\pi_H : E \rightarrow E$ the orthogonal projection on H .

Let $\varphi \in \mathcal{L}(E, F)$ and let $A \in \mathbb{K}^{m \times n}$ be a matrix representation of φ with respect to orthonormal bases. If $\langle x, y \rangle$ denotes the inner product of x and y , the conjugate transpose A^* of A represents the adjoint transformation φ^* such that

$$(1.1) \quad \forall x \in E \quad \forall y \in F, \quad \langle \varphi(x), y \rangle_F = \langle x, \varphi^*(y) \rangle_E.$$

The results recalled here are based on the following lemma.

LEMMA 1.1. *If $A \in \mathbb{K}^{m \times n}$, then*

$$(1.2) \quad \text{Im } A \oplus \text{Ker } A^* = F, \quad \text{Ker } A \oplus \text{Im } A^* = E.$$

The matrices AA^* and A^*A are positive semidefinite matrices, and in general they may be singular.

DEFINITION 1.2. *The Gram coefficients of A are defined to be the coefficients $\mathcal{G}_k(A) = a_k$ of the polynomial*

$$(1.3) \quad \det(I_m + zAA^*) = 1 + a_1z + \cdots + a_mz^m,$$

where z is a variable. We also define $\mathcal{G}_0(A) = 1$ and $\mathcal{G}_\ell(A) = 0$ if $\ell > m$.

If $Q(z) = \det(I_m + zAA^*)$, then $\det(zI_m - AA^*) = (-1)^m z^m Q(1/z)$. Hence the Gram coefficients of A are, except for the signs, the coefficients of the characteristic polynomial of AA^* .

LEMMA 1.3 (the Gram conditions for rank).

1. $\text{rk}(A) \leq r \iff \mathcal{G}_{r+1}(A) = 0 \iff \bigwedge_{k>r} \mathcal{G}_k(A) = 0$.
2. $\text{rk}(A) = r \iff \mathcal{G}_{r+1}(A) = 0 \neq \mathcal{G}_r(A)$.

In the following we assume $r = \text{rk}(A)$. So $\det(I_m + zAA^*) = \det(I_n + zA^*A) = 1 + a_1z + \cdots + a_rz^r$. The use of the Cayley–Hamilton theorem yields the following lemma.

LEMMA 1.4 (orthogonal projections onto the image and onto the kernel).

1. *The orthogonal projection π_I onto the subspace $I = \text{Im } A \subseteq F$ is equal to*

$$(1.4) \quad \pi_I = a_r^{-1}(a_{r-1}AA^* - a_{r-2}(AA^*)^2 + \cdots + (-1)^{r-1}(AA^*)^r).$$

2. *The orthogonal projection π_{I^*} onto the subspace $I^* = \text{Im } A^* \subseteq E$ is equal to*

$$(1.5) \quad \pi_{I^*} = a_r^{-1}(a_{r-1}A^*A - a_{r-2}(A^*A)^2 + \cdots + (-1)^{r-1}(A^*A)^r).$$

And the orthogonal projection onto the kernel of A is $I_n - \pi_{I^*}$.

Let φ_0 be the linear isomorphism from $\text{Im } A^*$ to $\text{Im } A$, defined by the restriction of φ .

DEFINITION 1.5. *Suppose $\text{rk}(A) = r$. The Moore–Penrose inverse of A in rank r is the linear transformation $\varphi^{\dagger r} : F \rightarrow E$ defined by*

$$\forall y \in F, \quad \varphi^{\dagger r}(y) = \varphi_0^{-1}(\pi_{\text{Im } A}(y)).$$

Let $A^{\dagger r}$ be the matrix associated with $\varphi^{\dagger r}$.

PROPOSITION 1.6 (the Moore–Penrose inverse). *Assume $\text{rk}(A) = r$. Let $v \in F$. Then the following hold:*

1. $A^{\dagger r} \in \mathbb{K}^{n \times m}$ is given by

$$(1.6) \quad \begin{aligned} A^{\dagger r} &= a_r^{-1}(a_{r-1}I_n - a_{r-2}(A^*A) + \cdots + (-1)^{r-1}(A^*A)^{r-1})A^* \\ &= a_r^{-1}A^*(a_{r-1}I_m - a_{r-2}(AA^*) + \cdots + (-1)^{r-1}(AA^*)^{r-1}). \end{aligned}$$

Moreover, $A = AA^{\dagger r}A$, $A^{\dagger r} = A^{\dagger r}AA^{\dagger r}$, $\pi_I = AA^{\dagger r}$, and $\pi_{I^*} = A^{\dagger r}A$.

2. The linear system $Ax = v$ has a solution if and only if $\mathcal{G}_{r+1}(A|v) = 0$.
3. If $v \in \text{Im } A$, then

$$(1.7) \quad v = AA^{\dagger_r}v$$

and $x = A^{\dagger_r}v$ is a solution of the linear system $Ax = v$. Moreover, it is the only solution in the linear space generated by the columns of A^* ; that is, its norm is minimal.

Note that (1.6) can be viewed as a definition of the matrix $A^{\dagger_r} \in \mathbb{K}^{n \times m}$, requiring only that $\text{rk}(A) \geq r$, that is, $a_r \neq 0$. This fact is useful in numerical analysis when the coefficients of A are real numbers known with finite precision that implies some doubts about the rank of the matrix.

1.2. Cramer identities. In this subsection, we give a description of the Moore–Penrose inverse as a weighted sum of Cramer identities.

LEMMA 1.7. *The Gram coefficient $\mathcal{G}_k(A)$ is equal to the sum of squares of modulus of k -minors of the matrix A .*

Observe that this lemma provides another way to understand Lemma 1.3. In the following, we consider only the real case. Obvious modifications have to be made in the complex case. Suppose $\text{rk}(A) = r$ and $v \in \text{Im } A$.

Notation 1.1.

- A_j denotes the j th column of A .
- $A_{\alpha,\beta}$ denotes the submatrix of A where the rows and columns retained are given by subscripts $\alpha = \{\alpha_1, \dots, \alpha_r\} \subset \{1, \dots, m\}$ and $\beta = \{\beta_1, \dots, \beta_r\} \subset \{1, \dots, n\}$, respectively.
- $A_{1:m,\beta}$ denotes the submatrix of A retaining the columns given by β . $A_{\alpha,1:n}$ denotes the submatrix of A retaining the rows given by α .
- $\mu_{\alpha,\beta}$ denotes the r -minor determinant of $A_{\alpha,\beta}$.
- Given $j \in \{1, \dots, r\}$, let $\sigma_{\alpha,\beta,j}$ denote the determinant of $A_{\alpha,\beta}$ but replacing the j th column with the column obtained from v retaining the rows α .
- Let $\text{Adj}_{\alpha,\beta}(A) = (I_n)_{1:n,\beta} \text{Adj}(A_{\alpha,\beta}) (I_m)_{\alpha,1:m}$.

Then for every subscript pair (α, β) , since $\text{rk}(A|v) = r$, we obtain a Cramer identity,

$$\begin{aligned} \mu_{\alpha,\beta} v &= \sum_{j=1}^r \sigma_{\alpha,\beta,j} A_{\beta_j} = (A_{\beta_1} \cdots A_{\beta_r}) \begin{pmatrix} \sigma_{\alpha,\beta,1} \\ \vdots \\ \sigma_{\alpha,\beta,r} \end{pmatrix} \\ &= (A_{\beta_1} \cdots A_{\beta_r}) \text{Adj}(A_{\alpha,\beta}) \begin{pmatrix} v_{\alpha_1} \\ \vdots \\ v_{\alpha_r} \end{pmatrix} \\ &= A (I_n)_{1:n,\beta} \text{Adj}(A_{\alpha,\beta}) (I_m)_{\alpha,1:m} v = A \text{Adj}_{\alpha,\beta}(A) v, \end{aligned}$$

that is,

$$(1.8) \quad \mu_{\alpha,\beta} v = A \text{Adj}_{\alpha,\beta}(A) v.$$

Since $\mathcal{G}_r(A) = \sum_{\alpha,\beta} \mu_{\alpha,\beta}^2 \neq 0$, where the summation extends over all $\binom{n}{r} \binom{m}{r}$ distinct sets of row and column indices, by multiplying every equality (1.8) by $\mu_{\alpha,\beta}$ and by adding up all these equalities, we obtain the following expression:

$$(1.9) \quad \mathcal{G}_r(A) v = A \left(\sum_{\alpha,\beta} \mu_{\alpha,\beta} \text{Adj}_{\alpha,\beta}(A) \right) v,$$

which is highly similar to (1.7) in Proposition 1.6. Indeed, [17, eqn. (2.13)] applied to our case yields

$$(1.10) \quad A^{\dagger r} = \mathcal{G}_r(A)^{-1} \sum_{\beta, \alpha} \mu_{\alpha, \beta} \text{Adj}_{\alpha, \beta}(A),$$

which is interpreted as a weighted sum of Cramer identities.

2. Generalization for an arbitrary field. This section is devoted to presenting a generalization of Gram coefficients and Moore–Penrose inverse matrices for an arbitrary field $\mathbb{K}$. Our work can be seen as an extension of the result of Mulmuley.

LEMMA 2.1 (Mulmuley’s algorithm). *Let $P \in \mathbb{K}^{n \times n}$ be a symmetric matrix, let t be a variable, and let $Q_n = \text{diag}(1, t, t^2, \dots, t^{n-1})$. If $\tilde{P} = PQ_n$ and $c_k(t)$ is the coefficient of z^{n-k} in the characteristic polynomial of $\tilde{P}$, then the rank of the matrix P is smaller than or equal to r if and only if $c_k(t) = 0$ for $k > r$.*

The classical results introduced in section 1.1 are based on orthogonal sums of images and kernels of $A \in \mathbb{K}^{m \times n}$ and $A^* \in \mathbb{K}^{n \times m}$ (1.2).

So, if we consider vector spaces over an arbitrary field $\mathbb{K}$, all the results would similarly work if we were able to define a good inner product and some suitable $\varphi^\circ : F \rightarrow E$ satisfying (1.2) and if we replace the expression *orthogonal projection onto $\text{Im } \varphi$* by *projection onto $\text{Im } \varphi$ parallel to $\text{Ker}(\varphi^\circ)$* .

2.1. General theory. With the purpose of generalizing (1.2) for an arbitrary field $\mathbb{K}$, we have to introduce a parameter t and work over the field $\mathbb{K}(t)$. Next we define the quadratic form $\Phi_{t,n}$ on $E' = \mathbb{K}(t)^n$ with values in $\mathbb{K}(t)$ as

$$\Phi_{t,n}(\xi_1, \dots, \xi_n) = \xi_1^2 + t\xi_2^2 + \dots + t^{n-1}\xi_n^2,$$

and the quadratic form $\Phi_{t,m}$ on $F' = \mathbb{K}(t)^m$ with values in $\mathbb{K}(t)$ as

$$\Phi_{t,m}(\zeta_1, \dots, \zeta_m) = \zeta_1^2 + t\zeta_2^2 + \dots + t^{m-1}\zeta_m^2.$$

The “associated inner products” with $\Phi_{t,n}$ and $\Phi_{t,m}$, denoted by $\langle \cdot, \cdot \rangle_{E'}$ and $\langle \cdot, \cdot \rangle_{F'}$, respectively, lead us naturally to the desired linear transformation φ° .

Given a $\mathbb{K}$ -linear transformation $\varphi \in \mathcal{L}(E, F)$, we get by extension of scalars a $\mathbb{K}(t)$ -linear transformation φ' in $\mathcal{L}(E', F')$. The matrix A of φ is the same as the matrix of φ' .

Thus, there exists only one linear transformation, $A^\circ : F' \rightarrow E'$, which verifies

$$(2.1) \quad \forall x \in E' \quad \forall y \in F', \quad \langle Ax, y \rangle_{F'} = \langle x, A^\circ y \rangle_{E'},$$

which is very similar to (1.1).

If $Q_n = \text{diag}(1, t, t^2, \dots, t^{n-1})$ and $Q_m = \text{diag}(1, t, t^2, \dots, t^{m-1})$ are the diagonal matrices associated with $\langle \cdot, \cdot \rangle_{E'}$ and $\langle \cdot, \cdot \rangle_{F'}$, respectively, A° is given by

$$A^\circ = Q_n^{-1} A^t Q_m.$$

Hence, for all $x \in \mathbb{K}(t)^{n \times 1}$, $y \in \mathbb{K}(t)^{m \times 1}$, we have $(Ax)^t Q_m y = x^t Q_n (A^\circ y)$.

In practice, if $A = (a_{i,j})$, then $A^\circ = (t^{j-i} a_{j,i})$, for instance,

$$A = \begin{bmatrix} a_{11} & a_{12} & a_{13} & a_{14} & a_{15} \\ a_{21} & a_{22} & a_{23} & a_{24} & a_{25} \\ a_{31} & a_{32} & a_{33} & a_{34} & a_{35} \end{bmatrix}, \quad A^\circ = \begin{bmatrix} a_{11} & t a_{21} & t^2 a_{31} \\ t^{-1} a_{12} & a_{22} & t a_{32} \\ t^{-2} a_{13} & t^{-1} a_{23} & a_{33} \\ t^{-3} a_{14} & t^{-2} a_{24} & t^{-1} a_{34} \\ t^{-4} a_{15} & t^{-3} a_{25} & t^{-2} a_{35} \end{bmatrix}.$$

Furthermore, $(A^\circ)^\circ = A$ and $(AB)^\circ = B^\circ A^\circ$. Next we state and prove the following key lemma.

LEMMA 2.2. *With the previous notation*

$$(2.2) \quad \text{Im } A \oplus \text{Ker } A^\circ = F', \quad \text{Ker } A \oplus \text{Im } A^\circ = E'.$$

Thus,

$$(2.3) \quad \begin{aligned} \text{Im } A &= \text{Im } AA^\circ, & \text{Ker } A^\circ &= \text{Ker } AA^\circ, \\ \text{Ker } A &= \text{Ker } A^\circ A, & \text{Im } A^\circ &= \text{Im } A^\circ A. \end{aligned}$$

Proof. Clearly A and A° have the same rank, so we only need to prove

$$\text{Im } A \cap \text{Ker } A^\circ = \{0\} \quad \text{and} \quad \text{Im } A^\circ \cap \text{Ker } A = \{0\}.$$

Equation (2.1) implies that

- the orthogonal subspace of $\text{Im } A$ (with respect to the bilinear form $\langle \cdot, \cdot \rangle_{F'}$) is equal to $\text{Ker } A^\circ$, and
- the orthogonal subspace of $\text{Im } A^\circ$ (with respect to the bilinear form $\langle \cdot, \cdot \rangle_{E'}$) is equal to $\text{Ker } A$ but, since $\langle \cdot, \cdot \rangle_{E'}$ is a nondegenerate bilinear form, the orthogonal subspace of $\text{Ker } A$ (with respect to the bilinear form $\langle \cdot, \cdot \rangle_{E'}$) is equal to $\text{Im } A^\circ$.

Moreover, $\text{Im } A$ and $\text{Ker } A$ are defined over $\mathbb{K}$ inside F and E . So, it is enough to prove the following statement:

Given a $\mathbb{K}$ -subspace H_1 of F (resp., of E), if H is the $\mathbb{K}(t)$ -subspace of F' (resp., of E') generated by H_1 , then $H \cap H^\perp = \{0\}$.

Suppose that $(p_1(t), \dots, p_m(t)) \in H \cap H^\perp$ and $(p_1(t), \dots, p_m(t)) \in \mathbb{K}[t]^m$. Let $v_1, \dots, v_s \in \mathbb{K}^m$ be a basis in H_1 and $a_1(t), \dots, a_s(t) \in \mathbb{K}(t)$ such that

$$(p_1(t), \dots, p_m(t)) = \sum_i a_i(t) v_i.$$

Let u be a new unknown and let us work in $\mathbb{K}[t, u]$. Since $\sum_i a_i(t) v_i \in H^\perp$, it follows that

$$P(t, u) = \left\langle \sum_i a_i(t) v_i, \sum_i a_i(u) v_i \right\rangle_{F'} = \sum_{i=1}^m p_i(t) p_i(u) t^{i-1} = 0.$$

Now we deduce from the last equality that $p_i(t) = 0$ for every i . We present a proof by induction on the highest degree of p_i 's.

If $\deg(p_i(t)) = 0$ for all i , then the statement is easily verified. In the general case, we first see that the independent coefficients of the p_i 's are zero. The equality $P(0, 0) = p_1(0)^2$ implies that $p_1(0) = 0$. And if $p_1(0) = \dots = p_k(0) = 0$, then the coefficient of t^k in $P(t, u)$ is equal to $p_{k+1}(0)^2$ and therefore $p_{k+1}(0) = 0$. We conclude that all the independent coefficients are zero and every $p_i(t)$ is divisible by t . So the result follows by induction. $\square$

Note that this proof is based on the one for Mulmuley's lemma in [16].

Lemma 2.2 is the analogue of Lemma 1.1. This allows us to generalize Definitions 1.2 and 1.5, Lemmas 1.3 and 1.4, and Proposition 1.6 in the following way.

AA° are in the Laurent polynomial ring $\mathbb{K}[t, 1/t]$.

DEFINITION 2.3. *The generalized Gram polynomials, $\mathcal{G}'_k(A)(t) = a_k(t) \in \mathbb{K}[t, 1/t]$, and the generalized Gram coefficients, $\mathcal{G}'_{k,\ell}(A) = a_{k,\ell} \in \mathbb{K}$, are given by the following expression:*

$$(2.4) \quad \begin{cases} \det(\mathbf{I}_m + zAA^\circ) = 1 + a_1(t)z + \cdots + a_m(t)z^m, \\ a_k(t) = t^{-k(n-k)} \left(\sum_{\ell=0}^{k(m+n-2k)} a_{k,\ell} t^\ell \right). \end{cases}$$

Observe that if matrix A is real, the usual Gram coefficients are given by

$$(2.5) \quad \mathcal{G}_k(A) = \mathcal{G}'_k(A)(1) = \sum_{\ell} a_{k,\ell}.$$

If $p = \min(m, n)$ and $p' = \max(m, n)$, then the total number of generalized Gram coefficients is equal to

$$\sum_{k=1}^p (k(m+n-2k) + 1) = p + \frac{1}{6}p(p+1)(3p' - p - 2) \leq \frac{1}{2}p(p+1)p'.$$

Generalized Gram coefficients provide the rank of the given matrix, generalizing Lemma 1.3.

LEMMA 2.4 (generalized Gram conditions for the rank).

1. $\text{rk}(A) \leq r \iff \bigwedge_{k>r} \mathcal{G}'_k(A)(t) = 0$.
2. $\text{rk}(A) = r \iff \bigwedge_{k>r} \mathcal{G}'_k(A)(t) = 0 \wedge \mathcal{G}'_r(A)(t) \neq 0$.

This can be seen as a reformulation of the key result in [16]. Similarly, we can generalize Lemma 1.4.

LEMMA 2.5 (orthogonal projections onto the image and onto the kernel). *Let $a_k(t) = \mathcal{G}'_k(A)$. If $\text{rk}(A) = r$, then*

1. *the projection onto the subspace $\text{Im } A \subseteq F'$ parallel to $\text{Ker } A^\circ$ is the orthogonal projection with respect to $\langle, \rangle_{F'}$ and is given by*

$$(2.6) \quad \pi_{\text{Im } A} = a_r^{-1}(a_{r-1}AA^\circ - a_{r-2}(AA^\circ)^2 + \cdots + (-1)^{r-1}(AA^\circ)^r);$$

2. *the projection onto the subspace $\text{Im } A^\circ \subseteq E'$ parallel to $\text{Ker } A$ is the orthogonal projection with respect to $\langle, \rangle_{E'}$ and is given by*

$$(2.7) \quad \pi_{\text{Im } A^\circ} = a_r^{-1}(a_{r-1}A^\circ A - a_{r-2}(A^\circ A)^2 + \cdots + (-1)^{r-1}(A^\circ A)^r);$$

and the projection onto $\text{Ker } A$ parallel to $\text{Im } A^\circ$ is $\mathbf{I}_n - \pi_{\text{Im } A^\circ}$.

REMARK 2.1. *Actually the unknown t can be replaced in every formula with any value $\tau \in \mathbb{K}$ which does not cause the denominator $a_r(t)$ to vanish (always possible if the field $\mathbb{K}$ has more than $r(m+n-2r)+1$ elements).*

Let φ_0 be the linear isomorphism from $\text{Im } A^\circ \subseteq E'$ to $\text{Im } A \subseteq F'$, which is the restriction of φ' . Next we introduce the generalization of the Moore–Penrose inverse.

DEFINITION 2.6. Suppose $\text{rk}(A) = r$. The generalized Moore–Penrose inverse of A in rank r is the linear transformation $\varphi^{\dagger_{r,t}} : F' \rightarrow E'$ defined by

$$\forall y \in F', \quad \varphi^{\dagger_{r,t}}(y) = \varphi_0^{-1}(\pi_{\text{Im } A}(y)).$$

Let $A^{\dagger_{r,t}}$ be the matrix associated with $\varphi^{\dagger_{r,t}}$.

The next result shows how to obtain the generalized Moore–Penrose inverse of A for an arbitrary field, and it is the most important result presented in the paper.

THEOREM 2.7 (generalized Moore–Penrose inverse). Let $a_k(t) = \mathcal{G}'_k(A)$, $v \in F$, and $\text{rk}(A) = r$. Then

1. the generalized Moore–Penrose inverse of $A \in \mathbb{K}^{m \times n}$ in rank r is given by

$$(2.8) \quad A^{\dagger_{r,t}} = a_r^{-1}(a_{r-1}I_n - a_{r-2}A^\circ A + \cdots + (-1)^{r-1}(A^\circ A)^{r-1})A^\circ;$$

moreover, $AA^{\dagger_{r,t}}A = A$, $A^{\dagger_{r,t}}AA^{\dagger_{r,t}} = A^{\dagger_{r,t}}$, $AA^{\dagger_{r,t}} = \pi_{\text{Im } A}$, and $A^{\dagger_{r,t}}A = \pi_{\text{Im } A^\circ}$;

2. the linear system $Ax = v$ has a solution if and only if $\mathcal{G}'_{r+1}(A|v) = 0$;
3. if $v \in \text{Im } A$, then

$$(2.9) \quad v = AA^{\dagger_{r,t}}v$$

and $x = A^{\dagger_{r,t}}v$ is a solution of the linear system $Ax = v$; moreover, it is the only solution in the linear space generated by the columns of A° .

COROLLARY 2.8.

- (i) If A is injective (i.e., $r = n$) and $v \in \text{Im } A$, the vector $A^{\dagger_{r,t}}v$ is the only solution of the corresponding linear system. Therefore, it does not depend on t , and the coordinates of $A^{\dagger_{r,t}}v$, which are, a priori, rational functions on the variable t , are in fact in $\mathbb{K}$: these constants in $\mathbb{K}$ are obtained by performing the euclidean divisions between the numerators and the denominator which are, in this case, always exact.
- (ii) Given $A^{\dagger_{r,t}}$ by (2.8), if we define

$$B_r = a_r(t)A^{\dagger_{r,t}} = \sum B_{r,k}t^k, \quad B_{r,k} \in \mathbb{K}^{n \times m},$$

then $AB_{r,k}A = a_{r,k}A$.

Moreover, for every $a_{r,k} \neq 0$, we can define

$$C_{r,k} = 1/a_{r,k} B_{r,k}, \quad D_{r,k} = C_{r,k}AC_{r,k},$$

and then it is easily checked that the matrix $D_{r,k}$ satisfies the equations

$$AD_{r,k}A = A, \quad D_{r,k}AD_{r,k} = D_{r,k}.$$

Thus, we can conclude that $D_{r,k} \in \mathbb{K}^{n \times m}$ is a generalized inverse of A and so

- (a) if $v \in \text{Im}(A)$, then $D_{r,k}v$ is a solution of $Ax = v$,
- (b) $AD_{r,k}$ is a projection onto $\text{Im } A$, and
- (c) $I - D_{r,k}A$ is a projection onto $\text{Ker } A$.

For more details see [7].

2.2. Generalizing Cramer identities. Next, we introduce a description of the generalized Gram coefficients of A as sums of squares of minors.

LEMMA 2.9. *Let $A \in \mathbb{K}^{m \times n} \subseteq \mathbb{K}(t)^{m \times n}$ and $\mu_{\alpha, \beta}$ be the k -minor where the rows and columns retained are given by subscripts $\alpha = \{\alpha_1, \dots, \alpha_k\} \subset \{1, \dots, m\}$ and $\beta = \{\beta_1, \dots, \beta_k\} \subset \{1, \dots, n\}$. If $S_{m, n, k, \ell}$ is defined as*

$$S_{m, n, k, \ell} = \{(\alpha, \beta) \mid |\alpha| - |\beta| = \ell - k(n - k)\},$$

then the generalized Gram coefficient $a_{k, \ell} = \mathcal{G}'_{k, \ell}(A)$ is given by

$$(2.10) \quad \mathcal{G}'_{k, \ell}(A) = \sum_{(\alpha, \beta) \in S_{m, n, k, \ell}} \mu_{\alpha, \beta}^2.$$

By the same reasoning as in the previous section, we obtain (1.8). Thus, if for every (α, β) in $S_{m, n, k, \ell}$, we multiply both sides of (1.8) by $\mu_{\alpha, \beta} t^{|\alpha| - |\beta|}$ and add up all these equalities, we obtain the expression

$$(2.11) \quad \mathcal{G}'_r(A) v = A \left(\sum_{\alpha, \beta} \mu_{\alpha, \beta} t^{|\alpha| - |\beta|} \text{Adj}_{\alpha, \beta}(A) \right) v,$$

which is very close to (2.9) in Theorem 2.7. In fact, by applying [17, eqn. (2.13)], we get the following proposition.

PROPOSITION 2.10. *Let $A \in \mathbb{K}^{m \times n}$; then*

$$(2.12) \quad A^{\dagger_{r, t}} = \mathcal{G}'_r(A)^{-1} \sum_{\alpha, \beta} \mu_{\alpha, \beta} t^{|\alpha| - |\beta|} \text{Adj}_{\alpha, \beta}(A).$$

As in the usual case, (2.12) shows that the generalized Moore–Penrose inverse can be interpreted as a weighted sum of Cramer identities. Observe that if we evaluate (2.12) at $t = 1$, then we find out the theory of the Moore–Penrose inverse in the real case.

2.3. Case of symmetric matrices. If $A = A^t$, there is a more simplified expression for its generalized Moore–Penrose inverse than (2.8). So, assume that $E = F$, $\text{rk}(A) = r$, and $A = A^t$. Since A is symmetric, $A^\circ = Q_n^{-1} A Q_n$. Let $\tilde{A} = A Q_n$. Then,

$$\begin{aligned} \text{Im } \tilde{A} &= \text{Im } A, & \text{Ker } \tilde{A} &= \text{Ker } A^\circ, \\ \text{Im } A &= \text{Im}(Q_n A^\circ), & \text{Ker } A &= Q_n(\text{Ker } A^\circ), \end{aligned}$$

and (2.2) can be rewritten as the orthogonal decomposition with respect to $\langle \cdot, \cdot \rangle_{E'}$:

$$\text{Im } \tilde{A} \oplus \text{Ker } \tilde{A} = \text{Im } A \oplus \text{Ker } A^\circ = E'.$$

Let $\tilde{\varphi}_0$ denote the linear automorphism of $\text{Im } A$ obtained as a restriction of $\tilde{A}$. So $A, \tilde{A}$, and φ_0 have the same rank r , and the direct sum $\text{Im } \tilde{A} \oplus \text{Ker } \tilde{A} = E'$ implies the following:

$$(2.13) \quad \det(I_n + z \tilde{A}) = 1 + b_1(t)z + \dots + b_r(t)z^r$$

with $b_r \neq 0$ and $b_{r+1} = \dots = b_n = 0$. As a result, Mulmuley's lemma is derived, as expected, and we have the following simplified versions of the previous results.

PROPOSITION 2.11 (generalized Moore–Penrose inverse in the symmetric case).

1. The orthogonal projection $\pi_{\text{Im } A}$ onto the subspace $\text{Im } A$ of E' with respect to $\langle \cdot, \cdot \rangle_{E'}$ is equal to

$$\pi_{\text{Im } A} = b_r^{-1}(b_{r-1} \tilde{A} - b_{r-2} \tilde{A}^2 + \cdots + (-1)^r b_1 \tilde{A}^{r-1} + (-1)^{r+1} \tilde{A}^r).$$

2. We have $A^{\dagger_{r,t}} = Q_n \tilde{A}^{\dagger_{r,t}}$ and $\tilde{A}^{\dagger_{r,t}}$ is given by

$$\tilde{A}^{\dagger_{r,t}} = b_r^{-1}(b_{r-1} \pi_{\text{Im } A} - b_{r-2} \tilde{A} + \cdots + (-1)^r b_1 \tilde{A}^{r-2} + (-1)^{r+1} \tilde{A}^{r-1}).$$

In [3], similar formulas for classical Moore–Penrose inverses of Hermitian matrices can be found.

3. The shape of the generalized Cramer rule: Examples.

3.1. Cramer’s rule for underdetermined systems of linear equations.

For the underdetermined system of linear equations,

$$(3.1) \quad \begin{pmatrix} a_{1,1} & a_{1,2} & a_{1,3} \\ a_{2,1} & a_{2,2} & a_{2,3} \end{pmatrix} \begin{pmatrix} x_1 \\ x_2 \\ x_3 \end{pmatrix} = \begin{pmatrix} b_1 \\ b_2 \end{pmatrix},$$

the generalized Gram polynomials are given by $\det(I_2 + zA A^\circ)$:

$$\begin{aligned} 1 &+ \left(a_{2,1}^2 t + (a_{1,1}^2 + a_{2,2}^2) + \frac{a_{1,2}^2 + a_{2,3}^2}{t} + \frac{a_{1,3}^2}{t^2} \right) z \\ &+ \left(\begin{vmatrix} a_{1,1} & a_{1,2} \\ a_{2,1} & a_{2,2} \end{vmatrix}^2 + \frac{\begin{vmatrix} a_{1,1} & a_{1,3} \\ a_{2,1} & a_{2,3} \end{vmatrix}^2}{t} + \frac{\begin{vmatrix} a_{1,2} & a_{1,3} \\ a_{2,2} & a_{2,3} \end{vmatrix}^2}{t^2} \right) z^2. \end{aligned}$$

Thus the matrix A has rank

- equal to 2 if and only if the polynomial

$$(3.2) \quad \begin{vmatrix} a_{1,1} & a_{1,2} \\ a_{2,1} & a_{2,2} \end{vmatrix}^2 + \frac{\begin{vmatrix} a_{1,1} & a_{1,3} \\ a_{2,1} & a_{2,3} \end{vmatrix}^2}{t} + \frac{\begin{vmatrix} a_{1,2} & a_{1,3} \\ a_{2,2} & a_{2,3} \end{vmatrix}^2}{t^2}$$

is not zero;

- equal to 1 if and only if the polynomial in (3.2) vanishes identically and the polynomial

$$(3.3) \quad a_{2,1}^2 t + (a_{1,1}^2 + a_{2,2}^2) + \frac{a_{1,2}^2 + a_{2,3}^2}{t} + \frac{a_{1,3}^2}{t^2}$$

is not zero; and

- equal to 0 if and only if both polynomials (3.2) and (3.3) vanish identically.

Assuming $\text{rank}(A) = 2$, the (simplified) solution of the considered system is presented “à la Cramer” by the following expressions describing a rational curve in $\mathbb{K}^3$:

$$\begin{aligned}
 x_1(t) &= \frac{\begin{vmatrix} a_{1,1} & a_{1,2} \\ a_{2,1} & a_{2,2} \end{vmatrix} \begin{vmatrix} b_1 & a_{1,2} \\ b_2 & a_{2,2} \end{vmatrix} + \frac{\begin{vmatrix} a_{1,1} & a_{1,3} \\ a_{2,1} & a_{2,3} \end{vmatrix} \begin{vmatrix} b_1 & a_{1,3} \\ b_2 & a_{2,3} \end{vmatrix}}{t}}{\begin{vmatrix} a_{1,1} & a_{1,2} \\ a_{2,1} & a_{2,2} \end{vmatrix}^2 + \frac{\begin{vmatrix} a_{1,1} & a_{1,3} \\ a_{2,1} & a_{2,3} \end{vmatrix}^2}{t} + \frac{\begin{vmatrix} a_{1,2} & a_{1,3} \\ a_{2,2} & a_{2,3} \end{vmatrix}^2}{t^2}}, \\
 x_2(t) &= \frac{\begin{vmatrix} a_{1,1} & a_{1,2} \\ a_{2,1} & a_{2,2} \end{vmatrix} \begin{vmatrix} a_{1,1} & b_1 \\ a_{2,1} & b_2 \end{vmatrix} + \frac{\begin{vmatrix} a_{1,2} & a_{1,3} \\ a_{2,2} & a_{2,3} \end{vmatrix} \begin{vmatrix} b_1 & a_{1,3} \\ b_2 & a_{2,3} \end{vmatrix}}{t^2}}{\begin{vmatrix} a_{1,1} & a_{1,2} \\ a_{2,1} & a_{2,2} \end{vmatrix}^2 + \frac{\begin{vmatrix} a_{1,1} & a_{1,3} \\ a_{2,1} & a_{2,3} \end{vmatrix}^2}{t} + \frac{\begin{vmatrix} a_{1,2} & a_{1,3} \\ a_{2,2} & a_{2,3} \end{vmatrix}^2}{t^2}}, \\
 x_3(t) &= \frac{\frac{\begin{vmatrix} a_{1,1} & a_{1,3} \\ a_{2,1} & a_{2,3} \end{vmatrix} \begin{vmatrix} a_{1,1} & b_1 \\ a_{2,1} & b_2 \end{vmatrix}}{t} + \frac{\begin{vmatrix} a_{1,2} & a_{1,3} \\ a_{2,2} & a_{2,3} \end{vmatrix} \begin{vmatrix} a_{1,2} & b_1 \\ a_{2,2} & b_2 \end{vmatrix}}{t^2}}{\begin{vmatrix} a_{1,1} & a_{1,2} \\ a_{2,1} & a_{2,2} \end{vmatrix}^2 + \frac{\begin{vmatrix} a_{1,1} & a_{1,3} \\ a_{2,1} & a_{2,3} \end{vmatrix}^2}{t} + \frac{\begin{vmatrix} a_{1,2} & a_{1,3} \\ a_{2,2} & a_{2,3} \end{vmatrix}^2}{t^2}}.
 \end{aligned}
 \tag{3.4}$$

Algebraically these expressions verify the system in (3.1), thus providing a rational curve in $\mathbb{K}^3$ giving a solution for (3.1) for every value of the parameter t (such that the denominator does not vanish when evaluated in t).

Knowing, for example, that

$$\begin{vmatrix} a_{1,2} & a_{1,3} \\ a_{2,2} & a_{2,3} \end{vmatrix} \neq 0,$$

the usual solution,

$$x_2 = \frac{\begin{vmatrix} b_1 & a_{1,3} \\ b_2 & a_{2,3} \end{vmatrix}}{\begin{vmatrix} a_{1,2} & a_{1,3} \\ a_{2,2} & a_{2,3} \end{vmatrix}} - \frac{\begin{vmatrix} a_{1,1} & a_{1,3} \\ a_{2,1} & a_{2,3} \end{vmatrix}}{\begin{vmatrix} a_{1,2} & a_{1,3} \\ a_{2,2} & a_{2,3} \end{vmatrix}} x_1, \quad x_3 = \frac{\begin{vmatrix} a_{1,2} & b_1 \\ a_{2,2} & b_2 \end{vmatrix}}{\begin{vmatrix} a_{1,2} & a_{1,3} \\ a_{2,2} & a_{2,3} \end{vmatrix}} - \frac{\begin{vmatrix} a_{1,2} & a_{1,1} \\ a_{2,2} & a_{2,1} \end{vmatrix}}{\begin{vmatrix} a_{1,2} & a_{1,3} \\ a_{2,2} & a_{2,3} \end{vmatrix}} x_1,$$

is obtained by reparameterizing the obtained rational curve through the simplification of the rational functions in (3.4) which provides x_2 and x_3 in terms of x_1 after performing the corresponding polynomial division with respect to t .

3.2. Cramer’s rule for overdetermined systems of linear equations. For the overdetermined system of linear equations,

$$\begin{pmatrix} a_{1,1} & a_{1,2} \\ a_{2,1} & a_{2,2} \\ a_{3,1} & a_{3,2} \end{pmatrix} \begin{pmatrix} x_1 \\ x_2 \end{pmatrix} = \begin{pmatrix} b_1 \\ b_2 \\ b_3 \end{pmatrix},
 \tag{3.5}$$

the generalized Gram polynomials are given by $\det(I_3 + zAA^\circ)$:

$$1 + \left(a_{3,1}^2 t^2 + (a_{3,2}^2 + a_{2,1}^2) t + (a_{2,2}^2 + a_{1,1}^2) + \frac{a_{1,2}^2}{t} \right) z \\ + \left(\begin{vmatrix} a_{2,1} & a_{2,2} \\ a_{3,1} & a_{3,2} \end{vmatrix}^2 t^2 + \begin{vmatrix} a_{1,1} & a_{1,2} \\ a_{3,1} & a_{3,2} \end{vmatrix}^2 t + \begin{vmatrix} a_{1,1} & a_{1,2} \\ a_{2,1} & a_{2,2} \end{vmatrix}^2 \right) z^2.$$

Under the assumption of rank 2 for the matrix A , the following curve (in $\mathbb{K}^2$) of solutions is obtained:

$$x_1(t) = \frac{\begin{vmatrix} a_{2,1} & a_{2,2} \\ a_{3,1} & a_{3,2} \end{vmatrix} \begin{vmatrix} b_2 & a_{2,2} \\ b_3 & a_{3,2} \end{vmatrix} t^2 + \begin{vmatrix} a_{1,1} & a_{1,2} \\ a_{3,1} & a_{3,2} \end{vmatrix} \begin{vmatrix} b_1 & a_{1,2} \\ b_3 & a_{3,2} \end{vmatrix} t + \begin{vmatrix} a_{1,1} & a_{1,2} \\ a_{2,1} & a_{2,2} \end{vmatrix} \begin{vmatrix} b_1 & a_{1,2} \\ b_3 & a_{2,2} \end{vmatrix}}{\begin{vmatrix} a_{2,1} & a_{2,2} \\ a_{3,1} & a_{3,2} \end{vmatrix}^2 t^2 + \begin{vmatrix} a_{1,1} & a_{1,2} \\ a_{3,1} & a_{3,2} \end{vmatrix}^2 t + \begin{vmatrix} a_{1,1} & a_{1,2} \\ a_{2,1} & a_{2,2} \end{vmatrix}^2}, \\ x_2(t) = \frac{\begin{vmatrix} a_{2,1} & a_{2,2} \\ a_{3,1} & a_{3,2} \end{vmatrix} \begin{vmatrix} a_{2,1} & b_2 \\ a_{3,1} & b_3 \end{vmatrix} t^2 + \begin{vmatrix} a_{1,1} & a_{1,2} \\ a_{3,1} & a_{3,2} \end{vmatrix} \begin{vmatrix} a_{1,1} & b_1 \\ a_{3,1} & b_3 \end{vmatrix} t + \begin{vmatrix} a_{1,1} & a_{1,2} \\ a_{2,1} & a_{2,2} \end{vmatrix} \begin{vmatrix} a_{1,1} & b_1 \\ a_{2,1} & b_2 \end{vmatrix}}{\begin{vmatrix} a_{2,1} & a_{2,2} \\ a_{3,1} & a_{3,2} \end{vmatrix}^2 t^2 + \begin{vmatrix} a_{1,1} & a_{1,2} \\ a_{3,1} & a_{3,2} \end{vmatrix}^2 t + \begin{vmatrix} a_{1,1} & a_{1,2} \\ a_{2,1} & a_{2,2} \end{vmatrix}^2}. \quad (3.6)$$

In this case, where the existence of solution is not assured, the evaluation of A does not provide the vector b . Instead, the conditions for the existence of solution are obtained:

$$\begin{pmatrix} a_{1,1} & a_{1,2} \\ a_{2,1} & a_{2,2} \\ a_{3,1} & a_{3,2} \end{pmatrix} \begin{pmatrix} x_1(t) \\ x_2(t) \end{pmatrix} = \begin{pmatrix} b_1 \\ b_2 \\ b_3 \end{pmatrix} - \frac{\begin{vmatrix} a_{1,1} & a_{1,2} & b_1 \\ a_{2,1} & a_{2,2} & b_2 \\ a_{3,1} & a_{3,2} & b_3 \end{vmatrix} \begin{pmatrix} \begin{vmatrix} a_{2,1} & a_{2,2} \\ a_{3,1} & a_{3,2} \end{vmatrix} t^2 \\ \begin{vmatrix} a_{1,1} & a_{1,2} \\ a_{3,1} & a_{3,2} \end{vmatrix} t \\ \begin{vmatrix} a_{1,1} & a_{1,2} \\ a_{2,1} & a_{2,2} \end{vmatrix} \end{pmatrix}}{\begin{vmatrix} a_{2,1} & a_{2,2} \\ a_{3,1} & a_{3,2} \end{vmatrix}^2 t^2 + \begin{vmatrix} a_{1,1} & a_{1,2} \\ a_{3,1} & a_{3,2} \end{vmatrix}^2 t + \begin{vmatrix} a_{1,1} & a_{1,2} \\ a_{2,1} & a_{2,2} \end{vmatrix}^2}. \quad (3.7)$$

The solution curve contains a true solution of the considered system (under the assumption of rank 2) if and only if, as expected,

$$\begin{vmatrix} a_{1,1} & a_{1,2} & b_1 \\ a_{2,1} & a_{2,2} & b_2 \\ a_{3,1} & a_{3,2} & b_3 \end{vmatrix} = 0.$$

Under this condition (and assumption of rank 2 for A) the solution curve in (3.6) reduces to exactly one point since every rational function defining the solution curve simplifies to a constant; depending on which maximal minor provides the rank 2 for A , a different constant is obtained for each rational function after the corresponding simplification.

3.3. Parametric solution to systems of linear equations. If the matrix A depends on some parameters, then this method provides a nice way to parameterize its rank and the solutions of any system of linear equations $Ax = v$ by using the coefficients of the characteristic polynomial AA° .

Example. We consider a linear system defined over $\mathbb{C}$, where the coefficient matrix A is given by

$$\begin{pmatrix} 4ba + 4a^2 + 4b^2 + 4 & 7 - 2a^2 - 2b^3 & 7a - 2a^3 - 2ab^3 + 3 \\ 4ba^2 + 4a^3 + 4ab^2 + 4a + 2ba + 2a^2 + 2b^2 + 2 & 7a - 2a^3 - 2ab^3 + 5a^2 + 5b^3 - 9 & 7a^2 - 2a^4 - 2a^2b^3 - 6a + 5a^3 + 5ab^3 + 1 \\ 8ba + 8a^2 + 8b^2 + 8 & 2 + a^2 + b^3 & 2a + a^3 + ab^3 + 4 \\ 4ba + 4a^2 + 4b^2 + 4 & 3 - a^2 - b^3 & 3a - a^3 - ab^3 + 1 \\ -6ba - 6a^2 - 6b^2 - 6 & 2 - 2a^2 - 2b^3 & 2a - 2a^3 - 2ab^3 - 2 \end{pmatrix},$$

and a, b are parameters taking values in $\mathbb{C}$. Observe that by applying our method, it is not necessary to consider the conjugates of the parameters appearing in the system.

We first calculate the generalized Gram polynomials in order to study the different possible ranks of A according to a and b . The matrix A° is given by

$$A^\circ = Q_3^{-1} A^t Q_5 = \begin{pmatrix} 1 & 0 & 0 \\ 0 & t^{-1} & 0 \\ 0 & 0 & t^{-2} \end{pmatrix} A^t \begin{pmatrix} 1 & 0 & 0 & 0 & 0 \\ 0 & t & 0 & 0 & 0 \\ 0 & 0 & t^2 & 0 & 0 \\ 0 & 0 & 0 & t^3 & 0 \\ 0 & 0 & 0 & 0 & t^4 \end{pmatrix}.$$

Thus, $\det(I_5 + zAA^\circ) = 1 + a_1(t)z + a_2(t)z^2 + a_3(t)z^3$. In this case $a_1(t)$ is never the zero polynomial. Moreover, with

$$\begin{aligned} a_3(t) &= a_{3,6}t^6 + a_{3,5}t^5 + a_{3,4}t^4 + \cdots + a_{3,0}, \\ a_2(t) &= a_{2,8}t^6 + a_{2,7}t^5 + \cdots + a_{2,0}t^{-2}, \end{aligned}$$

by (2.10), the Gram coefficients for $k \in \{3, 2\}$ are given by

$$a_{3,l} : \begin{cases} a_{3,0} = \mu_{\{1,2,3\},\{1,2,3\}}^2, & a_{3,1} = \mu_{\{1,2,4\},\{1,2,3\}}^2, \\ a_{3,2} = \mu_{\{1,2,5\},\{1,2,3\}}^2 + \mu_{\{1,3,4\},\{1,2,3\}}^2, \\ a_{3,3} = \mu_{\{1,3,5\},\{1,2,3\}}^2 + \mu_{\{2,3,4\},\{1,2,3\}}^2, \\ a_{3,4} = \mu_{\{2,3,5\},\{1,2,3\}}^2 + \mu_{\{1,4,5\},\{1,2,3\}}^2, \\ a_{3,5} = \mu_{\{2,4,5\},\{1,2,3\}}^2, & a_{3,6} = \mu_{\{3,4,5\},\{1,2,3\}}^2, \end{cases}$$

$$a_{2,l} : \begin{cases} a_{2,0} = \mu_{\{1,2\},\{2,3\}}^2, & a_{2,1} = \mu_{\{1,2\},\{1,3\}}^2 + \mu_{\{1,3\},\{2,3\}}^2, \\ a_{2,2} = \mu_{\{1,2\},\{1,2\}}^2 + \mu_{\{1,3\},\{1,3\}}^2 + \mu_{\{2,3\},\{2,3\}}^2 + \mu_{\{1,4\},\{2,3\}}^2, \\ a_{2,3} = \mu_{\{1,3\},\{1,2\}}^2 + \mu_{\{1,4\},\{1,3\}}^2 + \mu_{\{2,3\},\{1,3\}}^2 + \mu_{\{1,5\},\{2,3\}}^2 + \mu_{\{2,4\},\{2,3\}}^2, \\ a_{2,4} = \mu_{\{1,4\},\{1,2\}}^2 + \mu_{\{2,3\},\{1,2\}}^2 + \mu_{\{1,5\},\{1,3\}}^2 + \mu_{\{2,4\},\{1,3\}}^2 \\ \quad + \mu_{\{2,5\},\{2,3\}}^2 + \mu_{\{3,4\},\{2,3\}}^2, \\ a_{2,5} = \mu_{\{1,5\},\{1,2\}}^2 + \mu_{\{2,4\},\{1,2\}}^2 + \mu_{\{3,4\},\{1,3\}}^2 + \mu_{\{3,5\},\{2,3\}}^2 + \mu_{\{2,5\},\{1,3\}}^2, \\ a_{2,6} = \mu_{\{2,5\},\{1,2\}}^2 + \mu_{\{3,4\},\{1,2\}}^2 + \mu_{\{3,5\},\{1,3\}}^2 + \mu_{\{4,5\},\{2,3\}}^2, \\ a_{2,7} = \mu_{\{3,5\},\{1,2\}}^2 + \mu_{\{4,5\},\{1,3\}}^2, & a_{2,8} = \mu_{\{4,5\},\{1,2\}}^2, \end{cases}$$

and the different possible ranks of A according to the values of the generalized Gram polynomials are

1. $a_3(t) \neq 0 \iff (a^2 + b^3 - 2)(a^2 + 1 + ba + b^2) \neq 0 \iff \text{rk}(A) = 3;$
2. $a_3(t) = 0 \wedge a_2(t) \neq 0 \iff \text{rk}(A) = 2:$
 - $a^2 + b^3 - 2 = 0, a^2 + 1 + ba + b^2 \neq 0,$
 - $a^2 + 1 + ba + b^2 = 0, a^2 + b^3 - 2 \neq 0;$
3. $a_3(t) = 0 \wedge a_2(t) = 0 \iff \text{rk}(A) = 1:$
 - $a^6 + 2a^5 + 4a^4 - a^3 - a^2 - 6a + 5 = 0, b - a^2 + 2 - a^3 - a = 0 \Leftrightarrow a^2 + b^3 - 2 = 0, a^2 + 1 + ba + b^2 = 0.$

Once the rank is determined, in view of (2.8) and (2.9), the different possible solutions of the linear system are uniformly given by

- $\text{rk}(A) = 3: \quad \bar{x} = \frac{1}{a_3(t)} ((A^\circ A)^2 - a_1(t)A^\circ A + a_2(t)I_3) A^\circ v;$
- $\text{rk}(A) = 2:$

$$(3.8) \quad \bar{x} = \frac{1}{a_2(t)} (-A^\circ A + a_1(t)I_3) A^\circ v;$$

- $\text{rk}(A) = 1: \quad \bar{x} = \frac{1}{a_1(t)} A^\circ v.$

For instance, if we consider $a = 1, b = 1, v = (-6, -8, -8, -2, 4)$, then $\text{rk}(A) = \text{rk}(A|v) = 2$, and the solution is given by (3.8) as follows:

$$\bar{x} = \left(0, \frac{-2t}{t+4}, \frac{-4}{t+4}\right) = (0, -2, 0) + \frac{1}{t+4}(0, 8, -4)$$

such that $(0, -2, 0)$ is a concrete zero of our system and $\frac{1}{t+4}(0, 8, -4)$ is a solution of the homogeneous linear system defined by A .

Example. We consider a linear system defined over $\mathbb{Z}_{11}$, where the coefficient matrix A is given by

$$\begin{pmatrix} 7ab + 2a^2b + 9a^2 + 2a^3 + 7a & 7a + 1 + 9ab + 9b & 3 \\ 2a^3b + 7a^2b + 2a^4 + 9a^3 + 8a^2 & 6a^2 + 3a + 9a^2b + 3ab & 2a \\ & + ab + 9b + 10a + 9 & + 5b + 1 \\ 7ab + 2a^2b + 9a^2 + 2a^3 + 7a & 2a + 5 + ab + b & 4 \\ a^2b + 9ab + a^3 + 10a^2 + 9a & 3a^2 + 4 + ab + b & 5 + 3a \\ 9ab + a^3 + 2a^2 + a + 3a^3b + 10b & 3a + 7 + 7ab + 3a^2 & 3 + a + 4b \\ + a^4 + 10 + 7ab^2 + 2a^2b^2 + 9a^2b & + 10a^2b + ab^2 + b^2 & \end{pmatrix},$$

and a, b are parameters taking values in $\mathbb{Z}_{11}$.

We first calculate the generalized Gram polynomials in order to study the different possible ranks of A according to a and b . The matrix A° is given by

$$A^\circ = Q_3^{-1} A^t Q_5 = \begin{pmatrix} 1 & 0 & 0 \\ 0 & t^{-1} & 0 \\ 0 & 0 & t^{-2} \end{pmatrix} A^t \begin{pmatrix} 1 & 0 & 0 & 0 & 0 \\ 0 & t & 0 & 0 & 0 \\ 0 & 0 & t^2 & 0 & 0 \\ 0 & 0 & 0 & t^3 & 0 \\ 0 & 0 & 0 & 0 & t^4 \end{pmatrix}.$$

Thus, $\det(I_5 + zAA^\circ) = 1 + a_1(t)z + a_2(t)z^2 + a_3(t)z^3$. Note that $a_1(t)$ is never the zero polynomial; in fact, $a_{1,0} = 9$. Moreover, with

$$a_3(t) = a_{3,6}t^6 + a_{3,5}t^5 + a_{3,4}t^4 + \cdots + a_{3,0}, \quad a_2(t) = a_{2,8}t^6 + a_{2,7}t^5 + \cdots + a_{2,0}t^{-2},$$

by (2.10), the Gram coefficients for $k \in \{3, 2\}$ are given by

$$a_{3,l} : \begin{cases} a_{3,0} = \mu_{\{1,2,3\},\{1,2,3\}}^2, & a_{3,1} = \mu_{\{1,2,4\},\{1,2,3\}}^2, \\ a_{3,2} = \mu_{\{1,2,5\},\{1,2,3\}}^2 + \mu_{\{1,3,4\},\{1,2,3\}}^2, \\ a_{3,3} = \mu_{\{1,3,5\},\{1,2,3\}}^2 + \mu_{\{2,3,4\},\{1,2,3\}}^2, \\ a_{3,4} = \mu_{\{2,3,5\},\{1,2,3\}}^2 + \mu_{\{1,4,5\},\{1,2,3\}}^2, \\ a_{3,5} = \mu_{\{2,4,5\},\{1,2,3\}}^2, & a_{3,6} = \mu_{\{3,4,5\},\{1,2,3\}}^2, \end{cases}$$

$$a_{2,l} : \begin{cases} a_{2,0} = \mu_{\{1,2\},\{2,3\}}^2, & a_{2,1} = \mu_{\{1,2\},\{1,3\}}^2 + \mu_{\{1,3\},\{2,3\}}^2, \\ a_{2,2} = \mu_{\{1,2\},\{1,2\}}^2 + \mu_{\{1,3\},\{1,3\}}^2 + \mu_{\{2,3\},\{2,3\}}^2 + \mu_{\{1,4\},\{2,3\}}^2, \\ a_{2,3} = \mu_{\{1,3\},\{1,2\}}^2 + \mu_{\{1,4\},\{1,3\}}^2 + \mu_{\{2,3\},\{1,3\}}^2 + \mu_{\{1,5\},\{2,3\}}^2 + \mu_{\{2,4\},\{2,3\}}^2, \\ a_{2,4} = \mu_{\{1,4\},\{1,2\}}^2 + \mu_{\{2,3\},\{1,2\}}^2 + \mu_{\{1,5\},\{1,3\}}^2 + \mu_{\{2,4\},\{1,3\}}^2 \\ \quad + \mu_{\{2,5\},\{2,3\}}^2 + \mu_{\{3,4\},\{2,3\}}^2, \\ a_{2,5} = \mu_{\{1,5\},\{1,2\}}^2 + \mu_{\{2,4\},\{1,2\}}^2 + \mu_{\{3,4\},\{1,3\}}^2 + \mu_{\{3,5\},\{2,3\}}^2 + \mu_{\{2,5\},\{1,3\}}^2, \\ a_{2,6} = \mu_{\{2,5\},\{1,2\}}^2 + \mu_{\{3,4\},\{1,2\}}^2 + \mu_{\{3,5\},\{1,3\}}^2 + \mu_{\{4,5\},\{2,3\}}^2, \\ a_{2,7} = \mu_{\{3,5\},\{1,2\}}^2 + \mu_{\{4,5\},\{1,3\}}^2, & a_{2,8} = \mu_{\{4,5\},\{1,2\}}^2, \end{cases}$$

and the different possible ranks of A according to the values of the generalized Gram polynomials are

1. $a_3(t) \neq 0 \iff (a+b+1)(b-2)(a-10)(a-2) \neq 0 \iff \text{rk}(A) = 3$;
2. $a_3(t) = 0 \wedge a_2(t) \neq 0 \iff \text{rk}(A) = 2$:
 - $a = 2, b \neq 2$,
 - $b = 2, (a-2)(a-8) \neq 0$,
 - $a = 10, b \neq 0$,
 - $a+b+1 = 0, b(b-2) \neq 0$;
3. $a_3(t) = 0 \wedge a_2(t) = 0 \iff \text{rk}(A) = 1$:
 - $a = 2, b = 2$,

- $a = 8, b = 2,$
- $a = 10, b = 0.$

Once the rank is determined, in view of (2.8) and (2.9), the different possible solutions of the linear system are uniformly given by

- $\text{rk}(A) = 3:$ $\bar{x} = \frac{1}{a_3(t)} ((A^\circ A)^2 - a_1(t)A^\circ A + a_2(t)I_3) A^\circ v;$

- $\text{rk}(A) = 2:$

$$(3.9) \quad \bar{x} = \frac{1}{a_2(t)} (-A^\circ A + a_1(t)I_3) A^\circ v;$$

- $\text{rk}(A) = 1:$ $\bar{x} = \frac{1}{a_1(t)} A^\circ v.$

For instance, if we consider $a = 3, b = 2, v = (9, 5, 0, 2, 6)$, then $\text{rk}(A) = \text{rk}(A|v) = 2$, and the solution is given by (3.9) as follows:

$$\bar{x} = \left(1, \frac{t}{t+3}, \frac{6}{t+3}\right) = (1, 1, 0) + \frac{1}{t+3}(0, 8, 6)$$

such that $(1, 1, 0)$ is a concrete zero of our system and $\frac{1}{t+3}(0, 8, 6)$ is a solution of the homogeneous linear system defined by A .

REFERENCES

- [1] C. BALLARIN AND M. KAUSERS, *Solving parametric linear systems: An experiment with constraint algebraic programming*, SIGSAM Bull., 38 (2004), pp. 33–46.
- [2] R. K. BHASKARA, *The Theory of Generalized Inverses over a Commutative Ring*, Taylor & Francis, London, 2002.
- [3] D. BINI AND V. Y. PAN, *Polynomial and Matrix Computations*, Progr. Theoret. Comput. Sci., Birkhäuser Boston, Boston, 1994.
- [4] A. BORODIN, J. VON ZUR GATHEN, AND J. HOPCROFT, *Fast parallel matrix and GCD computations*, Inform. and Control, 52 (1982), pp. 241–256.
- [5] R. DAUTRAY AND J.-L. LIONS, *Mathematical Analysis and Numerical Methods for Science and Technology*, Springer, Berlin, 1988.
- [6] O. DESSOMBZ, F. THOUVEREZ, J. P. LAINÉ, AND L. JÉZÉQUEL, *Analysis of mechanical systems using interval computations applied to finite elements methods*, J. Sound Vibration, 239 (2001), pp. 949–968.
- [7] G. DIAZ-TOCA, L. GONZALEZ-VEGA, H. LOMBARDI, AND C. QUITTÉ, *Modules projectifs de type fini, applications linéaires croisées et inverses généralisés*, preprint, 2005.
- [8] H. FISCHER, *Automatic differentiation of the vector that solves a parametric linear system*, J. Comput. Appl. Math., 35 (1991), pp. 169–184.
- [9] G. H. GOLUB AND C. F. VAN LOAN, *Matrix Computations*, 3rd ed., Johns Hopkins University Press, Baltimore, MD, 1996.
- [10] D. A. HARVILLE, *Matrix Algebra from a Statistician's Perspective*, Springer, Berlin, 2000.
- [11] J. HEINTZ, *Definability and fast quantifier elimination in algebraically closed fields*, Theoret. Comput. Sci., 24 (1983), pp. 239–277.
- [12] L. V. KOLEV, *Outer solution of linear systems whose elements are affine functions of interval parameters*, Reliab. Comput., 8 (2002), pp. 493–501.
- [13] K. JENSEN, *Coloured Petri Nets*, Springer, Berlin, 1996.
- [14] P. LANCASTER AND M. TISMENETSKY, *The Theory of Matrices*, 2nd ed., Academic Press, New York, 1985.
- [15] R. MUHANNA AND R. L. MULLEN, *Uncertainty in mechanics problems—interval-based approach*, J. Engrg. Mech., 127 (2001), pp. 557–566.
- [16] K. MULMULEY, *A fast parallel algorithm to compute the rank of a matrix over an arbitrary field*, Combinatorica, 7 (1987), pp. 101–104.

- [17] K. PRASAD AND R. BAPAT, *The generalized Moore–Penrose inverse*, Linear Algebra Appl., 165 (1992), pp. 59–69.
- [18] W. Y. SIT, *An algorithm for solving parametric linear systems*, J. Symbolic Comput., 13 (1992), pp. 353–394.
- [19] H.-M. WINKELS AND M. MEIKA, *An integration of efficiency projections into the Geoffrion approach for multiobjective linear programming*, European J. Oper. Res., 16 (1984), pp. 113–127.

ON THE COMPLEXITY OF THE PRECONDITIONED CONJUGATE GRADIENT ALGORITHM FOR SOLVING TOEPLITZ SYSTEMS WITH A FISHER–HARTWIG SINGULARITY*

YI LU[†] AND CLIFFORD M. HURVICH[†]

Abstract. The Toeplitz matrix T_n with generating function $f(\omega) = |1 - e^{-i\omega}|^{-2d}h(\omega)$, where $d \in (-\frac{1}{2}, \frac{1}{2}) \setminus \{0\}$ and $h(\omega)$ is positive, continuous on $[-\pi, \pi]$, and differentiable on $[-\pi, \pi] \setminus \{0\}$, has a Fisher–Hartwig singularity [M. E. Fisher and R. E. Hartwig (1968), *Adv. Chem. Phys.*, 32, pp. 190–225]. The complexity of the preconditioned conjugate gradient (PCG) algorithm is known [R. H. Chan and M. Ng (1996), *SIAM Rev.*, 38, pp. 427–482] to be $O(n \log n)$ for Toeplitz systems when $d = 0$. However, the effect on the PCG algorithm of the Fisher–Hartwig singularity in T_n has not been explored in the literature. We show that the complexity of the conjugate gradient (CG) algorithm for solving $T_n x = b$ without any preconditioning grows asymptotically as $n^{1+|d|} \log(n)$. With T. Chan’s optimal circulant preconditioner C_n [T. Chan (1988), *SIAM J. Sci. Statist. Comput.*, 9, pp. 766–771], the complexity of the PCG algorithm is $O(n \log^3(n))$.

Key words. conjugate gradient algorithm, preconditioned conjugate gradient algorithm, Toeplitz matrix, circulant matrix, Fisher–Hartwig singularity, expected periodogram, spectral density

AMS subject classifications. 15A12, 62M10, 62M15, 62M20, 65F10, 65F15

DOI. 10.1137/040612117

1. Introduction. The (preconditioned) conjugate gradient algorithm is a very popular and effective iterative method for solving a symmetric positive definite system $T_n x = b$. It is well known (see Axelsson and Barker (1984) and Golub and Van Loan (1996)) that the number of iterations required to reach a specified degree of relative reduction in the error in the worst case is proportional to the square root of the condition number of T_n . Assume T_n has the Toeplitz structure

$$T_n = \begin{bmatrix} t_0 & t_{-1} & \dots & t_{2-n} & t_{1-n} \\ t_1 & t_0 & t_{-1} & \dots & t_{2-n} \\ \vdots & t_1 & t_0 & \dots & \vdots \\ t_{n-2} & \dots & \dots & \dots & t_{-1} \\ t_{n-1} & t_{n-2} & \dots & t_1 & t_0 \end{bmatrix}$$

and is symmetric with $t_k = t_{-k}$. The $\{t_r\}$ sequence is the Fourier transform of the generating function $f(\omega)$ of T_n , that is,

$$t_k = \int_{-\pi}^{\pi} f(\omega) \exp(-ik\omega) d\omega, \quad k = 0, \pm 1, \pm 2, \dots$$

We assume the generating function $f(\omega)$ has the following form.

Assumption 1.

$$f(\omega) = |1 - e^{-i\omega}|^{-2d} h(\omega), \quad \omega \in [-\pi, \pi],$$

*Received by the editors July 23, 2004; accepted for publication (in revised form) by D. B. Szyld March 30, 2005; published electronically December 7, 2005.

<http://www.siam.org/journals/simax/27-3/61211.html>

[†]Stern School of Business, New York University, 44 W. 4th Street, New York, NY 10012 (ylu@stern.nyu.edu, churvich@stern.nyu.edu).

where $d \in (\frac{-1}{2}, \frac{1}{2}) \setminus \{0\}$, $h(\omega)$ is even, positive, continuous on $[-\pi, \pi]$, and differentiable on $[-\pi, \pi] \setminus \{0\}$. Also $\log(h)$ is Riemann integrable on $[-\pi, \pi]$,

$$\int_{-\pi}^{\pi} \left| \frac{h'(\omega)}{h(\omega)} \right| d\omega < \infty,$$

there exists $c \in (0, \infty)$ such that

$$|h'(\omega)| \leq c|\omega|^{-1}, \quad \omega \in [-\pi, \pi] \setminus \{0\},$$

and $h(\omega) \in B_1^1 \cap L_\infty$, where the Besov space B_1^1 includes any function g such that g is in L^1 and

$$\int_{-\pi}^{\pi} \frac{1}{t^2} \int_{-\pi}^{\pi} |g(\omega+t) - 2g(\omega) + g(\omega-t)| d\omega dt < \infty.$$

Finally, we assume that all conditions for $h(\omega)$ also apply to $\frac{1}{h(\omega)}$.

Under Assumption 1, T_n has a Fisher-Hartwig singularity caused by the term $|1 - e^{-i\omega}|^{-2d}$ in the generating function. We have $f(\omega) \sim C|\omega|^{-2d}$ as $\omega \rightarrow 0^+$ and $t_r \sim \tilde{C}r^{2d-1}$ as $r \rightarrow \infty$, with $C > 0$ and $\tilde{C} \neq 0$. The condition number of T_n approaches ∞ as n increases. An example of such a Toeplitz matrix is the covariance matrix of a long-memory time series. The importance of long-memory models (i.e., $d \neq 0$) has been widely demonstrated (see, e.g., Beran (1994) and Robinson (1994)).

For the case when $d = 0$, R. H. Chan and M. K. Ng (1996) and R. H. Chan and M. Yeung (1992a) have shown that the complexity of the preconditional conjugate gradient (PCG) algorithm is $O(n \log(n))$ provided that a suitable preconditioning matrix M is chosen (e.g., the preconditioners of Strang (1986) or T. Chan (1988)). Only $O(1)$ iterations are necessary in this case because of the strong clustering of the eigenvalues of the preconditioned system. Each iteration requires $O(n \log(n))$ computations using the fast Fourier transform (FFT). This superlinear convergence (which implies $O(1)$ iterations) for these popular circulant preconditioners is proved under weaker assumptions on f (e.g., positive functions in the Wiener class, i.e., absolutely summable Fourier coefficients, by R. H. Chan (1989) and R. H. Chan and G. Strang (1989), and continuous and positive by Serra (1999) through the matrix Korovkin theory). Furthermore, optimal convergence (i.e., $O(1)$ iterations) is proved for the case when f is nonnegative and continuous with a zero of order $2p$, where p is a positive integer, if a higher-order generalized Jackson kernel generated circulant preconditioner is applied (Chan, Yip, and Ng (2000)). Serra (1995) considered the case when f has zeros of any positive real order p when a Toeplitz preconditioner is applied. It is found that the complexity of the PCG algorithm is $O(n^{1+c(p)} \log(n))$ where $c(p) \in [0, \frac{1}{2}]$ and $c(p)$ takes the maximum value $\frac{1}{2}$ when f has a zero of odd order.

Unfortunately, when $d \neq 0$, the assumptions on f in the existing results on PCG are generally violated. If $d \in (0, \frac{1}{2})$, then f is divergent at zero and therefore f cannot belong to the Wiener class. On the other hand, if $-\frac{1}{2} < d < 0$, then f has zero of order $2|d|$. We are not aware of any theoretical results on the complexity of the PCG algorithm for the cases $d \in (-\frac{1}{2}, \frac{1}{2}) \setminus \{0\}$.

In Table 1, we present numerical results on the number of iterations required to achieve a norm of the residual vector $\|T_n x - b\| \leq 10^{-20}$ in the nonpreconditioned conjugate gradient (CG) algorithm and the PCG algorithm with T. Chan's preconditioner, the generalized Jackson kernel generated preconditioner (Chan, Yip, and Ng

(2000)) of order 2 and 3, and the cosine bell kernel (Bloomfield (2000)) generated preconditioner for solving a Toeplitz system $T_n x = b$ satisfying Assumption 1. The generating function is given by (see Brockwell and Davis (1991, eq. (13.2.7)))

$$(1) \quad f(\omega) = \frac{\sigma_s^2}{2\pi} |2 \sin(\omega/2)|^{-2d}, \quad -\pi \leq \omega \leq \pi,$$

with $d = 0.37$, $\sigma_s^2 = 0.27$, and

$$(2) \quad t_k = \frac{\sigma_s^2 \Gamma(1-2d) \Gamma(k+d)}{\Gamma(d) \Gamma(1-d) \Gamma(k-d+1)}, \quad k = 0, 1, 2, \dots$$

Also, we choose $b' = (t_1, t_2, \dots, t_n)$ as in the Yule–Walker equations in time series literature described in Brockwell and Davis (1991). The results in Table 1 confirm that preconditioning is beneficial. We will prove that the number of iterations for the CG algorithm without preconditioning is $O(n^{|d|})$, but with T. Chan’s preconditioner, the PCG algorithm requires only $O(\log^2(n))$ iterations.

TABLE 1
Number of iterations for CG and PCG algorithms: $n = 2^4$ to 2^{15} .

n	2^4	2^5	2^6	2^7	2^8	2^9	2^{10}	2^{11}	2^{12}	2^{13}	2^{14}	2^{15}
CG (no prec.)	14	18	25	32	42	54	67	88	112	143	186	238
PCG (Strang)	16	18	16	14	13	13	12	13	13	13	13	13
PCG (T. Chan)	7	7	7	8	9	8	9	9	10	11	10	11
PCG (Jackson, $r = 2$)	8	9	9	9	10	10	11	12	11	12	12	12
PCG (Jackson, $r = 3$)	9	10	10	10	10	11	11	11	13	13	13	14
PCG (cosine bell)	7	7	8	8	8	9	9	9	10	11	12	12

The number of iterations for the PCG algorithm in Table 1 appears in fact to grow much slower than $\log(n)$. This suggests that there is the potential of improvement on the results in this paper. Chen, Hurvich, and Lu (2004) indeed provided a tighter bound $O(\log^{3/2}(n))$ for the number of iterations by exploiting additional results from the time series literature.

In this paper, we concentrate on T. Chan’s (1988) optimal circulant preconditioner even though many successful circulant preconditioners have been proposed in the literature (R. H. Chan and M. Ng (1996)). We will exploit the close relationship between T. Chan’s preconditioner and time series theory. With the difference of a constant 2π , T. Chan’s preconditioner is defined as a Toeplitz matrix C_n with j th diagonals $\tilde{t}_j$ given by

$$\tilde{t}_j = \begin{cases} \frac{1}{2\pi} \frac{(n-j)t_j + jt_{j-n}}{n}, & 0 \leq j < n, \\ \tilde{t}_{n+j}, & 0 < -j < n. \end{cases}$$

The structure of this paper is as follows. Section 2 introduces some concepts in time series and their relation to T. Chan’s circulant preconditioner and Toeplitz system. Section 3 studies the extreme eigenvalues of the Toeplitz matrix $T_n[f]$ with a Fisher–Hartwig singularity, thereby deriving the condition number of $T_n[f]$ and the complexity of the ordinary CG algorithm. Section 4 studies the extreme eigenvalues of the preconditioned Toeplitz matrix when T. Chan’s circulant preconditioner is applied. Section 5 gives the resulting condition number of the preconditioned Toeplitz matrix and the complexity for the PCG algorithm. Proofs of the lemmas are given in the appendix (section 6).

2. T. Chan's circulant preconditioner, Toeplitz system, and related concepts in time series theory. For any function g that is positive and continuous, we define a circulant matrix

$$(3) \quad C_n[g] = F_n^* \Lambda_g F_n,$$

where $F_n^* = \{V_0^*, V_1^*, \dots, V_{n-1}^*\}$, $\Lambda_g = \text{diag}\{g(\omega_0), \dots, g(\omega_{n-1})\}$, and

$$V_j = \frac{1}{\sqrt{n}} \{1, \exp(i\omega_j), \exp(i2\omega_j), \dots, \exp(i(n-1)\omega_j)\},$$

where $\omega_j = \frac{2\pi j}{n}$ is the j th Fourier frequency. Note that F_n is an $n \times n$ unitary matrix, that is, $F_n F_n^* = F_n^* F_n = \mathbf{I}$, an $n \times n$ identity matrix.

As Chan and Yeung (1992b) have shown, we have the following important property for T. Chan's preconditioner.

LEMMA 1. *The eigenvalues λ_k of T. Chan's preconditioner C_n are given by*

$$(4) \quad \lambda_k = \frac{1}{2\pi} \sum_{|j| < n} t_j \left(1 - \frac{|j|}{n}\right) \exp(i\omega_j k) \quad , \quad k = 0, \dots, n-1.$$

Note that the λ_k in (4) can also be expressed as the convolution of the Fejer kernel K_n and f , that is,

$$(5) \quad \lambda_k = (K_n * f)(\omega_k),$$

where

$$(K_n * f)(\omega) = \int_{-\pi}^{\pi} K_n(\omega - \lambda) f(\lambda) d\lambda$$

and

$$K_n(\omega) = \frac{\sin^2(n\omega/2)}{2\pi n \sin^2(\omega/2)}$$

is the Fejer kernel. We use $(K_n * f)_j$ to represent $(K_n * f)(\omega_j)$.

Therefore, T. Chan's circulant preconditioner can be written as

$$C_n[K_n * f] = F_n^* \Lambda_{K_n * f} F_n,$$

where $\Lambda_{K_n * f}$ is a diagonal matrix with diagonal elements $(K_n * f)_0, (K_n * f)_1, \dots, (K_n * f)_{n-1}$. Our assumptions on f imply that $K_n * f$ is always positive. Thus, $C_n[K_n * f]$ is positive definite.

Suppose we have weakly stationary time series data $\{y_t\}_{t=0}^{n-1}$ with spectral density function $f(\omega)$, i.e., the generating function of matrix T_n . The discrete Fourier transform (DFT) of the time series at Fourier frequency ω_j is denoted by J_j and given by

$$J_j = \frac{1}{n} \sum_{t=0}^{n-1} y_t \exp(-i\omega_j t), \quad j = 0, 1, \dots, n-1.$$

The periodogram is defined by

$$I_j = \frac{n}{2\pi} |J_j|^2, \quad j = 0, 1, \dots, n-1.$$

Then it is well known (Bloomfield (2000)) that the expected value of the periodogram is given by

$$E(I_j) = (K_n * f)(\omega_j), \quad j = 0, 1, \dots, n-1.$$

In view of (5), the eigenvalues of $C_n[K_n * f]$ are therefore given by

$$(6) \quad \lambda_j = E(I_j), \quad j = 0, 1, \dots, n-1.$$

Equation (6) reveals an important link between T. Chan's circulant preconditioners and the frequency domain analysis of a time series. In particular, any existing theory on the expected value of the periodogram at the Fourier frequencies yields information on the behavior of the eigenvalues of T. Chan's circulant preconditioner.

If $f(\omega)$ satisfies Assumption 1, we call $\{y_t\}_{t=0}^{n-1}$ a time series with long memory (Koopmans (1974) and Brockwell and Davis (1991)). Results on the expected value of the periodogram for long-memory time series have recently been given by Hurvich, Deo, and Brodsky (1998) and by Moulines and Soulier (1999), and such results will play a key role in our subsequent derivation of the computational complexity of the PCG algorithm using T. Chan's circulant preconditioner. In fact, the interconnection between periodograms and preconditioners extends well beyond the case of C_n . We will not pursue this here, however.

The solution of Toeplitz systems plays an important role in time series analysis. Covariance matrix of a time series has Toeplitz structure. When the time series has long memory, the covariance matrix has Fisher–Hartwig singularity. Levinson's (1946) algorithm has been the standard method for solving Toeplitz system in statistics literature. In this paper, we show that the PCG algorithm is much more efficient for long-memory time series. The following examples show the importance of efficient solution of a Toeplitz system in time series applications.

- Given a zero-mean weakly stationary time series $\{y_t\}_{t=0}^{n-1}$ with spectral density f , the minimum mean squared error linear predictor based on $Y = (y_0, \dots, y_{n-1})'$ is obtained as the solution for x in a linear system of form $T_n(f)x = b$ involving the $n \times n$ Toeplitz covariance matrix $T_n(f) = \text{cov}(Y)$, where b is known. This system is called Yule–Walker equations in time series literature as described in Brockwell and Davis (1991).
- Suppose we can assume a parametric model, indexed by a vector θ , for $\{y_t\}$. Let $T_{n,\theta}$ denote the $n \times n$ covariance matrix for $(y_0, \dots, y_{n-1})$ under the model θ . To estimate the model parameter θ , we need to evaluate the Gaussian log likelihood function (Brockwell and Davis (1991)) which, multiplied by -2 , is

$$(7) \quad -2 \log L(\theta) = n \log 2\pi + \log(\det(T_{n,\theta})) + y' T_{n,\theta}^{-1} y.$$

Therefore, maximum likelihood estimation of this type of model require fast evaluation of Toeplitz system $T_{n,\theta}^{-1} y$.

- A long-memory stochastic volatility (LMSV) model in Deo, Hurvich, and Lu (2005) for financial returns $\{r_t\}$ is given by

$$(8) \quad r_t = \sigma \exp(y_t/2) \varepsilon_t,$$

where $\sigma > 0$, the ε_t are independently and identically distributed with mean zero and variance 1. The volatility process y_t is a latent long-memory time series that cannot be observed directly. The spectral density of $\{\log r_t^2\}$ is

$f(\omega) = f_1(\omega) + \text{var}(\log \epsilon_t^2)/(2\pi)$, where $f_1(\omega)$ is the spectral density of $\{y_t\}$. As pointed out in Harvey (1998), the minimum mean square linear estimator (MMSLE) of latent process y_t is given by

$$(9) \quad \hat{y} = T_y T_z^{-1}(z - \mu e),$$

where $\mu = E[\log r_t^2]$, $z = (\log r_0^2, \log r_1^2, \dots, \log r_{n-1}^2)'$, T_z is the covariance matrix of z , T_y is the covariance matrix of $(y_0, \dots, y_{n-1})'$, and e is an $n \times 1$ vector of ones. In (9), T_z and T_y are both $n \times n$ Toeplitz matrices. The PCG algorithm is necessary to evaluate $T_z^{-1}(z - \mu e)$ efficiently, especially when n is large and y has long memory.

3. The largest and smallest eigenvalues of the Toeplitz matrix with Fisher–Hartwig singularity. From now on, we denote the largest and smallest eigenvalue of any matrix A as $\lambda_{\max}(A)$ and $\lambda_{\min}(A)$, respectively. We also define the notion of asymptotic equivalence: $f(n) \sim g(n)$ if and only if there exist two positive universal constants c and C such that uniformly

$$cg(n) \leq f(n) \leq Cg(n).$$

By the results from Brockwell and Davis [(1991, Proposition 4.5.3, pp. 137)], we know that $\lambda_{\min}(T_n[f]) \geq 2\pi C$ if $f(\omega) \geq C > 0$ and $\lambda_{\max}(T_n[f]) \leq 2\pi C$ if $f(\omega) \leq C < \infty$, where C represents any positive constant. Moreover if f is not identically equal to C , then the above inequalities are strict (see Serra (1998a)).

Therefore, the following hold:

- If $d \in (0, \frac{1}{2})$, f is bounded below by a positive constant. Therefore, the smallest eigenvalue $\lambda_{\min}(T_n[f]) \geq C_{\min}$ for some constant $C_{\min} > 0$ that is independent of n . For the largest eigenvalue, we use the fact (Grenander and Szegő (1984)) that the largest eigenvalue of a Hermitian matrix is bounded by its ∞ -norm. That is,

$$\lambda_{\max}(T_n[f]) \leq \|T_n[f]\|_{\infty},$$

where $\|A\|_{\infty}$ is defined as the maximum of the sums of the moduli of the elements in each row of the matrix A . The sums of the moduli of the elements in each row in $T_n[f]$ are bounded by

$$2(|t_0| + |t_1| + \dots + |t_{n-1}|) \sim 2 \sum_{j=0}^{n-1} j^{2d-1} \sim 2C_0 n^{2d},$$

where C_0 is a positive constant independent of d . Therefore,

$$\lambda_{\max}(T_n[f]) = O(n^{2d}).$$

Moreover if f satisfies Assumption 1 and h is a positive constant, then the t_r are positive numbers asymptotic to j^{2d-1} and therefore $\sum_{j=1}^n (T_n[f])_{i,j} \sim n^{2d}$ for every $i \in \{1, \dots, n\}$. Consequently, if we denote by e the vector of size n and composed by all ones, then there exists a positive constant C such that

$$\frac{e^T T_n[f] e}{e^T e} \geq C n^{2d},$$

which implies that $\lambda_{\max}(T_n[f]) \geq Cn^{2d}$. Therefore by putting together the latter and $\lambda_{\max}(T_n[f]) = O(n^{2d})$ we deduce

$$\lambda_{\max}(T_n[f]) \sim n^{2d}.$$

Now if we are under the general Assumption 1 (i.e., h is not necessarily constant), then by the positivity and boundedness of h we deduce that $f \sim f/h$ and therefore by Serra (1998a) we have $\lambda_{\max}(T_n[f]) \sim n^{2d}$ in the general case as well.

- If $d \in (-\frac{1}{2}, 0)$, f is bounded above by a positive constant, so the largest eigenvalue $\lambda_{\max}(T_n[f]) \leq C_{\max}$ for some constant $C_{\max} > 0$ that is independent of n . By Böttcher and Grudsky (1998), we also have

$$\lambda_{\min}(T_n[f]) \sim n^{2d}, \quad d \in \left(-\frac{1}{2}, 0\right).$$

In summary, for any $d \in (-\frac{1}{2}, \frac{1}{2})$, the condition number of $T_n[f]$ is

$$K(T_n[f]) \sim n^{2|d|}.$$

Therefore, the number of iterations in the ordinary CG algorithm is upper-bounded by $C\sqrt{K(T_n[f])} \sim n^d$ with C positive universal constant: moreover, for n large enough, the eigenvalues of $T_n[f]$ behave as an equispaced sampling of f over $(0, \pi)$ (see, e.g., Serra (1998b)), and therefore, by Axelsson and Lindskog (1986), it follows that the number of CG iterations is exactly asymptotic to n^d . Since each iteration needs $O(n \log(n))$ operations using the FFT, the total complexity of the CG algorithm grows exactly as $n^{1+|d|} \log(n)$. This upper bound for the complexity grows with $|d|$ and approaches $Cn^{1.5} \log(n)$ for some universal positive constant C . We will show that the PCG algorithm using T. Chan's preconditioner has a lower computational complexity, $O(n \log^3(n))$, which moreover does not depend on $d \in (-\frac{1}{2}, \frac{1}{2})$.

4. The largest and smallest eigenvalues of the circulant preconditioned Toeplitz covariance matrix. Using the notation $A \approx B$ if the matrices A and B are similar, we have

$$\begin{aligned} C_n^{-1}[K_n * f]T_n[f] &\approx C_n^{-\frac{1}{2}}[K_n * f]T_n[f]C_n^{-\frac{1}{2}}[K_n * f] \\ &\approx \Lambda_{K_n * f}^{-\frac{1}{2}}F_nT_n[f]F_n^*\Lambda_{K_n * f}^{-\frac{1}{2}}. \end{aligned}$$

Define

$$(10) \quad M = \Lambda_{K_n * f}^{-\frac{1}{2}}F_nT_n[f]F_n^*\Lambda_{K_n * f}^{-\frac{1}{2}},$$

Then $C_n^{-1}[K_n * f]T_n[f] \approx M$, and therefore the two matrices have the same condition number.

The (j, k) entry of $F_nT_n[f]F_n^*$ is

$$\begin{aligned} V_jT_n[f]V_k^* &= \frac{1}{n} \sum_{t=0}^{n-1} \sum_{s=0}^{n-1} \exp(i\omega_j t) \exp(-i\omega_k s) \int_{-\pi}^{\pi} f(\lambda) \exp(-i(t-s)\lambda) d\lambda \\ &= \frac{1}{n} \int_{-\pi}^{\pi} f(\lambda) \sum_{t=0}^{n-1} \exp(i(\omega_j - \lambda)t) \sum_{s=0}^{n-1} \exp(-i(\omega_k - \lambda)s) d\lambda \\ &= 2\pi \int_{-\pi}^{\pi} f(\lambda) \overline{D_n(\omega_j - \lambda)} D_n(\omega_k - \lambda) d\lambda, \end{aligned}$$

where

$$D_n(\lambda) = \frac{1}{\sqrt{2\pi n}} \sum_{t=0}^{n-1} \exp(-i\lambda t)$$

is the Dirichlet kernel. So

$$M_{j,k} = 2\pi \int_{-\pi}^{\pi} f(\lambda) \overline{D_n(\omega_j - \lambda)} D_n(\omega_k - \lambda) / \sqrt{(K_n * f)_j (K_n * f)_k} d\lambda$$

for $j, k = 0, 1, \dots, n-1$. It is worth noting that

$$\int_{-\pi}^{\pi} f(\lambda) \overline{D_n(\omega_j - \lambda)} D_n(\omega_k - \lambda) / \sqrt{(K_n * f)_j (K_n * f)_k} d\lambda$$

is the correlation of the DFT of the time series at Fourier frequencies ω_j and ω_k .

Note that M is a Hermitian positive definite matrix and for $k = 0, 1, \dots, n-1$, diagonal element

$$\begin{aligned} M_{k,k} &= 2\pi \int_{-\pi}^{\pi} f(\lambda) |D_n(\omega_k - \lambda)|^2 / (K_n * f)_k d\lambda \\ &= 2\pi \int_{-\pi}^{\pi} f(\lambda) K_n(\omega_k - \lambda) d\lambda / (K_n * f)_k \\ &= 2\pi \end{aligned}$$

since $K_n(\lambda) = D_n^2(\lambda)$ and $f(\lambda)$ is symmetric around 0. It follows that the trace of M is

$$(11) \quad \text{tr}(M) = 2\pi n.$$

4.1. The largest eigenvalue of M . Now we study the largest eigenvalue of the preconditioned matrix, or equivalently, the largest eigenvalue of M using the following theorem (Merikoski and Virtanen (1997), Theorem 1):

Let A be an $n \times n$ matrix with real and positive eigenvalues. Then

$$(12) \quad \lambda_{\max}(A) \leq \text{tr}(A) - (n-1) \left\{ \frac{\det(A)}{\text{tr}(A)} \right\}^{\frac{1}{n-1}}.$$

By definition of M in (10)

$$\begin{aligned} \det(M) &= \det(\Lambda_{K_n * f}^{-\frac{1}{2}}) \det(F_n) \det(T_n[f]) \det(F_n^*) \det(\Lambda_{K_n * f}^{-\frac{1}{2}}) \\ &= \frac{\det(T_n[f])}{\det(\Lambda_{K_n * f})}. \end{aligned}$$

We will compute $\det(M)$ with the following lemmas, which are proved in the appendix.

LEMMA 2. *With f satisfying Assumption 1,*

$$\det(T_n[f]) \sim (2\pi)^n G(h)^n n^{d^2} E$$

as $n \rightarrow \infty$, where

$$G(h) = \exp((\log(h))_0),$$

E is a positive constant independent of n , and

$$(\log(h))_0 = \frac{1}{2\pi} \int_0^{2\pi} \log(h(\omega)) d\omega.$$

LEMMA 3. With f satisfying Assumption 1,

$$\det(\Lambda_{K_n * f}) = O \left(n^{C \log(n)} \prod_{j=1}^{n-1} h_{n,j} \right),$$

where $h_{n,j} = h(\omega_j)$ and C is a positive constant independent of n .

Note that

$$\begin{aligned} \log \left(\frac{G(h)^n}{\prod_{j=1}^{n-1} h_{n,j}} \right) &= \log \left(\frac{\exp(\frac{n}{2\pi} \int_0^{2\pi} \log(h(\omega)) d\omega)}{\prod_{j=1}^{n-1} h_{n,j}} \right) \\ &= \frac{n}{2\pi} \left(\int_0^{2\pi} \log(h(\omega)) d\omega - \frac{2\pi}{n} \sum_{j=1}^{n-1} \log(h_{n,j}) \right), \end{aligned}$$

which tends to a constant as $n \rightarrow \infty$ by Assumption 1 and the fact that for any function $g(x)$, the Riemann approximation for $\int_0^{2\pi} g(x) dx$ has an error of $O(1/n)$ when $\int_0^{2\pi} |g'(x)| dx < \infty$. Therefore,

$$\frac{G(h)^n}{\prod_{j=1}^{n-1} h_{n,j}}$$

tends to a positive constant as $n \rightarrow \infty$, and we conclude that

$$\begin{aligned} \det(M) &= \frac{\det(T_n[f])}{\det(\Lambda_{K_n * f})} \\ &\geq C_1 (2\pi)^n \frac{G(h)^n n^{d^2}}{n^{C_2 \log(n)} \prod_{j=1}^{n-1} h_{n,j}} \\ &= C_1 (2\pi)^n n^{d^2 - C_2 \log(n)} \frac{G(h)^n}{\prod_{j=1}^{n-1} h_{n,j}} \\ &\sim C_1 (2\pi)^n n^{d^2 - C_2 \log(n)}, \end{aligned}$$

where C_1 and C_2 are finite positive constants independent of n .

Now we use (12) to find an upper bound for the largest eigenvalue of M . We will need the following lemma, which is proved in the appendix.

LEMMA 4. If f satisfies Assumption 1,

$$n - C_1^{\frac{1}{n-1}} (n-1) n^{\frac{d^2 - 1 - C_2 \log(n)}{n-1}} = C_2 \log^2(n) + (1 - d^2) \log(n) + O(1)$$

for any positive C_1 and C_2 that are independent of n .

Thus, by (11), we have

$$\begin{aligned}
\lambda_{\max}(M) &\leq \operatorname{tr}(M) - (n-1) \left\{ \frac{\det(M)}{\operatorname{tr}(M)} \right\}^{\frac{1}{n-1}} \\
&= 2\pi n - (n-1) \left\{ \frac{\det(M)}{2\pi n} \right\}^{\frac{1}{n-1}} \\
&\leq 2\pi n - (n-1) \left\{ \frac{C_1(2\pi)^n n^{d^2 - C_2 \log(n)}}{2\pi n} \right\}^{\frac{1}{n-1}} \\
&= 2\pi \left\{ n - C_1^{\frac{1}{n-1}} (n-1) n^{\frac{d^2 - 1 - C_2 \log(n)}{n-1}} \right\} \\
&= O(\log^2(n)).
\end{aligned}$$

4.2. The smallest eigenvalue of M . We first rewrite the circulant preconditioned matrix as follows:

$$C_n^{-1}[K_n * f]T_n[f] = \{C_n[K_n * f^{-1}]C_n[K_n * f]\}^{-1} \{C_n[K_n * f^{-1}]T_n^{-1}[f^{-1}]\} \{T_n[f^{-1}]T_n[f]\}.$$

Since $\lambda_{\min}(AB) \geq \lambda_{\min}(A)\lambda_{\min}(B)$ for any positive semidefinite square matrices A, B , we have

$$\begin{aligned}
\lambda_{\min}(C_n^{-1}[K_n * f]T_n[f]) &\geq \lambda_{\min} \left\{ (C_n[K_n * f^{-1}]C_n[K_n * f])^{-1} \right\} \\
&\quad \times \lambda_{\min} \{C_n[K_n * f^{-1}]T_n^{-1}[f^{-1}]\} \\
&\quad \times \lambda_{\min} \{T_n[f^{-1}]T_n[f]\}.
\end{aligned}$$

As we already know, the eigenvalues of $C_n[K_n * f]$ are given by the expected value of the periodogram $E(I_{n,j}^f)$ generated by spectral density f . Similarly the eigenvalues of $C_n[K_n * f^{-1}]$ are given by the expected periodogram $E(I_{n,j}^{f^{-1}})$ generated by spectral density f^{-1} . Furthermore, the eigenvalues of $C_n[K_n * f^{-1}]C_n[K_n * f]$ are given by $E(I_{n,j}^{f^{-1}})E(I_{n,j}^f)$.

To obtain a lower bound for each term in the product of the above equation, we will need the following lemmas, which are proved in the appendix.

LEMMA 5. *With f satisfying Assumption 1,*

$$\lambda_{\min} \left\{ (C_n[K_n * f^{-1}]C_n[K_n * f])^{-1} \right\} \geq C,$$

where C represents a positive constant independent of n .

LEMMA 6. *With f satisfying Assumption 1,*

$$\lambda_{\min} \{C_n[K_n * f^{-1}]T_n^{-1}[f^{-1}]\} \geq C \log^{-2}(n),$$

where C represents a positive constant independent of n .

LEMMA 7. *With f satisfying Assumption 1,*

$$\lambda_{\min} \{T_n[f^{-1}]T_n[f]\} \geq C,$$

where C represents a positive constant independent of n .

In summary, we always have

$$\lambda_{\min}(C_n^{-1}[K_n * f]T_n[f]) \geq C \log^{-2}(n),$$

where C is a finite positive constant independent of n .

5. Bound for the complexity of the PCG algorithm for Toeplitz matrix with Fisher–Hartwig singularity. Now we have a bound for the condition number of the circulant preconditioned matrix using the upper bound for the largest eigenvalue and lower bound for the smallest eigenvalue. The condition number of $C_n^{-1}[K_n * f]T_n[f]$ satisfies

$$\begin{aligned} K(C_n^{-1}[K_n * f]T_n[f]) &= O\left(\frac{\log^2(n)}{C \log^{-2}(n)}\right) \\ &= O(\log^4(n)). \end{aligned}$$

Therefore, the number of iterations in the PCG algorithm is bounded by $\sqrt{K}$, i.e., $O(\log^2(n))$. Since each iteration needs $O(n \log(n))$ operations using the FFT, the total complexity of the PCG algorithm is $O(n \log^3(n))$. This result shows that the PCG algorithm is indeed more efficient than the CG algorithm of complexity $O(n^{1+|d|} \log(n))$. Note that CG itself is also more efficient than Levinson's (1946) algorithm, which has complexity $O(n^2)$.

6. Appendix: Proof of lemmas.

Proof of Lemma 2. For the determinant of a Toeplitz matrix, there is a great deal of literature generalizing the famous strong Szegő limit theorem (Grenander and Szegő (1984), Böttcher and Silbermann (1999)). We invoke Theorem 5.47 in Böttcher and Silbermann (1999, p. 177) with $\alpha_1 = -d$, $t_1 = 1$, $\beta_1 = 0$, $N = 1$, and $b = h$. Since our h is in L^∞ and B_1^1 , we have

$$\det\left(\frac{1}{2\pi}T_n[f]\right) \sim G(h)^n n^{d^2} E,$$

where

$$G(h) = \exp((\log(h))_0)$$

and E is a constant independent of n . The symbol $(\log(h))_0$ denotes the Fourier coefficient of $\log(h)$ at frequency 0, that is, $\frac{1}{2\pi} \int_0^{2\pi} \log(h(\omega)) d\omega$. Note that the $\frac{1}{2\pi}$ in the determinant formula comes from the fact that the theorem in Böttcher and Silbermann (1999) assumes the elements of $T_n[f]$ related to f by

$$t_k = \frac{1}{2\pi} \int_{-\pi}^{\pi} f(\omega) \exp(-ik\omega) d\omega, \quad k = 0, \dots, n-1.$$

However, as pointed out before, in our context, t_k in Toeplitz matrix $T_n[f]$ is the autocovariance at lag k and related to spectral density f by

$$t_k = \int_{-\pi}^{\pi} f(\omega) \exp(-ik\omega) d\omega, \quad k = 0, \dots, n-1.$$

Therefore, we conclude that

$$\det(T_n[f]) \sim (2\pi)^n \exp\left(\frac{n}{2\pi} \int_0^{2\pi} \log(h(\omega)) d\omega\right) n^{d^2} E.$$

Note that $\det(T_n[f])$ is always positive since matrix $T_n[f]$ is positive definite.

Proof of Lemma 3. From Bloomfield (2000), $\Lambda_{K_n * f}$ is a diagonal matrix with element $E(I_{n,0}), E(I_{n,1}), \dots, E(I_{n,n-1})$, where $I_{n,j}$ is the periodogram of $\{y_t\}_{t=1}^n$ at j th Fourier frequency. It follows from Deo and Hurvich (1998) that

$$E(I_{n,0}) = C_0 n^{2d},$$

where C_0 is a finite positive constant independent of n . From now on, C, C_0, C_1 all represent positive finite constants in general. We denote for $j \geq 1$

$$D_{n,j} = \frac{E(I_{n,j})}{f_{n,j}},$$

where $f_{n,j}$ is the spectral density of y_t at j th Fourier frequency. In our long-memory context (Assumption 1), we have

$$f_{n,j} = \left| 2 \sin \left(\frac{2\pi j}{2n} \right) \right|^{-2d} h_{n,j} \quad \text{for } j \geq 1.$$

Therefore,

$$\begin{aligned} |\det(\Lambda_{K_n * f})| &= \prod_{j=0}^{n-1} E(I_{n,j}) \\ &= E(I_n(0)) \prod_{j=1}^{n-1} D_{n,j} \prod_{j=1}^{n-1} f_{n,j} \\ &= C_0 n^{2d} \prod_{j=1}^{n-1} D_{n,j} \prod_{j=1}^{n-1} \left| 2 \sin \left(\frac{\pi j}{n} \right) \right|^{-2d} \prod_{j=1}^{n-1} h_{n,j}. \end{aligned}$$

From formulas 1021 and 1022 of Jolley (1961), we have

$$\prod_{j=1}^{n-1} \left| 2 \sin \left(\frac{\pi j}{n} \right) \right| = n.$$

Therefore

$$\begin{aligned} \det(\Lambda_{K_n * f}) &= C_0 n^{2d} \prod_{j=1}^{n-1} D_{n,j} \prod_{j=1}^{n-1} \left| 2 \sin \left(\frac{\pi j}{n} \right) \right|^{-2d} \prod_{j=1}^{n-1} h_{n,j} \\ &= C_0 \prod_{j=1}^{n-1} D_{n,j} \prod_{j=1}^{n-1} h_{n,j}. \end{aligned}$$

From Moulines and Soulier (1999), we have, for $1 \leq j \leq \frac{n}{2}$ and some positive constant C independent of j and n , that

$$D_{n,j} \leq 1 + C \frac{\log(j)}{j}.$$

This implies

$$\begin{aligned}
\prod_{j=1}^{n-1} D_{n,j} &= \exp \left(2 \log \left(\prod_{j=1}^{\frac{n}{2}} D_{n,j} \right) \right) = \exp \left(2 \sum_{j=1}^{\frac{n}{2}} \log(D_{n,j}) \right) \\
&\leq \exp \left(2 \sum_{j=1}^{\frac{n}{2}} \log \left(1 + C \frac{\log(j)}{j} \right) \right) \\
&\sim \exp \left(2 \sum_{j=1}^{\frac{n}{2}} C \frac{\log(j)}{j} \right) \\
&\sim \exp (C(\log(n-1))^2) \\
&= (n-1)^{C \log(n-1)} \\
&\sim n^{C \log(n)}.
\end{aligned}$$

Finally, we have

$$\det(\Lambda_{K_n * f}) = O \left(n^{C \log(n)} \prod_{j=1}^{n-1} h_{n,j} \right).$$

Proof of Lemma 4. First, we have

$$\begin{aligned}
n - C_1^{\frac{1}{n-1}} (n-1) n^{\frac{d^2-1-C_2 \log(n)}{n-1}} &= n - C_1^{\frac{1}{n-1}} (n-1) \exp \left(\frac{(d^2-1) \log(n) - C_2 \log^2(n)}{n-1} \right) \\
&= n \left\{ 1 - C_1^{\frac{1}{n-1}} \exp \left(\frac{(d^2-1) \log(n) - C_2 \log^2(n)}{n-1} \right) \right\} \\
&\quad + C_1^{\frac{1}{n-1}} \exp \left(\frac{(d^2-1) \log(n) - C_2 \log^2(n)}{n-1} \right).
\end{aligned}$$

Note that, as $n \rightarrow \infty$, we have $\frac{(d^2-1) \log(n) - C_2 \log^2(n)}{n-1} \rightarrow 0$ and $C_1^{\frac{1}{n-1}} \rightarrow 1$. Therefore

$$C_1^{\frac{1}{n-1}} \exp \left(\frac{(d^2-1) \log(n) - C_2 \log^2(n)}{n-1} \right) \rightarrow 1.$$

Since $\exp(x) = 1 + x + O(x^2)$ and $C_1^x = 1 + \log(C_1)x + O(x^2)$ when $x \rightarrow 0$, setting

$$E_n = n \left\{ 1 - C_1^{\frac{1}{n-1}} \exp \left(\frac{(d^2-1) \log(n) - C_2 \log^2(n)}{n-1} \right) \right\},$$

we also have

$$\begin{aligned}
E_n &= n \left\{ 1 - \left(1 + \frac{\log(C_1)}{n-1} + O(n^{-2}) \right) \exp \left(\frac{(d^2-1) \log(n) - C_2 \log^2(n)}{n-1} \right) \right\} \\
&= n \left\{ 1 - \left(1 + \frac{\log(C_1)}{n-1} + O(n^{-2}) \right) \left(1 + \frac{(d^2-1) \log(n) - C_2 \log^2(n)}{n-1} + O(\log^4(n)n^{-2}) \right) \right\} \\
&= C_2 \log^2(n) + (1-d^2) \log(n) + O(1).
\end{aligned}$$

Therefore,

$$n - C_1^{\frac{1}{n-1}} (n-1)n^{\frac{d^2-1-C_2 \log(n)}{n-1}} = C_2 \log^2(n) + (1-d^2) \log(n) + O(1).$$

Proof of Lemma 5. We will need the following existing result on the expected periodogram and spectral density of long-memory time series. We use the notation $f_{n,j} = f(\omega_j)$ and $I_{n,j} = I_n(\omega_j)$. It should also be noted that $I_{n,j} = I_{n,n-j}$, $j = 0, \dots, \frac{n}{2}$ and the memory parameter $d \in (-\frac{1}{2}, \frac{1}{2})$.

Under Assumption 1, it follows from Moulines and Soulier (1999), Lemma 2 that there exists finite positive constants C_1 and C_2 such that for all integers $j \in \{1, \dots, n/2\}$,

$$C_2 \leq \frac{E(I_{n,j})}{f_{n,j}} \leq C_1.$$

Therefore, $E(I_{n,j}) \sim f_{n,j}$ and for $j \in \{1, \dots, n/2\}$,

$$E(I_{n,j}^{f^{-1}})E(I_{n,j}^f) \sim f_{n,j}f_{n,j}^{-1} \sim C.$$

In addition, Deo and Hurvich (1998) have shown that $E(I_{n,0}) \sim C_0 n^{2d}$. Therefore, we have for any $j \in \{0, 1, \dots, n/2\}$

$$E(I_{n,j}^{f^{-1}})E(I_{n,j}^f) \sim C.$$

Now we prove $\lambda_{\min}([C_n[K_n * f^{-1}]C_n[K_n * f]]^{-1})$ is bounded below by some positive constant independent of n . Note that if f is with memory parameter d , then f^{-1} is with memory parameter $-d$.

$$\begin{aligned} \lambda_{\min} \left\{ (C_n[K_n * f^{-1}]C_n[K_n * f])^{-1} \right\} &= \frac{1}{\lambda_{\max} \{C_n[K_n * f^{-1}]C_n[K_n * f]\}} \\ &= \frac{1}{\max_j \left(E(I_{n,j}^{f^{-1}})E(I_{n,j}^f) \right)} \\ &= \frac{1}{\max_j \left(\frac{E(I_{n,j}^{f^{-1}})}{f_{n,j}^{-1}} \right) \left(\frac{E(I_{n,j}^f)}{f_{n,j}} \right)} \\ &\geq \frac{1}{C_1}. \end{aligned}$$

The last inequality used the result in Moulines and Soulier (1999).

Proof of Lemma 6. By the matrix similarity theory and properties of circulant matrices, matrix product $T_n[f^{-1}]C_n^{-1}[K_n * f^{-1}]$ has the same eigenvalue as matrix product $C_n^{-\frac{1}{2}}[K_n * f^{-1}]T_n^{-1}[f^{-1}]C_n^{-\frac{1}{2}}[K_n * f^{-1}]$, which is exactly the M matrix defined in (10), except the generating function is now f^{-1} . The upper bound for the largest eigenvalue of M obtained in section 4 still applies. That is,

$$\begin{aligned} \lambda_{\max}(T_n[f^{-1}]C_n^{-1}[K_n * f^{-1}]) &= \lambda_{\max}(M) \\ &\leq C \log^2(n). \end{aligned}$$

Therefore,

$$\begin{aligned}\lambda_{\min} (C_n[K_n * f^{-1}]T_n^{-1}[f^{-1}]) &= \frac{1}{\lambda_{\max} (T_n[f^{-1}]C_n^{-1}[K_n * f^{-1}])} \\ &\geq \frac{1}{C \log^2(n)} \\ &= \frac{1}{C} \log^{-2}(n).\end{aligned}$$

Proof of Lemma 7. Let

$$u_n(\omega) = f^{-\frac{1}{2}}(\omega) \left(1, e^{i\omega}, \dots, e^{i(n-1)\omega}\right)^*, \quad v_n(\omega) = f^{\frac{1}{2}}(\omega) \left(1, e^{i\omega}, \dots, e^{i(n-1)\omega}\right)^*.$$

Then

$$T_n(f) = \int_{-\pi}^{\pi} u_n(\omega) u_n^*(\omega) d\omega, \quad T_n(f^{-1}) = \int_{-\pi}^{\pi} v_n(\omega) v_n^*(\omega) d\omega.$$

Since $\int_{-\pi}^{\pi} u_n(\omega) v_n^*(\omega) d\omega = 2\pi I_n$,

$$\lambda_{\min} [T_n(f^{-1}) T_n(f)] - 4\pi^2 = \lambda_{\min} [T_n(f^{-1}) T_n(f) - 4\pi^2 I_n] \geq 0.$$

The inequality above is a result of the following generalization of the Cauchy–Schwarz inequality, proved in Chen, Hurvich, and Lu (2004): Let $\alpha(x)$ and $\beta(x)$ be two $n \times m$ matrices, $m \leq n$. Let

$$U = \int \alpha(x) \alpha^*(x) dx, \quad V = \int \beta(x) \beta^*(x) dx, \quad G = \int \alpha(x) \beta^*(x) dx.$$

If U is positive definite, then

$$\lambda_{\min}(UV - GG^*) \geq 0.$$

Acknowledgments. We are grateful for helpful discussions with Albrecht Böttcher, Willa Chen, Rohit Deo, Michael Ng, Stefano Serra, and Andy Yip. We especially appreciate the critical improvements proposed by Professor Stefano Serra and two anonymous referees.

REFERENCES

- G. S. AMMAR AND W. B. GRAGG (1988), *Superfast solution of real positive definite Toeplitz systems*, SIAM J. Matrix Anal. Appl., 9, pp. 61–76.
- O. AXELSSON AND V. BARKER (1984), *Finite Element Solution of Boundary Value Problems: Theory and Computation*, Academic Press, Orlando, FL; reprinted, Classics Appl. Math. 35, SIAM, Philadelphia, 2001.
- O. AXELSSON AND G. LINDSKOG (1986), *The rate of convergence of the preconditioned conjugate gradient method*, Numer. Math., 48, pp. 499–523.
- J. BERAN (1994), *Statistics for Long Memory Processes*, Chapman and Hall, New York.
- P. BLOOMFIELD (2000), *Fourier Analysis of Time Series: An Introduction*, John Wiley, New York.
- A. BÖTTCHER AND S. GRUDSKY (1998), *On the condition numbers of large semi-definite Toeplitz matrices*, Linear Algebra Appl., 279, pp. 285–301.
- A. BÖTTCHER AND B. SILBERMANN (1999), *Introduction to Large Truncated Toeplitz Matrices*, Springer-Verlag, New York.
- P. J. BROCKWELL AND R. A. DAVIS (1991), *Time Series: Theory and Methods*, Springer-Verlag, New York.

- R. H. CHAN (1989), *Circulant preconditioners for Hermitian Toeplitz systems*, SIAM J. Matrix Anal. Appl., 10, pp. 542–550.
- R. H. CHAN AND M. K. NG (1996), *Conjugate gradient methods for Toeplitz systems*, SIAM Rev., 38, pp. 427–482.
- R. H. CHAN AND G. STRANG (1989), *Toeplitz equations by conjugate gradients with circulant preconditioner*, SIAM J. Sci. Statist. Comput., 10, pp. 104–119.
- R. H. CHAN AND M. YEUNG (1992a), *Circulant preconditioners for Toeplitz matrices with positive continuous generating functions*, Math. Comp., 58, pp. 233–240.
- R. H. CHAN AND M.-C. YEUNG (1992b), *Circulant preconditioners constructed from Kernels*, SIAM J. Numer. Anal., 29, pp. 1093–1103.
- R. H. CHAN, A. M. YIP, AND M. NG (2000), *The best circulant preconditioners for Hermitian Toeplitz systems*, SIAM J. Numer. Anal., 38, pp. 876–896.
- T. F. CHAN (1988), *An optimal circulant preconditioner for Toeplitz systems*, SIAM J. Sci. Statist. Comput., 9, pp. 766–771.
- T. CHAN AND P. HANSEN (1992), *A look-ahead Levinson algorithm for general Toeplitz systems*, IEEE Trans. Signal Process., 40, pp. 1079–1090.
- W. CHEN, C. HURVICH, AND Y. LU (2004), *On the Correlation Matrix of the Discrete Fourier Transform and the Fast Solution of Large Toeplitz System for Long Memory Time Series*, Working Paper; available from <http://w4.stern.nyu.edu/ioms/docs/sg/workingpapers/pcg-CHL.pdf>, Stern School of Business, New York University.
- R. DEO AND C. HURVICH (1998), *Linear trend with fractionally integrated errors*, J. Time Series Anal., 19, pp. 379–397.
- R. DEO, C. HURVICH, AND Y. LU (2005), *Forecasting realized volatility using a long memory stochastic volatility model: Estimation, prediction, and seasonal adjustment*, J. Econometrics, to appear.
- M. E. FISHER AND R. E. HARTWIG (1968), *Toeplitz determinants—some applications, theorems, and conjectures*, Adv. Chem. Phys., 32, pp. 190–225.
- G. H. GOLUB AND C. F. VAN LOAN (1996), *Matrix Computations*, 3rd ed., Johns Hopkins University Press, Baltimore, MD, 1996.
- U. GRENANDER AND G. SZEGÖ (1984), *Toeplitz Forms and Their Applications*, 2nd ed., Chelsea, New York.
- A. HARVEY (1998), *Long Memory in Stochastic Volatility*, in Forecasting Volatility in Financial Markets, J. Knight and S. Satchell, eds., Butterworth-Heinemann, Oxford, pp. 307–320.
- R. A. HORN AND C. R. JOHNSON (1985), *Matrix Analysis*, Cambridge University Press, Cambridge, UK.
- C. HURVICH, R. DEO, AND J. BRODSKY (1998), *The mean squared error of Geweke and Porter-Hudak's estimator of the memory parameter of a long-memory time series*, J. Time Series Anal., 19, pp. 19–46.
- L. B. W. JOLLEY (1961), *Summation of Series*, Dover, New York.
- L. H. KOOPMANS (1974), *The Spectral Analysis of Time Series*, Academic Press, New York.
- N. LEVINSON (1946), *The Wiener RMS (root mean square) error criterion in filter design and prediction*, J. Math. Phys., 25, pp. 261–178.
- J. K. MERIKOSKI AND A. VIRTANEN (1997), *Bounds for eigenvalues using the trace and determinant*, Linear Algebra Appl., 264, pp. 101–108.
- J. K. MERIKOSKI, U. URPALA, A. VIRTANEN, T. Y. TAM, AND F. UHLIG (1997), *A best upper bound for the 2-norm condition number of a matrix*, Linear Algebra Appl., 254, pp. 355–365.
- E. MOULINES AND P. SOULIER (1999), *Broadband log-periodogram regression of time series with long-range dependence*, Ann. Statist., 27, pp. 1415–1439.
- P. M. ROBINSON (1994), *Time series with strong dependence*, in Proceedings of the Sixth World Congress, Advances in Econometrics, Vol. 1, Cambridge University Press, Cambridge, UK, pp. 47–95.
- S. SERRA (1999), *A Korovkin-type theory for finite Toeplitz operators via matrix algebras*, Numer. Math., 82, pp. 117–142.
- S. SERRA (1998a), *On the extreme eigenvalues of Hermitian (block) Toeplitz matrices*, Linear Algebra Appl., 270, pp. 109–129.
- S. SERRA (1998b), *An ergodic theorem for classes of preconditioned matrices*, Linear Algebra Appl., 282, pp. 161–183.
- S. SERRA (1995), *New PCG based algorithms for the solution of Hermitian Toeplitz systems*, Calcolo, 32, pp. 153–176.
- G. STRANG (1986), *A proposal for Toeplitz matrix calculations*, Stud. Appl. Math., 74, pp. 171–176.
- H. WIDOM (1973), *Toeplitz determinants with singular generating functions*, Amer. J. Math., 95, pp. 333–383.

RELATIONS AMONG THE MINIMUM NORM COEFFICIENTS FOR DEGENERATE NONSTATIONARY FLOWS*

MASAYA MATSUURA[†]

Abstract. In previous papers, we have investigated degenerate stochastic processes (or more generally, degenerate flows in inner product spaces), which is important in the analysis of time series obtained from deterministic dynamical systems. In this paper, we shall generalize the results in [M. Matsuura, *Methodol. Comput. Appl. Probab.*, 5 (2003), pp. 369–387] and derive recursive relations for the minimum norm coefficients of the equations which describe the time evolutions of degenerate nonstationary flows. The obtained results can be considered as recursive relations among the minimum norm least squares solutions of matrix equations.

Key words. degenerate flow, minimum norm least squares solution, recursive relation, generalized inverse, fluctuation-dissipation theorem

AMS subject classifications. 15A24, 15A09, 60G07

DOI. 10.1137/S0895479803432964

1. Introduction. In this paper, we derive recursive relations of the minimum norm coefficients of the equations that describe the time evolution of degenerate nonstationary flows in inner product spaces.

Let $(W, (\star, \star))$ be any real inner product space with an inner product $(\star, \star)$. By a d -dimensional flow $\mathbf{Z} = (Z(n); l \leq n \leq r)$ in W , we mean a function $Z : \{l, l+1, \dots, r-1, r\} \rightarrow W^d$, where l, r , and d ($l \leq r$, $1 \leq d$) are integers. The most important example of a flow is a square-integrable stochastic process. As another example, we shall consider in section 4, flows constructed from the row vectors of a matrix. A d -dimensional flow $\mathbf{Z} = (Z(n); l \leq n \leq r)$ is said to be nondegenerate if $\{Z_j(n); 1 \leq j \leq d, l \leq n \leq r\}$ is linearly independent in W . Otherwise, $\mathbf{Z}$ is said to be degenerate.

Okabe [15] has investigated the so-called KM_2O -Langevin equation that governs the time evolution of general flows and proved fluctuation-dissipation theorem for nondegenerate stationary flows by applying the fundamental idea of the fluctuation-dissipation principle in nonequilibrium statistical physics. In [12, 13], we have treated flows without nondegeneracy property and stationarity property. In degenerate cases, the coefficients of the equation are not uniquely determined. We proved in [12] that the coefficients with minimum norm property can be obtained as the limits of some functions. We also obtained recursive relations among the minimum norm coefficients in the stationary case [11]; in this paper, we generalize it to the nonstationary case. Thanks to this result, in the analysis of a d -dimensional degenerate flow $\mathbf{Z} = (Z(n); 0 \leq n \leq N)$, the problem of finding the minimum norm solution of nd -dimensional equations ($1 \leq n \leq N+1$) can be reduced to that of d -dimensional equations (Algorithm 3.1).

*Received by the editors August 6, 2003; accepted for publication (in revised form) by H. J. Werner May 18, 2005; published electronically December 7, 2005. Partially supported by Grand-in-Aid for Young Scientists 15740053, for Scientific Research 14340030 and Superrobust Computation Project of the 21st Century COE Program “Information Science and Technology Strategic Core” of the Ministry of Education, Culture, Sports, Science, and Technology, Japan.

<http://www.siam.org/journals/simax/27-3/43296.html>

[†]Department of Mathematical Informatics, Graduate School of Information Science and Technology, University of Tokyo, Bunkyo-ku, Tokyo 113-8656, Japan (masaya@mist.i.u-tokyo.ac.jp).

Lev-Ari and Kailath [9] have investigated nondegenerate nonstationary processes and obtained the so-called generalized Levinson algorithm together with the generalized Shur algorithm. On the other hand, Inouye [6, 7] and Dégerine [1] have proposed Levinson-type algorithms for degenerate stationary processes. In the context of these papers, the recursive relation obtained in this paper can be seen as a further generalization of these algorithms to the degenerate nonstationary case. Some applications in the stationary case are presented in [2, 3]. In general, it is not easy to deal with nonstationary time series. However, we expect that our method will work effectively in the analysis of time series derived from physical systems, if we can conduct repeated measurements under the same environment. In this case, we can obtain a sufficiently large number of sample passes. Therefore, we can estimate the covariance function by taking the sample means, even if the process is nonstationary. In addition, if efficient estimation procedures of covariance functions are developed for some important classes of nonstationary processes, then our algorithm will be feasible in more general settings.

The most remarkable feature of our algorithm is that it derives minimum norm coefficients, which is not the case in [1, 6, 7]. Concerning observation noise, minimum norm coefficients have some advantages over other types of coefficients. We shall see the detail at the end of section 2.

We shall briefly state the content of this paper. In section 2 we shall show some fundamental facts on flows which are needed in the later discussion. Section 3 is devoted to the proofs of the main results. In section 4, we shall study one-dimensional cases and apply the results to minimum norm least squares solutions of matrix equations.

At the end of this section, we explain some definitions. We denote by I , the identity matrix of any size. For a given matrix A , we denote its transpose matrix, trace and rank by ${}^tA$, $\text{tr } A$, and $\text{rank } A$, respectively. Furthermore, the Moore–Penrose inverse and the Euclidean norm of the matrix A are denoted by A^+ and $\|A\|_E$, respectively. As for the sum of vectors, we define $\sum_{i=n}^m * \equiv 0$ if $n > m$, where $*$'s are any vectors.

2. Auxiliary results on flows in inner product spaces. In this section, we shall show some statements which are needed in the later discussion. Let $(W, (\star, \star))$ be any real inner product space with an inner product $(\star, \star)$. By a d -dimensional flow $\mathbf{Z} = (Z(n); l \leq n \leq r)$ in W , we mean a function $Z : \{l, l+1, \dots, r-1, r\} \rightarrow W^d$, where l, r , and d ($l \leq r$, $1 \leq d$) are integers. For example, given a probability space $(\Omega, \mathcal{F}, P)$ and a square-integrable d -dimensional stochastic process $\mathbf{S} = (S(n); l \leq n \leq r)$ on it, we can interpret $\mathbf{S}$ as a d -dimensional flow in $L^2(\Omega, \mathcal{F}, P)$ with the inner product $(X, Y) \equiv E(XY)$. A d -dimensional flow $\mathbf{Z} = (Z(n); l \leq n \leq r)$ is said to be nondegenerate if $\{Z_j(n); 1 \leq j \leq d, l \leq n \leq r\}$ is linearly independent in W , where $Z_j(n)$ is the j th component of $Z(n)$. Otherwise, $\mathbf{Z}$ is said to be degenerate. For two integers n_1 and n_2 ($l \leq n_1 \leq n_2 \leq r$), we denote by $\mathbf{M}_{n_1}^{n_2}(\mathbf{Z})$ the subspace of W spanned by $\{Z_j(m); 1 \leq j \leq d, n_1 \leq m \leq n_2\}$.

Given two flows $\mathbf{X}_+ = (X_+(n); 0 \leq n \leq N_+)$ and $\mathbf{X}_- = (X_-(l); -N_- \leq l \leq 0)$ in W , we derive a new d -dimensional flow $\nu_+(\mathbf{X}_+) = (\nu_+(\mathbf{X}_+)(n); 0 \leq n \leq N_+)$ (resp., $\nu_-(\mathbf{X}_-) = (\nu_-(\mathbf{X}_-)(l); -N_- \leq l \leq 0)$) by projecting each component of $X_+(n)$ (resp., $X_-(l)$) onto the subspace $(\mathbf{M}_0^{n-1}(\mathbf{X}_+))^\perp$ (resp., $(\mathbf{M}_{l+1}^0(\mathbf{X}_-))^\perp$), that is,

$$(2.1) \quad \begin{aligned} \nu_+(\mathbf{X}_+)(0) &\equiv X_+(0), \\ \nu_+(\mathbf{X}_+)(n) &\equiv X_+(n) - P_{\mathbf{M}_0^{n-1}(\mathbf{X}_+)} X_+(n) \quad (1 \leq n \leq N_+), \end{aligned}$$

$$(2.2) \quad \begin{aligned} \nu_-(\mathbf{X}_-)(0) &\equiv X_-(0), \\ \nu_-(\mathbf{X}_-)(l) &\equiv X_-(l) - P_{\mathbf{M}_{l+1}^0(\mathbf{X}_-)} X_-(l) \quad (-N_- \leq l \leq -1). \end{aligned}$$

Furthermore, we define two matrix functions $V_\pm(\mathbf{X}_\pm) = (V_\pm(\mathbf{X}_\pm)(n); 0 \leq n \leq N_\pm)$ by

$$(2.3) \quad V_\pm(\mathbf{X})(n) \equiv (\nu_\pm(\mathbf{X}_\pm)(\pm n), {}^t\nu_\pm(\mathbf{X}_\pm)(\pm n)) \quad (0 \leq n \leq N_\pm),$$

where $(\star, {}^t\star)$ denotes the inner product matrix of the vectors $\star$ and $\star$.

For the flows $\mathbf{X}_\pm$, there exist matrix functions $\gamma_\pm(\mathbf{X}_\pm) = (\gamma_\pm(\mathbf{X}_\pm)(n, k); 0 \leq k < n \leq N_\pm)$ such that

$$(2.4) \quad X_\pm(\pm n) = - \sum_{k=0}^{n-1} \gamma_\pm(\mathbf{X}_\pm)(n, k) X_\pm(\pm k) + \nu_\pm(\mathbf{X}_\pm)(\pm n) \quad (1 \leq n \leq N_\pm).$$

We call the above equations KM₂O-Langevin equations associated with the flows $\mathbf{X}_\pm$. In general, these matrix functions are not uniquely determined. We denote by $\mathcal{LMD}_\pm(\mathbf{X}_\pm)$ the sets of all matrix functions $\gamma_\pm(\mathbf{X}_\pm)$ for which (2.4) holds. We note that if the flow $\mathbf{X}_+$ (resp., $\mathbf{X}_-$) is nondegenerate, the matrix function $\gamma_+(\mathbf{X}_+)$ (resp., $\gamma_-(\mathbf{X}_-)$) is uniquely determined only through relation (2.4).

To analyze the detail of the sets $\mathcal{LMD}_\pm(\mathbf{X}_\pm)$ we define a norm $\|\star\|$ on these sets by

$$(2.5) \quad \|\gamma_\pm(\mathbf{X}_\pm)\| \equiv \left(\sum_{n=1}^{N_\pm} \sum_{k=0}^{n-1} \|\gamma_\pm(\mathbf{X}_\pm)(n, k)\|_{\mathbb{E}}^2 \right)^{\frac{1}{2}}.$$

Then there exist unique elements $\gamma_\pm^0(\mathbf{X}_\pm) \in \mathcal{LMD}_\pm(\mathbf{X}_\pm)$ that satisfy

$$\|\gamma_\pm^0(\mathbf{X}_\pm)\| \leq \|\gamma_\pm(\mathbf{X}_\pm)\| \quad (\gamma_\pm(\mathbf{X}_\pm) \in \mathcal{LMD}_\pm(\mathbf{X}_\pm)),$$

where equalities hold only for $\gamma_\pm(\mathbf{X}_\pm) = \gamma_\pm^0(\mathbf{X}_\pm)$. We note that to minimize the norm $\|\gamma_\pm(\mathbf{X}_\pm)\|$, we have only to minimize $\sum_{k=0}^{n-1} \|\gamma_\pm(\mathbf{X}_\pm)(n, k)\|_{\mathbb{E}}^2$ for each n ($1 \leq n \leq N_\pm$) separately. We call the matrix function $\gamma_+^0(\mathbf{X}) = (\gamma_+^0(\mathbf{X}_+)(n, k); 0 \leq k < n \leq N_+)$ (resp., $\gamma_-^0(\mathbf{X}) = (\gamma_-^0(\mathbf{X}_-)(n, k); 0 \leq k < n \leq N_-)$) the minimum forward (resp., backward) KM₂O-Langevin dissipation matrix function associated with the flow $\mathbf{X}_+$ (resp., $\mathbf{X}_-$). In particular, we define matrix functions $\delta_+^0(\mathbf{X}_+) = (\delta_+^0(\mathbf{X}_+)(n); 1 \leq n \leq N_+)$ and $\delta_-^0(\mathbf{X}_-) = (\delta_-^0(\mathbf{X}_-)(n); 1 \leq n \leq N_-)$ by

$$(2.6) \quad \delta_\pm^0(\mathbf{X}_\pm)(n) \equiv \gamma_\pm^0(\mathbf{X}_\pm)(n, 0).$$

Moreover, we define a system $\mathcal{LM}_\pm(\mathbf{X}_\pm)$ of $d \times d$ matrices by

$$(2.7) \quad \begin{aligned} \mathcal{LM}_\pm(\mathbf{X}_\pm) &\equiv \{\gamma_\pm^0(\mathbf{X}_\pm)(n, k), \delta_\pm^0(\mathbf{X}_\pm)(n), V_\pm(\mathbf{X}_\pm)(m); \\ &\quad 0 \leq k < n \leq N_\pm, 0 \leq m \leq N_\pm\}. \end{aligned}$$

Given a single flow $\mathbf{X} = (X(n); 0 \leq n \leq N)$ in W , we define new flows $\mathbf{X}_+^{(s)} = (X_+^{(s)}(n); 0 \leq n \leq N-s)$ and $\mathbf{X}_-^{(s)} = (X_-^{(s)}(l); -s \leq l \leq 0)$ for each s ($0 \leq s \leq N$) as follows:

$$(2.8) \quad X_+^{(s)}(n) \equiv X(n+s) \quad (0 \leq n \leq N-s), \quad X_-^{(s)}(l) \equiv X(l+s) \quad (-s \leq l \leq 0).$$

Then we have the following theorem. In nondegenerate cases, the statement here gives the essentially same recursive relations as in the generalized Levinson algorithm in [9].

THEOREM 2.1. *Let $\mathbf{X} = (X(n); 0 \leq n \leq N)$ be any flow in W . Then the following relations hold for all admissible values of s, n , and k :*

- (i) $\gamma_{\pm}^0(\mathbf{X}_{\pm}^{(s)})(n, k) = \gamma_{\pm}^0(\mathbf{X}_{\pm}^{(s \pm 1)})(n-1, k-1) + \delta_{\pm}^0(\mathbf{X}_{\pm}^{(s)})(n) \gamma_{\mp}^0(\mathbf{X}_{\mp}^{(s \pm n \mp 1)})(n-1, n-k-1),$
- (ii) $V_{\pm}(\mathbf{X}_{\pm}^{(s)})(n) = \{I - \delta_{\pm}^0(\mathbf{X}_{\pm}^{(s)})(n) \delta_{\mp}^0(\mathbf{X}_{\mp}^{(s \pm n)})(n)\} V_{\pm}(\mathbf{X}_{\pm}^{(s \pm 1)})(n-1),$
- (iii) $\delta_{+}^0(\mathbf{X}_{+}^{(s)})(n) V_{-}(\mathbf{X}_{-}^{(s+n-1)})(n-1) = V_{+}(\mathbf{X}_{+}^{(s+1)})(n-1) {}^t \delta_{-}^0(\mathbf{X}_{-}^{(s+n)})(n),$
- (iv) $V_{+}(\mathbf{X}_{+}^{(s)})(0) = V_{-}(\mathbf{X}_{-}^{(s)})(0),$
- (v) $\nu_{\pm}(\mathbf{X}_{\pm}^{(s)})(n) = \nu_{\pm}(\mathbf{X}_{\pm}^{(s \pm 1)})(n-1) + \delta_{\pm}^0(\mathbf{X}_{\pm}^{(s)})(n) \nu_{\mp}(\mathbf{X}_{\mp}^{(s \pm n \mp 1)})(n-1).$

Proof. We have only to prove (i) since the rest immediately follows from Theorems 4.2 and 4.4 in [13] and Corollary 2 in [8]. Assume that $\mathbf{X}$ is nondegenerate. Then, $\gamma_{\pm}^0(\mathbf{X}_{\pm}^{(s)})$ are the only elements of $\mathcal{LMD}_{\pm}(\mathbf{X}_{\pm}^{(s)})$. Thus Theorem 4.1 in [13] gives (i). To investigate the case where $\mathbf{X}$ is degenerate, we define for each real number $w > 0$, a new flow $\mathbf{X}^w = (X^w(n); 0 \leq n \leq N)$ by $X^w(n) = X(n) + w \xi(n)$, where $\xi = (\xi(n); 0 \leq n \leq N)$ is a flow in W such that $(X(m), {}^t \xi(n)) = 0$ and $(\xi(m), \xi(n)) = \delta_{mn} I$ ($0 \leq m, n \leq N$). Then from Lemma 4.2 and Theorem 4.5 in [12], we know that $(\mathbf{X}^w)_{\pm}^{(s)}$ are nondegenerate and $\lim_{w \rightarrow 0} \gamma_{\pm}((\mathbf{X}^w)_{\pm}^{(s)}) = \gamma_{\pm}^0(\mathbf{X}_{\pm}^{(s)})$. Thus the proof for degenerate flows is reduced to that for nondegenerate flows. $\square$

Let $R = (R(\mathbf{X})(m, n); 0 \leq m, n \leq N)$ be the covariance function of $\mathbf{X}$, that is, $R(\mathbf{X})(m, n) = (X(m), {}^t X(n))$. Then we can prove the following lemmas in the same way as in Theorems 1 and 2 in [8].

LEMMA 2.1. *For any choice of $\gamma_{\pm}(\mathbf{X}_{\pm}^{(s)}) \in \mathcal{LMD}_{\pm}(\mathbf{X}_{\pm}^{(s)})$, the following relations hold for all admissible values of s and n including $n = 1$ (recall that $\sum_{k=0}^{-1} * \equiv 0$):*

$$\begin{aligned} & \delta_{\pm}(\mathbf{X}_{\pm}^{(s)})(n) V_{\mp}(\mathbf{X}_{\mp}^{(s \pm n \mp 1)})(n-1) \\ &= - \left\{ R(s \pm n, s) + \sum_{k=0}^{n-2} \gamma_{\pm}^0(\mathbf{X}_{\pm}^{(s \pm 1)})(n-1, k) R(s \pm k \pm 1, s) \right\}, \end{aligned}$$

where $\delta_{\pm}(\mathbf{X}_{\pm}^{(s)})(n) \equiv \gamma_{\pm}(\mathbf{X}_{\pm}^{(s)})(n, 0)$.

LEMMA 2.2. *Let $\delta_{+}(s, n)$ ($0 \leq s \leq N-1, 1 \leq n \leq N-s$) and $\delta_{-}(s, n)$ ($1 \leq s \leq N, 1 \leq n \leq s$) be any matrices that satisfy*

$$\begin{aligned} (2.9) \quad & \delta_{\pm}(s, n) V_{\mp}(\mathbf{X}_{\mp}^{(s \pm n \mp 1)})(n-1) \\ &= - \left\{ R(s \pm n, s) + \sum_{k=0}^{n-2} \gamma_{\pm}^0(\mathbf{X}_{\pm}^{(s \pm 1)})(n-1, k) R(s \pm k \pm 1, s) \right\}. \end{aligned}$$

We define matrix functions $\gamma_{+}(s) = (\gamma_{+}(s, n, k); 0 \leq k < n \leq N-s)$ ($0 \leq s \leq N-1$) and $\gamma_{-}(s) = (\gamma_{-}(s, n, k); 0 \leq k < n \leq s)$ ($1 \leq s \leq N$) as follows:

$$(2.10) \quad \gamma_{\pm}(s, n, 0) \equiv \delta_{\pm}(s, n),$$

$$(2.11) \quad \begin{aligned} \gamma_{\pm}(s, n, k) &\equiv \gamma_{\pm}^0(\mathbf{X}_{\pm}^{(s \pm 1)})(n-1, k-1) \\ &+ \delta_{\pm}(s, n) \gamma_{\mp}^0(\mathbf{X}_{\mp}^{(s \pm n \mp 1)})(n-1, n-k-1) \quad (1 \leq k \leq n). \end{aligned}$$

Then $\gamma_{\pm}(s) \in \mathcal{LMD}(\mathbf{X}_{\pm}^{(s)})$ for all admissible values of s .

At the end of this section, we shall note that the minimum norm element $\gamma_+^0(\mathbf{X})$ has the following advantage over other elements of $\mathcal{LMD}_+(\mathbf{X})$. From the viewpoint of time series analysis, the flow ξ used in the proof of Theorem 2.1 can be interpreted as an observation noise process. Since $\lim_{w \rightarrow 0} \gamma_+(\mathbf{X}^w) = \gamma_+^0(\mathbf{X})$, the minimum norm elements are convenient to examine the relation between theoretical degenerate equations and the nondegenerate models derived from time series with observation noise. There is another reason why we focus our attention on $\gamma_+^0(\mathbf{X})$. If the norm $\|\gamma_+(\mathbf{X})\|$ is large, then the amount $\sum_{k=0}^{n-1} \gamma_+(\mathbf{X})(n, k)X(k)$ is very sensitive to the noise added to $X(k)$'s. Therefore, it is important to choose an element of $\mathcal{LMD}_+(\mathbf{X})$ with small norm.

3. Recursive relations between minimum norm coefficients. In the rest of this paper, we use the notation $\gamma_{\pm}^0(s)$, $\gamma_{\pm}^0(s, n, k)$, $\delta_{\pm}^0(s, n)$, and $V_{\pm}(s, n)$ instead of $\gamma_{\pm}^0(\mathbf{X}_{\pm}^{(s)})$, $\gamma_{\pm}^0(\mathbf{X}_{\pm}^{(s)})(n, k)$, $\delta_{\pm}^0(\mathbf{X}_{\pm}^{(s)})(n)$, and $V_{\pm}(\mathbf{X}_{\pm}^{(s)})(n)$, if there is no fear of confusion.

We now show the following theorems.

THEOREM 3.1. *Let $\mathbf{X} = (X(n); 0 \leq n \leq N)$ be any flow in W . Furthermore, let $\gamma_+(s) = (\gamma_+(s, n, k); 0 \leq k < n \leq N-s)$ (resp., $\gamma_-(s) = (\gamma_-(s, n, k); 0 \leq k < n \leq s)$) be any element of $\mathcal{LMD}_+(\mathbf{X}_+^{(s)})$ (resp., $\mathcal{LMD}_-(\mathbf{X}_-^{(s)})$). Then, the following conditions are equivalent with each other:*

- (i) $\gamma_{\pm}(s) = \gamma_{\pm}^0(s)$ for all admissible values of s , and
- (ii) $\gamma_{\pm}(s)$ satisfy

$$(3.1) \quad \gamma_{\pm}(s, n, k) = \gamma_{\pm}(s \pm 1, n-1, k-1) + \delta_{\pm}(s, n) \gamma_{\mp}(s \pm n \mp 1, n-1, n-k-1),$$

$$(3.2) \quad \begin{aligned} &\left\{ \sum_{k=0}^{n-1} \gamma_{\pm}(s, n, k)^t \gamma_{\mp}(s \pm n \mp 1, n-1, n-k-1) \right\} \\ &\times \{I - V_{\mp}(s \pm n \mp 1, n-1) V_{\mp}(s \pm n \mp 1, n-1)^+\} = 0, \end{aligned}$$

for all admissible values of s, n , and k , where $\delta_{\pm}(s, n) \equiv \gamma_{\pm}(s, n, 0)$ and $\gamma_{\pm}(s, n, n) \equiv I$.

THEOREM 3.2. *Let $\mathbf{X} = (X(n); 0 \leq n \leq N)$ be any flow in W , and $R = (R(m, n); 0 \leq m, n \leq N)$ be the covariance function of $\mathbf{X}$. Then, for all admissible values of s and n ,*

$$(3.3) \quad \begin{aligned} \delta_{\pm}^{\min}(s, n) &= \delta_{\pm}^0(s, n) + \left\{ \sum_{k=1}^{n-1} \gamma_{\pm}^0(s, n, k)^t \gamma_{\mp}^0(s \pm n \mp 1, n-1, n-k-1) \right\} \\ &\times \{I - V_{\mp}(s \pm n \mp 1, n-1) V_{\mp}(s \pm n \mp 1, n-1)^+\}, \end{aligned}$$

$$(3.4) \quad \begin{aligned} \delta_{\pm}^0(s, n) &= \{\delta_{\pm}^{\min}(s, n) - B_{\pm}(s \pm 1, n-1)(I - V_{\mp}(s \pm n \mp 1, n-1) V_{\mp}(s \pm n \mp 1, n-1)^+)\} \\ &\times \{I + A_{\mp}(s \pm n \mp 1, n-1)(I - V_{\mp}(s \pm n \mp 1, n-1) V_{\mp}(s \pm n \mp 1, n-1)^+)\}^{-1}, \end{aligned}$$

where $\delta_{\pm}^{\min}(s, n)$, $A_{\pm}(s, n)$ and $B_{\pm}(s, n)$ are defined by

$$(3.5) \quad \delta_{\pm}^{\min}(s, n) \equiv - \left\{ R(s \pm n, s) + \sum_{k=0}^{n-2} \gamma_{\pm}^0(s \pm 1, n-1, k) R(s \pm k \pm 1, s) \right\} V_{\mp}(s \pm n \mp 1, n-1)^+,$$

$$(3.6) \quad A_{\pm}(s, n) \equiv \sum_{k=0}^{n-1} \gamma_{\pm}^0(s, n, k) {}^t \gamma_{\pm}^0(s, n, k),$$

$$(3.7) \quad B_{\pm}(s, n) \equiv \sum_{k=0}^{n-1} \gamma_{\pm}^0(s, n, k) {}^t \gamma_{\mp}^0(s \pm n \mp 1, n, n-k-1).$$

Proof of Theorems 3.1 and 3.2. Let $\gamma_+(s) = (\gamma_+(s, n, k); 0 \leq k < n \leq N-s)$ (resp., $\gamma_-(s) = (\gamma_-(s, n, k); 0 \leq k < n \leq s)$) be any element of $\mathcal{LMD}_+(\mathbf{X}_+^{(s)})$ (resp., $\mathcal{LMD}_-(\mathbf{X}_-^{(s)})$) such that

$$(3.8) \quad \gamma_{\pm}(s, n, k) = \gamma_{\pm}^0(s \pm 1, n-1, k-1) + \delta_{\pm}(s, n) \gamma_{\mp}^0(s \pm n \mp 1, n-1, n-k-1) \quad (1 \leq k < n).$$

Among these $\gamma_{\pm}(s)$, the functions $\gamma_{\pm}^0(s)$ minimize the values $\sum_{k=0}^{n-1} \|\gamma_{\pm}(s, n, k)\|_{\mathbb{E}}^2$. We first obtain from Lemmas 2.1 and 2.2 that

$$(3.9) \quad \begin{aligned} \delta_{\pm}^0(s, 1) &= -R(s \pm 1, s) V_{\mp}(s, 0)^+ \\ &= \delta_{\pm}^0(s, 1) V_{\mp}(s, 0) V_{\mp}(s, 0)^+, \end{aligned}$$

which means $\gamma_{\pm}^0(s)$ satisfy (3.2) for $n=1$. Let n be any fixed integer such that $n \geq 2$. We now calculate $\sum_{k=0}^{n-1} \|\gamma_{\pm}(s, n, k)\|_{\mathbb{E}}^2$. From (3.8)

$$(3.10) \quad \begin{aligned} & \sum_{k=0}^{n-1} \|\gamma_{\pm}(s, n, k)\|_{\mathbb{E}}^2 \\ &= \text{tr} \left(\delta_{\pm}(s, n) \left(I + \sum_{k=0}^{n-2} \gamma_{\mp}^0(s \pm n \mp 1, n-1, k) {}^t \gamma_{\mp}^0(s \pm n \mp 1, n-1, k) \right) {}^t \delta_{\pm}(s, n) \right) \\ &+ \text{tr} \left(\left(2 \sum_{k=0}^{n-2} \gamma_{\pm}^0(s \pm 1, n-1, k) {}^t \gamma_{\mp}^0(s \pm n \mp 1, n-1, n-k-2) \right) {}^t \delta_{\pm}(s, n) \right) \\ &+ \sum_{k=0}^{n-2} \text{tr}(\gamma_{\pm}^0(s \pm 1, n-1, k) {}^t \gamma_{\pm}^0(s \pm 1, n-1, k)). \end{aligned}$$

Here we note from Lemma 2.2 that for any $\delta_{\pm}(s, n)$ such that

$$(3.11) \quad \begin{aligned} & \delta_{\pm}(s, n) V_{\mp}(s \pm n \mp 1, n-1) \\ &= - \left\{ R(s \pm n, s) + \sum_{k=0}^{n-2} \gamma_{\pm}^0(s \pm 1, n-1, k) R(s \pm k \pm 1, s) \right\}, \end{aligned}$$

$\gamma_{\pm}(s)$ obtained by (3.8) become elements of $\mathcal{LMD}_{\pm}(\mathbf{X}_{\pm}^{(s)})$. Thus, putting

$$\begin{aligned} X &\equiv \delta_{\pm}(s, n), & A &\equiv A_{\mp}(s \pm n \mp 1, n-1), \\ B &\equiv B_{\pm}(s \pm 1, n-1), & C &\equiv V_{\mp}(s \pm n \mp 1, n-1), \\ D &\equiv -\left\{ R(s \pm n, s) + \sum_{k=0}^{n-2} \gamma_{\pm}^0(s \pm 1, n-1, k) R(s \pm k \pm 1, s) \right\} \end{aligned}$$

in Lemma 3.2 in [11], we have (3.4), which can be rewritten as

$$\begin{aligned} \delta_{\pm}^{\min}(s, n) &= \delta_{\pm}^0(s, n) + \{\delta_{\pm}^0(s, n) A_{\mp}(s \pm n \mp 1, n-1) + B_{\pm}(s \pm 1, n-1)\} \\ (3.12) \quad &\times \{I - V_{\mp}(s \pm n \mp 1, n-1) V_{\mp}(s \pm n \mp 1, n-1)^+\}. \end{aligned}$$

Furthermore, Lemma 3.3 in [11] gives

$$\begin{aligned} \{\delta_{\pm}^0(s, n) + \delta_{\pm}^0(s, n) A_{\mp}(s \pm n \mp 1, n-1) + B_{\pm}(s \pm 1, n-1)\} \\ (3.13) \quad \times \{I - V_{\mp}(s \pm n \mp 1, n-1) V_{\mp}(s \pm n \mp 1, n-1)^+\} = 0. \end{aligned}$$

In the above relations, we know from Theorem 2.1 (i) that

$$\begin{aligned} &\delta_{\pm}^0(s, n) A_{\mp}(s \pm n \mp 1, n-1) + B_{\pm}(s \pm 1, n-1) \\ &= \sum_{k=0}^{n-2} \{\gamma_{\pm}^0(s \pm 1, n-1, k) \\ &\quad + \delta_{\pm}^0(s, n) \gamma_{\mp}^0(s \pm n \mp 1, n-1, n-k-2)\} {}^t\gamma_{\mp}^0(s \pm n \mp 1, n-1, n-k-2) \\ &= \sum_{k=1}^{n-1} \gamma_{\pm}^0(s, n, k) {}^t\gamma_{\mp}^0(s \pm n \mp 1, n-1, n-k-1). \end{aligned}$$

Thus we obtain (3.3) and the following relation:

$$\begin{aligned} &\left\{ \sum_{k=0}^{n-1} \gamma_{\pm}^0(s, n, k) {}^t\gamma_{\mp}^0(s \pm n \mp 1, n-1, n-k-1) \right\} \\ (3.14) \quad &\times \{I - V_{\mp}(s \pm n \mp 1, n-1) V_{\mp}(s \pm n \mp 1, n-1)^+\} = 0. \end{aligned}$$

Hence $\gamma_{\pm}^0(s)$ satisfies (3.2).

Next we shall prove the converse part of Theorem 3.1 by mathematical induction. Let $\gamma_{\pm}(s)$ be any elements of $\mathcal{LMD}_{\pm}(\mathbf{X}_{\pm}^{(s)})$ that satisfy (3.1) and (3.2). Then, from (3.2) and Lemma 2.1

$$\begin{aligned} \delta_{\pm}(s, 1) &= \delta_{\pm}(s, 1) V_{\mp}(s, 0) V_{\mp}(s, 0)^+ \\ (3.15) \quad &= -R(s \pm 1, s) V_{\mp}(s, 0)^+ \\ &= \delta_{\pm}^0(s, 1). \end{aligned}$$

Let $n \geq 2$. We assume that $\gamma_{\pm}(s, m, k) = \gamma_{\pm}^0(s, m, k)$ ($0 \leq k < m < n$). Then from this assumption and (3.1) and (3.2), we know that for all admissible values of s ,

$$\begin{aligned}
 & \{\delta_{\pm}(s, n)(I + A_{\mp}(s \pm n \mp 1, n - 1)) + B_{\pm}(s \pm 1, n - 1)\} \\
 & \quad \times \{I - V_{\mp}(s \pm n \mp 1, n - 1)V_{\mp}(s \pm n \mp 1, n - 1)^+\} \\
 (3.16) \quad & = \left\{ \sum_{k=0}^{n-1} \gamma_{\pm}(s, n, k) {}^t\gamma_{\mp}(s \pm n \mp 1, n - 1, n - k - 1) \right\} \\
 & \quad \times \{I - V_{\mp}(s \pm n \mp 1, n - 1)V_{\mp}(s \pm n \mp 1, n - 1)^+\} \\
 & = 0.
 \end{aligned}$$

Hence from Lemma 2.1, we can apply Lemmas 3.2 and 3.3 in [11] to obtain $\delta_{\pm}(s, n) = \delta_{\pm}^0(s, n)$. This gives, with (3.1), Theorem 2.1(i) and the assumption of mathematical induction that $\gamma_{\pm}(s, n, k) = \gamma_{\pm}^0(s, n, k)$ ($0 \leq k < n$). Thus, we complete the proof. $\square$

Combining Theorems 2.1 and 3.2, we have the following algorithm by which $\mathcal{LM}_{\pm}(\mathbf{X}_{\pm}^{(s)})$ are obtained from the covariance function of $\mathbf{X}$.

ALGORITHM 3.1. *Given the covariance function $R = (R(m, n); 0 \leq m, n \leq N)$ of a flow $\mathbf{X} = (X(n); 0 \leq n \leq N)$, we can calculate $\mathcal{LM}_{\pm}(\mathbf{X}_{\pm}^{(s)})$ as follows:*

[Step 0]

$$(3.17) \quad V_{\pm}(s, 0) \equiv R(s, s).$$

[Step 1]

$$(3.18) \quad \delta_{\pm}^{\min}(s, 1) \equiv -R(s \pm 1, s)V_{\mp}(s, 0)^+,$$

$$(3.19) \quad \delta_{\pm}^0(s, 1) \equiv \delta_{\pm}^{\min}(s, 1),$$

$$(3.20) \quad \gamma_{\pm}^0(s, 1, 0) \equiv \delta_{\pm}^0(s, 1),$$

$$(3.21) \quad V_{\pm}(s, 1) \equiv \{I - \delta_{\pm}^0(s, 1)\delta_{\mp}^0(s \pm 1, 1)\}V_{\pm}(s \pm 1, 0),$$

$$(3.22) \quad A_{\pm}(s, 1) \equiv \gamma_{\pm}^0(s, 1, 0) {}^t\gamma_{\pm}^0(s, 1, 0),$$

$$(3.23) \quad B_{\pm}(s, 1) \equiv \gamma_{\pm}^0(s, 1, 0) {}^t\gamma_{\mp}^0(s, 1, 0).$$

Assume that $\delta_{\pm}^0(s, n - 1)$, $\gamma_{\pm}^0(s, n - 1, k)$, $V_{\pm}(s, n - 1)$, $A_{\pm}(s, n - 1)$, and $B_{\pm}(s, n - 1)$

have been obtained at step $n - 1$.

[Step n]

(3.24)

$$\delta_{\pm}^{\min}(s, n) \equiv - \left\{ R(s \pm n, s) + \sum_{k=0}^{n-2} \gamma_{\pm}^0(s \pm 1, n-1, k) R(s \pm k \pm 1, s) \right\} V_{\mp}(s \pm n \mp 1, n-1)^+,$$

(3.25)

$$\begin{aligned} \delta_{\pm}^0(s, n) &\equiv \{ \delta_{\pm}^{\min}(s, n) - B_{\pm}(s \pm 1, n-1)(I - V_{\mp}(s \pm n \mp 1, n-1)V_{\mp}(s \pm n \mp 1, n-1)^+) \} \\ &\quad \times \{ I + A_{\mp}(s \pm n \mp 1, n-1)(I - V_{\mp}(s \pm n \mp 1, n-1)V_{\mp}(s \pm n \mp 1, n-1)^+) \}^{-1}, \end{aligned}$$

(3.26)

$$\gamma_{\pm}^0(s, n, 0) \equiv \delta_{\pm}^0(s, n),$$

(3.27)

$$\gamma_{\pm}^0(s, n, k) \equiv \gamma_{\pm}^0(s \pm 1, n-1, k-1) + \delta_{\pm}^0(s, n) \gamma_{\mp}^0(s \pm n \mp 1, n-1, n-k-1) \quad (1 \leq k \leq n-1),$$

(3.28)

$$V_{\pm}(s, n) \equiv \{ I - \delta_{\pm}^0(s, n) \delta_{\mp}^0(s \pm n, n) \} V_{\pm}(s \pm 1, n-1),$$

(3.29)

$$\begin{aligned} A_{\pm}(s, n) &\equiv A_{\pm}(s \pm 1, n-1) + \delta_{\pm}^0(s, n)(I + A_{\mp}(s \pm n \mp 1, n-1))^t \delta_{\pm}^0(s, n) \\ &\quad + B_{\pm}(s \pm 1, n-1)^t \delta_{\pm}^0(s, n) + \delta_{\pm}^0(s, n) B_{\mp}(s \pm n \mp 1, n-1), \end{aligned}$$

(3.30)

$$B_{\pm}(s, n) \equiv \sum_{k=0}^{n-1} \gamma_{\pm}^0(s, n, k)^t \gamma_{\mp}^0(s \pm n \mp 1, n, n-k-1).$$

Proof. We have only to show (3.29), which can be easily done by substituting Theorem 2.1(i) into definition (3.6). $\square$

Concerning $V_{\pm}(s, n)V_{\pm}(s, n)^+$, we have the following theorem, which can be verified in the same way as Theorem 3.3 and Lemma 3.5 in [11].

THEOREM 3.3. *Let $\mathbf{X} = (X(n); 0 \leq n \leq N)$ be any flow in W . Then, for all admissible values of s and n ,*

- (i) $V_{\pm}(s, n) \geq V_{\pm}(s \mp 1, n+1)$,
- (ii) $V_{\pm}(s, n)V_{\pm}(s, n)^+ = V_{\pm}(s \mp 1, n+1)V_{\pm}(s \mp 1, n+1)^+$ if and only if $\text{rank } V_{\pm}(s, n) = \text{rank } V_{\pm}(s \mp 1, n+1)$.

4. One-dimensional flows and matrix equations. The relations obtained in the previous section can be simplified in the one-dimensional case. Let $\mathbf{X} = (X(n); 0 \leq n \leq N)$ be a one-dimensional flow in W . Then we have the following theorem.

THEOREM 4.1. *For all admissible values of s and n including $n = 1$,*

$$(4.1) \quad \delta_{\pm}^0(s, n) = \begin{cases} -\frac{R(s \pm n, s) + \sum_{k=0}^{n-2} \gamma_{\pm}^0(s \pm 1, n-1, k) R(s \pm k \pm 1, s)}{V_{\mp}(s \pm n \mp 1, n-1)} \\ -\frac{B_{\pm}(s \pm 1, n-1)}{1 + A_{\mp}(s \pm n \mp 1, n-1)} \end{cases} \begin{matrix} (V_{\mp}(s \pm n \mp 1, n-1) \neq 0), \\ (V_{\mp}(s \pm n \mp 1, n-1) = 0). \end{matrix}$$

Proof. We note $V_{\pm}(s, n) \in \mathbb{R}$. Thus $V_{\pm}(s, n)^+ = 0$ if $V_{\pm}(s, n) = 0$. Substituting this into (3.18), (3.19), (3.24), and (3.25), we have the proof. $\square$

As an application of Algorithm 3.1 and Theorem 4.1, we shall consider the matrix equation

$$(4.2) \quad YA = B,$$

where A is the given $Nd \times M$ matrix consisting of $d \times M$ submatrices A_n ($0 \leq n \leq N-1$), that is, $A = {}^t({}^tA_0, {}^tA_1, \dots, {}^tA_{N-1})$. Moreover, B is a given $d \times M$ matrix and Y is the unknown $d \times Nd$ matrix. We define $\mathbf{X} = (X(n); 0 \leq n \leq N)$ by

$$(4.3) \quad X(n) \equiv A_n \quad (0 \leq n \leq N), \quad A_N \equiv B.$$

Then $\mathbf{X}$ can be regarded as a d -dimensional flow in W , where W is the vector space which consists of all M -dimensional row vectors with the natural inner product. Moreover, we define $d \times nd$ matrices $\Gamma_{\pm}^{(s,n)}$ by

$$(4.4) \quad \Gamma_{\pm}^{(s,n)} \equiv (\gamma_{\pm}^0(s, n, 0), \gamma_{\pm}^0(s, n, 1), \dots, \gamma_{\pm}^0(s, n, n-1)),$$

where $\gamma_{\pm}^0(s) = \gamma_{\pm}(\mathbf{X}_{\pm}^{(s)})$ are the minimum KM₂O-Langevin dissipation matrix functions associated with the flows $\mathbf{X}_{\pm}^{(s)}$. Then we know that $\Gamma_{+}^{(0,N)}$ is the minimum norm least squares solution of (4.2), that is,

$$(4.5) \quad \Gamma_{+}^{(0,N)} = BA^+.$$

More generally, $\Gamma_{\pm}^{(s,n)}$ is the minimum norm least squares solution of the equation

$$(4.6) \quad Y^t({}^tA_s, {}^tA_{s \pm 1}, \dots, {}^tA_{s \pm n \mp 1}) = A_{s \pm n}.$$

Hence, the relations in Algorithm 3.1 can be regarded as recursive relations among the minimum norm least squares solutions of (4.2) and (4.6). We note that in the above argument, the condition on the sizes of the matrices A and B is not essential, since we can adjust their sizes by adding zero row vectors.

In particular, Theorem 4.1 is applicable to the case where $d = 1$.

Acknowledgment. The author would like to thank the referee for his/her helpful comments and suggestions.

REFERENCES

- [1] S. DÉGERINE, *Canonical partial autocorrelation function of a multivariate time series*, Ann. Statist., 18 (1990), pp. 961–971.

- [2] S. DÉGERINE AND R. MALKI, *Second-order blind separation of sources based on canonical partial innovations*, IEEE. Trans. Signal Processing, 48 (2000), pp. 629–641.
- [3] N. DELFOSSE AND P. LOUBATON, *Adaptive blind separation of independent sources: A second-order stable algorithm for the general case*, IEEE. Trans. Circuits and System. I: Fundamental Theory and Applications, 47 (2000), pp. 1056–1071.
- [4] J. DURBIN, *The fitting of time series models*, Rev. Inst. Int. Stat., 28 (1960), pp. 233–244.
- [5] C. GUEGUEN, *Linear prediction in the singular case and the stability of eigen models*, in Proc. Int. Conf. Acoust. Speech, Signal Processing, Atlanta, GA, 1981, pp. 881–885.
- [6] Y. INOUE, *Modeling of multichannel time series and extrapolation of matrix-valued autocorrelation sequences*, IEEE Trans. Acoust., Speech, Signal Process., 31 (1983), pp. 45–55.
- [7] Y. INOUE, *Autoregressive model fitting for multichannel time series of degenerate rank: Limit properties*, IEEE Trans. Circuits and Systems, 32 (1985), pp. 252–259.
- [8] M. KLIMEK, E. KARLSSON, M. MATSUURA AND Y. OKABE, *A geometric proof of the fluctuation-dissipation theorem for the KM_2O -Langevin equation*, Hokkaido Math. J., 31 (2002), pp. 615–628.
- [9] H. LEV-ARI AND T. KAILATH, *Schur and Levinson algorithms for nonstationary processes*, Proc. 1981 IEEE Int. Conf. On Acoust., Speech, Signal Processing, 1981, pp. 860–864.
- [10] N. LEVINSON, *The Wiener rms (root mean square) error criterion in filter design and prediction*, J. Math. Phys. Mass. Inst. Tech., 25 (1947), pp. 261–278.
- [11] M. MATSUURA, *On a new fluctuation-dissipation theorem for degenerate stationary flows*, Methodol. Comput. Appl. Probab., 5 (2003), pp. 369–387.
- [12] M. MATSUURA AND Y. OKABE, *On a non-linear prediction problem for one-dimensional stochastic processes*, Japan. J. Math., 27 (2001), pp. 51–112.
- [13] M. MATSUURA AND Y. OKABE, *On the theory of KM_2O -Langevin equations for non-stationary and degenerate flows*, J. Math. Soc. Japan, 55 (2003), pp. 523–563.
- [14] Y. OKABE, *Langevin equations and causal analysis*, Amer. Math. Soc. Transl. Ser. 2, 161 (1994), pp. 19–50.
- [15] Y. OKABE, *On the theory of KM_2O -Langevin equations for stationary flows. I. Characterization theorem*, J. Math. Soc. Japan, 51 (1999), pp. 817–841.
- [16] Y. OKABE, *On the theory of KM_2O -Langevin equations for stationary flows. II. Construction theorem*, Acta Appl. Math., 63 (2000), pp. 307–322.
- [17] Y. OKABE AND M. MATSUURA, *On the theory of KM_2O -Langevin equations for stationary flows. III. Extension theorem*, Hokkaido Math. J., 29 (2000), pp. 369–382.
- [18] H. SAKAI, *Some results on multichannel and periodic autoregression processes*, Technical rept. TK123-1, Information System Lab., Stanford Univ., Stanford, CA, 1988.
- [19] P. WHITTLE, *On the fitting of multivariate autoregressions, and the approximate canonical factorization of a spectral density matrix*, Biometrika, 50 (1963), pp. 129–134.
- [20] R. A. WIGGINS AND E. A. ROBINSON, *Recursive solution to the multichannel filtering problem*, J. Geophys. Res., 70 (1965), pp. 1885–1891.

DISC SEPARATION OF THE SCHUR COMPLEMENT OF DIAGONALLY DOMINANT MATRICES AND DETERMINANTAL BOUNDS*

JIANZHOU LIU[†] AND FUZHEN ZHANG[‡]

Abstract. We consider the Geršgorin disc separation from the origin for (doubly) diagonally dominant matrices and their Schur complements, showing that the separation of the Schur complement of a (doubly) diagonally dominant matrix is greater than that of the original grand matrix. As application we discuss the localization of eigenvalues and present some upper and lower bounds for the determinant of diagonally dominant matrices.

Key words. Brauer theorem, comparison matrix, diagonally dominant matrix, doubly diagonally dominant matrix, Geršgorin theorem, H -matrix, M -matrix, Schur complement, separation

AMS subject classifications. 15A45, 15A48

DOI. 10.1137/040620369

1. Introduction. Let A be an $n \times n$ complex matrix. The famous Geršgorin theorem gives a union of discs in the complex plane that contain all eigenvalues of A . An individual disc comprises the complex numbers z for which

$$|z - a_{ii}| \leq P_i(A),$$

where

$$P_i(A) = \sum_{j=1, j \neq i}^n |a_{ij}|.$$

This reveals immediately that if A is *strictly diagonally (row) dominant*, i.e.,

$$|a_{ii}| > P_i(A)$$

for all $i = 1, 2, \dots, n$, then no discs contain O , the origin, and thus A is nonsingular. In such a case, the quantities $|a_{ii}| - P_i(A)$ measure the separations of the discs from the origin and give estimates of the “shortest” eigenvalue of the matrix.

A well-known result due to Brauer generalizes the Geršgorin theorem to a union of ovals (of Cassini) that are guaranteed to contain all eigenvalues of A [7, p. 380]. An oval is given to comprise all complex numbers z satisfying

$$|z - a_{ii}||z - a_{jj}| \leq P_i(A)P_j(A), \quad i < j.$$

*Received by the editors December 7, 2004; accepted for publication (in revised form) by H. J. Woerdeman May 18, 2005; published electronically December 7, 2005. This work was supported in part by the Science and Research Fund of Education Department (02A010) and the Natural Science Foundation (04JJ40005) of Hunan Province of China, and also by the FCAS Faculty Development Fund of Nova Southeastern University.

<http://www.siam.org/journals/simax/27-3/62036.html>

[†]Department of Mathematics, Xiangtan University, Xiangtan, Hunan 411105, China (liujz@xtu.edu.cn).

[‡]Division of Math, Science and Technology, Nova Southeastern University, 3301 College Ave., Fort Lauderdale, FL 33314, and School of Mathematics and Systems Science, Shenyang Normal University, Shenyang, Liaoning 110034, China (zhang@nova.edu).

The Geršgorin discs and the Cassini ovals are both effective tools for locating the eigenvalues (spectrum) of a square matrix. As the former ensures the nonsingularity of the strictly diagonally dominant matrices (SD), the latter guarantees the same property of the strictly doubly diagonally dominant matrices (SDD) [7, p. 381]; recall that an *SDD* [10, 13] is a matrix such that for all i and j , $1 \leq i < j \leq n$,

$$(1.1) \quad |a_{ii}||a_{jj}| > P_i(A)P_j(A).$$

Doubly diagonally dominant matrices (for which equality in (1.1) is allowed to occur) are generalizations of diagonally dominant matrices. Both matrix classes possess the properties of nonsingularity and closedness under the Schur complementation (see, e.g., [2, 13]). These properties have been demonstrated to serve as rich and basic tools in numerical analysis (mostly for convergence of iterations) and in matrix analysis (mostly for deriving matrix inequalities). As a result of Kahan (see [15]) shows a relation between the fields of values of a matrix and its Schur complement, this study shows a relation between the Geršgorin discs of a matrix and its Schur complement.

In this paper we first consider the disc separations of the Schur complement; more precisely, we compare a disc separation of the Schur complement to that of the original matrix and show that each Geršgorin disc of the Schur complement is paired with a particular Geršgorin disc of the original matrix; the latter is further from the origin than the former. As applications of our main results, we then discuss localization of eigenvalues and present some upper and lower bounds for the determinants of the diagonally dominant matrices.

We refer the reader to [18, 19] for the *spread* of a Hermitian matrix, which is defined to be the largest eigenvalue of the matrix minus the smallest eigenvalue. It measures the separation between the extreme eigenvalues. One may refer to [1] for general theory on nonnegative matrices and M -matrices, [3, 4, 5, 6, 9, 13, 16] for detailed discussions on (generalized) diagonally dominant matrices, [11, 12, 14, 17] for computational aspects of H -matrices, and [21] for an extensive survey of the results on the Schur complement.

2. Lemmas. We begin this section by recalling a few terms and results that are to be used to prove our theorems. Let $A = (a_{ij})$ be an $n \times n$ complex matrix. The *comparison matrix*, denoted by $\mu(A) = (c_{ij})$, of A is defined to be

$$c_{ij} = \begin{cases} |a_{ij}| & \text{if } i = j, \\ -|a_{ij}| & \text{if } i \neq j. \end{cases}$$

A matrix A is an M -matrix if it can be written in the form $A = mI - P$, where P is a nonnegative matrix and $m > \rho(P)$, the spectral radius of P . Note that if A is an M -matrix, then so is the Schur complement of A and $\det A > 0$ (see, e.g., [8, Chap. 2, sec. 2.5]). A matrix A is an H -matrix if $\mu(A)$ is an M -matrix. We denote by $\mathbb{H}_n$ and $\mathbb{M}_n$ the sets of $n \times n$ H - and M -matrices, respectively.

For matrix $A = (a_{ij})$, we denote $|A| = (|a_{ij}|)$. If the entries of the matrix A are all nonnegative, then we write $A \geq 0$. For real matrices A and B of the same size, if $A - B$ is a nonnegative matrix, we designate $A \geq B$.

LEMMA 1. *If A is an H -matrix, then*

$$(2.1) \quad [\mu(A)]^{-1} \geq |A^{-1}|.$$

Proof. For the proof, see, e.g., [8, pp. 117, 131]. $\square$

LEMMA 2. *If $A \in \text{SD}_n$ or SDD_n , then $\mu(A) \in \mathbb{M}_n$, i.e., $A \in \mathbb{H}_n$.*

LEMMA 3. If $A \in \text{SD}_n$ or SDD_n and α is a proper subset of $N \equiv \{1, 2, \dots, n\}$, then the Schur complement of A is in $\text{SD}_{|\alpha^c|}$ or $\text{SDD}_{|\alpha^c|}$, where $\alpha^c = N - \alpha$ is the complement of α in N and $|\alpha^c|$ is the cardinality of α^c .

Lemmas 2 and 3 can be found in, e.g., [8, p. 114] and [13, Theorem 2.1].

We recall that the Schur complement of A with respect to the nonsingular submatrix $A(\alpha)$, denoted by $A/A(\alpha)$ or simply A/α , is defined to be

$$(2.2) \quad A(\alpha^c) - A(\alpha^c, \alpha)[A(\alpha)]^{-1}A(\alpha, \alpha^c),$$

where $A(\alpha, \beta)$ stands for the submatrix of A lying in the rows indexed by α and the columns indexed by β ; $A(\alpha)$ is an abbreviation for $A(\alpha, \alpha)$.

Our next lemma, showing a way of constructing M -matrices from a given SD matrix, is key to the proofs of our theorems in the following section.

LEMMA 4. Let $A \in \text{SD}_n$, $\alpha = \{i_1, i_2, \dots, i_k\}$ be a proper subset of N , and let $\alpha^c = N - \alpha = \{j_1, j_2, \dots, j_l\}$, $k + l = n$. For any $j_t \in \alpha^c$, denote

$$B_{j_t} \equiv \begin{pmatrix} x & -|a_{j_t i_1}| & \cdots & -|a_{j_t i_k}| \\ -\sum_{u=1}^l |a_{i_1 j_u}| & & & \\ \vdots & & \mu[A(\alpha)] & \\ -\sum_{u=1}^l |a_{i_k j_u}| & & & \end{pmatrix}, \quad x > 0.$$

Then B_{j_t} is a doubly diagonally dominant matrix if and only if

$$(2.3) \quad x \geq \max_{1 \leq w \leq k} \frac{P_{i_w}(A)}{|a_{i_w i_w}|} \sum_{v=1}^k |a_{j_t i_v}|.$$

When strict inequality in (2.3) holds, B_{j_t} is an M -matrix of order $k + 1$, and thus $\det B_{j_t} > 0$. If equality in (2.3) occurs, then $\det B_{j_t} \geq 0$.

Proof. For simplicity, write $B_{j_t} \equiv B \equiv (b_{pq})$. The off-diagonal entries of B are all nonpositive. Notice that $b_{11} = x$, $b_{w+1, w+1} = |a_{i_w i_w}|$, $w = 1, 2, \dots, k$, and

$$P_1(B) = \sum_{v=1}^k |a_{j_t i_v}|, \quad P_{w+1}(B) = P_{i_w}(A).$$

In order for B to be strictly doubly diagonally dominant, for each w ,

$$|b_{11}| |b_{w+1, w+1}| = x |a_{i_w i_w}| > P_1(B) P_{w+1}(B) = \sum_{v=1}^k |a_{j_t i_v}| P_{i_w}(A),$$

which yields (2.3), while for $w, y = 1, 2, \dots, k$ with $w \neq y$, since $A \in \text{SD}_n$,

$$|b_{w+1, w+1}| |b_{y+1, y+1}| = |a_{i_w i_w}| |a_{i_y i_y}| > P_{i_w}(A) P_{i_y}(A) = P_{w+1}(B) P_{y+1}(B).$$

Therefore, by definition, B is strictly doubly diagonally dominant. By Lemma 2, $B = \mu(B)$ is an M -matrix and thus $\det B > 0$. The equality case follows from a continuity argument (with $x + \epsilon$ in B_{j_t} and letting $\epsilon \rightarrow 0^+$). $\square$

3. Disc separation of the Schur complement. Let A be an $n \times n$ SD. Then all diagonal entries of A are necessarily nonzero. Moreover, each disc $\{z : |z - a_{ii}| \leq P_i(A)\}$ is separated from the origin, and $|a_{ii}| - P_i(A)$ is the (shortest) distance from the disc to the origin. As the Schur complement of an SD is strictly diagonally dominant, we are interested in the separations of the discs of the Schur complements from the origin compared with those of the original matrix. Our first theorem shows that the discs of the Schur complements are separated further from the origin. For this purpose, consider the ratios $\frac{|a_{i_v i_v}| - P_{i_v}(A)}{|a_{i_v i_v}|}$, and define for each $j_t \in \alpha^c$

$$(3.1) \quad w_{j_t} = \min_{1 \leq v \leq k} \frac{|a_{i_v i_v}| - P_{i_v}(A)}{|a_{i_v i_v}|} \sum_{u=1}^k |a_{j_t i_u}|.$$

Note that for a number a , nonsingular matrix S , and vectors x and y

$$\frac{1}{\det S} \det \begin{pmatrix} a & x \\ y & S \end{pmatrix} = a - xS^{-1}y.$$

We are now ready to present our main result.

THEOREM 1. Let A be an $n \times n$ SD, $\alpha = \{i_1, i_2, \dots, i_k\} \subset N$, $\alpha^c = N - \alpha = \{j_1, j_2, \dots, j_l\}$, $k+l = n$. Let w_{j_t} be defined as in (3.1) and denote $A/\alpha = (a'_{ts})$. Then

$$(3.2) \quad |a'_{tt}| - P_t \left(\frac{A}{\alpha} \right) \geq |a_{j_t j_t}| - P_{j_t}(A) + w_{j_t} \geq |a_{j_t j_t}| - P_{j_t}(A) > 0$$

and

$$(3.3) \quad |a'_{tt}| + P_t \left(\frac{A}{\alpha} \right) \leq |a_{j_t j_t}| + P_{j_t}(A) - w_{j_t} \leq |a_{j_t j_t}| + P_{j_t}(A).$$

Proof. Note that, by (2.1) and Lemma 2,

$$\{\mu[A(\alpha)]\}^{-1} \geq [A(\alpha)]^{-1}.$$

For $t = 1, 2, \dots, l$, by definition of the Schur complement (2.2), we compute

$$\begin{aligned} |a'_{tt}| - P_t \left(\frac{A}{\alpha} \right) &= |a'_{tt}| - \sum_{\substack{s \neq t \\ s=1}}^l |a'_{ts}| \\ &= \left| a_{j_t j_t} - (a_{j_t i_1}, \dots, a_{j_t i_k})[A(\alpha)]^{-1} \begin{pmatrix} a_{i_1 j_t} \\ \vdots \\ a_{i_k j_t} \end{pmatrix} \right| \\ &\quad - \sum_{\substack{s \neq t \\ s=1}}^l \left| a_{j_t j_s} - (a_{j_t i_1}, \dots, a_{j_t i_k})[A(\alpha)]^{-1} \begin{pmatrix} a_{i_1 j_s} \\ \vdots \\ a_{i_k j_s} \end{pmatrix} \right| \\ &\geq |a_{j_t j_t}| - \sum_{\substack{s \neq t \\ s=1}}^l |a_{j_t j_s}| - \sum_{s=1}^l \left| (a_{j_t i_1}, \dots, a_{j_t i_k})[A(\alpha)]^{-1} \begin{pmatrix} a_{i_1 j_s} \\ \vdots \\ a_{i_k j_s} \end{pmatrix} \right| \\ &\geq |a_{j_t j_t}| - \sum_{\substack{s \neq t \\ s=1}}^l |a_{j_t j_s}| - \sum_{s=1}^l (|a_{j_t i_1}|, \dots, |a_{j_t i_k}|) \{\mu[A(\alpha)]\}^{-1} \begin{pmatrix} |a_{i_1 j_s}| \\ \vdots \\ |a_{i_k j_s}| \end{pmatrix} \end{aligned}$$

$$\begin{aligned}
&= |a_{j_t j_t}| - P_{j_t}(A) + \sum_{u=1}^k |a_{j_t i_u}| + w_{j_t} - w_{j_t} \\
&\quad - \sum_{s=1}^l (|a_{j_t i_1}|, \dots, |a_{j_t i_k}|) \{\mu[A(\alpha)]\}^{-1} \begin{pmatrix} |a_{i_1 j_s}| \\ \vdots \\ |a_{i_k j_s}| \end{pmatrix} \\
&= |a_{j_t j_t}| - P_{j_t}(A) + w_{j_t} \\
&\quad + \sum_{u=1}^k |a_{j_t i_u}| - w_{j_t} - (|a_{j_t i_1}|, \dots, |a_{j_t i_k}|) \{\mu[A(\alpha)]\}^{-1} \begin{pmatrix} \sum_{s=1}^l |a_{i_1 j_s}| \\ \vdots \\ \sum_{s=1}^l |a_{i_k j_s}| \end{pmatrix} \\
&= |a_{j_t j_t}| - P_{j_t}(A) + w_{j_t} \\
&\quad + \frac{1}{\det\{\mu[A(\alpha)]\}} \det \begin{pmatrix} \sum_{u=1}^k |a_{j_t i_u}| - w_{j_t} & -|a_{j_t i_1}| & \dots & -|a_{j_t i_k}| \\ -\sum_{s=1}^l |a_{i_1 j_s}| & & & \\ \vdots & & \mu[A(\alpha)] & \\ -\sum_{s=1}^l |a_{i_k j_s}| & & & \end{pmatrix} \\
&= |a_{j_t j_t}| - P_{j_t}(A) + w_{j_t} + \frac{1}{\det\{\mu[A(\alpha)]\}} \det B,
\end{aligned}$$

where B is the matrix containing $\mu[A(\alpha)]$ on the right-hand side. By taking x in Lemma 4 to be $\sum_{u=1}^k |a_{j_t i_u}| - w_{j_t}$ (which is greater than or equal to the expression on the right-hand side of (2.3) by computation), we see $\det B \geq 0$, and thus (3.2) follows.

In a similar way, one may show that

$$\begin{aligned}
|a'_{tt}| + P_t \left(\frac{A}{\alpha} \right) &= |a'_{tt}| + \sum_{\substack{s \neq t \\ s=1}}^l |a'_{ts}| \\
&\leq |a_{j_t j_t}| + P_{j_t}(A) - w_{j_t} - \frac{1}{\det\{\mu[A(\alpha)]\}} \det B \\
&\leq |a_{j_t j_t}| + P_{j_t}(A) - w_{j_t} \\
&\leq |a_{j_t j_t}| + P_{j_t}(A).
\end{aligned}$$

This reveals (3.3). $\square$

The inequalities (3.2) immediately yield the following well-known result [2].

COROLLARY 1. *The Schur complement of an SD is strictly diagonally dominant.*

COROLLARY 2. *Let $A \in \text{SD}_n$ and take $\alpha = \{1, 2, \dots, n-1\}$. Then*

$$|a_{nn}| + \max_{1 \leq i \leq n-1} \frac{P_i(A)}{|a_{ii}|} P_n(A) \geq \left| \frac{A}{\alpha} \right| \geq |a_{nn}| - \max_{1 \leq i \leq n-1} \frac{P_i(A)}{|a_{ii}|} P_n(A) > 0.$$

Proof. Notice that α^c contains only one element $j_t = n$. Thus, A/α is nothing

but a number, and so $P_t(A/\alpha) = 0$, and, from (3.1),

$$\begin{aligned} w_{j_t} = w_n &= \min_{1 \leq i \leq n-1} \frac{|a_{ii}| - P_i(A)}{|a_{ii}|} \sum_{u=1}^{n-1} |a_{nu}| \\ &= \min_{1 \leq i \leq n-1} \left(1 - \frac{P_i(A)}{|a_{ii}|} \right) P_n(A) \\ &= P_n(A) - \max_{1 \leq i \leq n-1} \frac{P_i(A)}{|a_{ii}|} P_n(A). \end{aligned}$$

Substituting this into (3.2) and (3.3) results in the desired inequalities. $\square$

Remark 1. Corollary 2 can be proven directly by computation. Note that

$$P_n(A) - w_n = P_n(A) \max_{1 \leq i \leq n-1} \frac{P_i(A)}{|a_{ii}|}.$$

COROLLARY 3. *Let $A = (a_{ij}) \in \text{SD}_n$ (complex). Denote $A^{-1} = (a^{ij})$. Then*

$$a_{ii} > 0 \iff a^{ii} > 0.$$

If $d_+(X)$ stands for the number of positive entries on the main diagonal of X ,

$$d_+[A(\alpha)] = d_+[A^{-1}(\alpha)].$$

Proof. Without loss of generality, take $a_{ii} = a_{nn}$. Then $a^{nn} = (A/\beta)^{-1}$, where $\beta = \{1, 2, \dots, n-1\}$ (see, e.g., [20, p. 37]). Since A is an SD, by (3.2), $A/\beta > 0$, and thus $a^{nn} > 0$. The identity $d_+[A(\alpha)] = d_+[A^{-1}(\alpha)]$ follows at once. $\square$

Remark 2. The t th disc of A/α is paired with the j_t th disc of A .

Now we turn our attention to doubly diagonally dominant matrices. If A is in SDD but not in SD, by (1.1), there is one and only one index i_0 , say, such that

$$(3.4) \quad |a_{i_0 i_0}| \leq P_{i_0}(A).$$

THEOREM 2. *Let A be an $n \times n$ SDD and i_0 , $1 \leq i_0 \leq n$, be such as in (3.4). Then for any index set α containing i_0 , writing $\alpha = \{i_1, i_2, \dots, i_k\}$, $\alpha^c = N - \alpha = \{j_1, j_2, \dots, j_l\}$, and $A/\alpha = (a'_{ts})$,*

$$(3.5) \quad |a'_{tt}| - P_t \left(\frac{A}{\alpha} \right) \geq |a_{j_t j_t}| - P_{j_t}(A) + \left(1 - \frac{P_{i_0}(A)}{|a_{i_0 i_0}|} \right) \sum_{v=1}^k |a_{j_t i_v}|$$

$$(3.6) \quad \geq |a_{j_t j_t}| - \frac{P_{i_0}(A)}{|a_{i_0 i_0}|} P_{j_t}(A) > 0$$

and

$$\begin{aligned} |a'_{tt}| + P_t \left(\frac{A}{\alpha} \right) &\leq |a_{j_t j_t}| + P_{j_t}(A) - \left(1 - \frac{P_{i_0}(A)}{|a_{i_0 i_0}|} \right) \sum_{v=1}^k |a_{j_t i_v}| \\ &\leq |a_{j_t j_t}| + \frac{P_{i_0}(A)}{|a_{i_0 i_0}|} P_{j_t}(A). \end{aligned}$$

Proof. Once again we compute, by definition (2.2),

$$\begin{aligned}
|a'_{tt}| - P_t\left(\frac{A}{\alpha}\right) &= |a'_{tt}| - \sum_{\substack{s \neq t \\ s=1}}^l |a'_{ts}| \\
&= \left| a_{j_t j_t} - (a_{j_t i_1}, \dots, a_{j_t i_k})[A(\alpha)]^{-1} \begin{pmatrix} a_{i_1 j_t} \\ \vdots \\ a_{i_k j_t} \end{pmatrix} \right| \\
&\quad - \sum_{\substack{s \neq t \\ s=1}}^l \left| a_{j_t j_s} - (a_{j_t i_1}, \dots, a_{j_t i_k})[A(\alpha)]^{-1} \begin{pmatrix} a_{i_1 j_s} \\ \vdots \\ a_{i_k j_s} \end{pmatrix} \right| \\
&\geq |a_{j_t j_t}| - P_{j_t}(A) + \left(1 - \frac{P_{i_0}(A)}{|a_{i_0 i_0}|}\right) \sum_{v=1}^k |a_{j_t i_v}| + \frac{P_{i_0}(A)}{|a_{i_0 i_0}|} \sum_{v=1}^k |a_{j_t i_v}| \\
&\quad - \sum_{s=1}^l (|a_{j_t i_1}|, \dots, |a_{j_t i_k}|) \{\mu[A(\alpha)]\}^{-1} \begin{pmatrix} |a_{i_1 j_s}| \\ \vdots \\ |a_{i_k j_s}| \end{pmatrix} \\
&= |a_{j_t j_t}| - P_{j_t}(A) + \left(1 - \frac{P_{i_0}(A)}{|a_{i_0 i_0}|}\right) \sum_{v=1}^k |a_{j_t i_v}| \\
&\quad + \frac{1}{\det\{\mu[A(\alpha)]\}} \det \begin{pmatrix} \frac{P_{i_0}(A)}{|a_{i_0 i_0}|} \sum_{v=1}^k |a_{j_t i_v}| & -|a_{j_t i_1}| & \dots & -|a_{j_t i_k}| \\ -\sum_{s=1}^l |a_{i_1 j_s}| & & & \\ \vdots & & \mu[A(\alpha)] & \\ -\sum_{s=1}^l |a_{i_k j_s}| & & & \end{pmatrix} \\
&= |a_{j_t j_t}| - P_{j_t}(A) + \left(1 - \frac{P_{i_0}(A)}{|a_{i_0 i_0}|}\right) \sum_{v=1}^k |a_{j_t i_v}| + \frac{1}{\det\{\mu[A(\alpha)]\}} \det \tilde{B},
\end{aligned}$$

where $\tilde{B}$ is the large size matrix containing $\mu[A(\alpha)]$. By taking x in Lemma 4 to be $\frac{P_{i_0}(A)}{|a_{i_0 i_0}|} \sum_{v=1}^k |a_{j_t i_v}|$, we see that $\det \tilde{B} \geq 0$, and thus (3.5) follows. Replacing $\sum_{v=1}^k |a_{j_t i_v}|$ in (3.5) by $P_{j_t}(A)$ results in (3.6). The second set of inequalities in the theorem are similarly proven. $\square$

4. Applications: Bounds for determinants and localization of eigenvalues. In this section we make use of the results in the previous section to present some upper and lower bounds for determinants. Also, as an application of Theorem 1, we show a result on the localization of eigenvalues of the diagonally dominant matrices.

Let $\{j_1, j_2, \dots, j_n\}$ be a rearrangement of the elements in $N = \{1, 2, \dots, n\}$. Denote $\alpha_1 = \{j_n\}$, $\alpha_2 = \{j_n, j_{n-1}\}$, $\dots$, $\alpha_n = \{j_1, j_2, \dots, j_n\} = N$. Then $\alpha_{n-k+1} - \alpha_{n-k} = \{j_k\}$, $k = 1, 2, \dots, n$, with $\alpha_0 = \emptyset$, and

$$P_{j_k}[A(\alpha_{n-k+1})] = \sum_{u \in \alpha_{n-k}} |a_{j_k u}|.$$

Let $\mathcal{J}$ represent any rearrangement $\{j_1, j_2, \dots, j_n\}$ of the elements in N with $\alpha_1, \alpha_2, \dots, \alpha_n$ defined as above.

THEOREM 3. *Let A be an $n \times n$ SD. Then*

$$(4.1) \quad \det A \geq \max_{\mathcal{J}} \prod_{k=1}^n \left\{ |a_{j_k j_k}| - \max_{u \in \alpha_{n-k}} \frac{P_u[A(\alpha_{n-k+1})]}{|a_{uu}|} P_{j_k}[A(\alpha_{n-k+1})] \right\}$$

and

$$(4.2) \quad \det A \leq \min_{\mathcal{J}} \prod_{k=1}^n \left\{ |a_{j_k j_k}| + \max_{u \in \alpha_{n-k}} \frac{P_u[A(\alpha_{n-k+1})]}{|a_{uu}|} P_{j_k}[A(\alpha_{n-k+1})] \right\}.$$

Proof. For the first inequality, since α_{n-k} is contained in α_{n-k+1} and $\alpha_{n-k+1} - \alpha_{n-k} = \{j_k\}$, we have, by Corollary 2, for each k ,

$$\left| \frac{A(\alpha_{n-k+1})}{\alpha_{n-k}} \right| \geq |a_{j_k j_k}| - \max_{u \in \alpha_{n-k}} \frac{P_u[A(\alpha_{n-k+1})]}{|a_{uu}|} P_{j_k}[A(\alpha_{n-k+1})] > 0.$$

It follows that

$$\begin{aligned} |\det A| &= \left| \frac{\det A}{\det[A(\alpha_{n-1})]} \right| \left| \frac{\det[A(\alpha_{n-1})]}{\det[A(\alpha_{n-2})]} \right| \cdots \left| \frac{\det[A(\alpha_2)]}{\det[A(\alpha_1)]} \right| |\det[A(\alpha_1)]| \\ &= \left| \det \left(\frac{A}{\alpha_{n-1}} \right) \right| \left| \det \left[\frac{A(\alpha_{n-1})}{\alpha_{n-2}} \right] \right| \cdots \left| \det \left[\frac{A(\alpha_2)}{\alpha_1} \right] \right| |\det A(\alpha_1)| \\ &= |a_{j_n j_n}| \prod_{k=1}^{n-1} \left| \frac{A(\alpha_{n-k+1})}{\alpha_{n-k}} \right| \\ &\geq |a_{j_n j_n}| \prod_{k=1}^{n-1} \left\{ |a_{j_k j_k}| - \max_{u \in \alpha_{n-k}} \frac{P_u[A(\alpha_{n-k+1})]}{|a_{uu}|} P_{j_k}[A(\alpha_{n-k+1})] \right\}. \end{aligned}$$

This implies (4.1). The inequality (4.2) in the theorem is similarly proven. $\square$

Notice that when $A \in \text{SD}_n$, for any $1 \leq k \leq n$ and $u \in \alpha_{n-k}$,

$$0 \leq \frac{P_u[A(\alpha_{n-k+1})]}{|a_{uu}|} \leq 1.$$

We have the following claim immediately from the theorem.

COROLLARY 4. *Let A be an $n \times n$ SD. Then*

$$\det A \geq \max_{\mathcal{J}} \prod_{k=1}^n \left\{ |a_{j_k j_k}| - \sum_{u \in \alpha_{n-k}} |a_{j_k u}| \right\}$$

and

$$\det A \leq \min_{\mathcal{J}} \prod_{k=1}^n \left\{ |a_{j_k j_k}| + \sum_{u \in \alpha_{n-k}} |a_{j_k u}| \right\}.$$

Example. Let $n = 3$ and take $j_1 = 3$, $j_2 = 1$, and $j_3 = 2$. With $\alpha_1 = \{j_3\} = \{2\}$, $\alpha_2 = \{j_3, j_2\} = \{1, 2\}$, and $\alpha_3 = \{j_3, j_2, j_1\} = \{1, 2, 3\}$, by Corollary 4, we have for any $A \in \text{SD}_3$

$$|\det A| \geq |a_{22}|(|a_{33}| - |a_{31}| - |a_{32}|)(|a_{11}| - |a_{12}|)$$

and

$$|\det A| \leq |a_{22}|(|a_{33}| + |a_{31}| + |a_{32}|)(|a_{11}| + |a_{12}|).$$

For an analogous result for SDDs, let $\mathcal{I}$ denote all rearrangements of the elements in N with $\alpha_1 = \{i_0 \equiv j_n\}$, where

$$|a_{i_0 i_0}| \leq P_{i_0}(A).$$

The proof of the following theorem is similar to that of the previous theorem.

THEOREM 4. *Let $A \in \text{SDD}_n$ with $|a_{i_0 i_0}| \leq P_{i_0}(A)$. Then*

$$\det A \geq \max_{\mathcal{I}} |a_{i_0 i_0}| \prod_{k=1}^{n-1} \left\{ |a_{j_k j_k}| - \frac{P_{i_0}[A(\alpha_{n-k+1})]}{|a_{i_0 i_0}|} P_{j_k}[A(\alpha_{n-k+1})] \right\}$$

and

$$\det A \leq \min_{\mathcal{I}} |a_{i_0 i_0}| \prod_{k=1}^{n-1} \left\{ |a_{j_k j_k}| + \frac{P_{i_0}[A(\alpha_{n-k+1})]}{|a_{i_0 i_0}|} P_{j_k}[A(\alpha_{n-k+1})] \right\}.$$

We conclude the article by presenting a result that falls in the localization of eigenvalues of the diagonally dominant matrices.

THEOREM 5. *Let $A \in \text{SD}_n$ be a matrix with real diagonal entries and α be a proper subset of $N = \{1, 2, \dots, n\}$. Then A/α and $A(\alpha^c)$ have the same number of eigenvalues whose real parts are greater (less) than w (resp., $-w$), where*

$$w = \min_{j \in \alpha^c} \left[|a_{jj}| - P_j(A) + \min_{i \in \alpha} \frac{|a_{ii}| - P_i(A)}{|a_{ii}|} \sum_{i \in \alpha} |a_{ji}| \right].$$

Proof. By (3.2) in Theorem 1, the matrix $\mu(A/\alpha) - wI$ is diagonally dominant, and thus is $A/\alpha - wI$. In addition, by Theorem 1 again, since $a'_{tt} - w > 0$ if and only if $a'_{tt} > 0$, there are the same number of positive entries on the main diagonals of $A/\alpha - wI$ and A/α ; that is, $d_+(A/\alpha - wI) = d_+(A/\alpha)$. Thus, by the Geršgorin theorem, $A/\alpha - wI$ has $d_+(A/\alpha)$ eigenvalues with positive real part (on the open right-half complex plane). On the other hand, the eigenvalues of $A/\alpha - wI$ are the eigenvalues of A/α minus w , so A/α has $d_+(A/\alpha)$ eigenvalues with positive real part greater than w . Observe that $A^{-1}(\alpha^c) = (A/\alpha)^{-1}$ (see, e.g., [20, p. 184]) and further that $(A/\alpha)^{-1}$ and A/α have the same number of eigenvalues with positive real part. By Corollary 3, we conclude that A/α and $A(\alpha^c)$ have the same number of eigenvalues whose real parts are greater than w . For the number of negative parts of eigenvalues, the above argument works with $-A/\alpha$ in place of A/α . $\square$

An immediate geometric explanation of Theorem 5 is that if the principal submatrix $A(\alpha^c)$ has no eigenvalue between the vertical lines $x = -w$ and $x = w$, i.e., the band $|z| \leq w$, then the Schur complement A/α has no eigenvalue in the band.

REFERENCES

- [1] A. BERMAN AND R. J. PLEMMONS, *Nonnegative Matrices in the Mathematical Sciences*, Academic Press, New York, 1979.
- [2] D. CARLSON AND T. MARKHAM, *Schur complements of diagonally dominant matrices*, Czech. Math. J., 29 (1979), pp. 246–251.

- [3] M. FIEDLER AND V. PTÁK, *On the matrices with non-positive off-diagonal elements and positive principal minors*, Czech. Math. J., 12 (1962), pp. 382–400.
- [4] M. FIEDLER AND V. PTÁK, *Diagonally dominant matrices*, Czech. Math. J., 17 (1967), pp. 420–433.
- [5] Y.-M. GAO AND X.-H. WANG, *Criteria for generalized diagonally dominant matrices and M -matrices*, Linear Algebra Appl., 169 (1992), pp. 257–268.
- [6] Y.-M. GAO AND X.-H. WANG, *Criteria for generalized diagonally dominant matrices and M -matrices II*, Linear Algebra Appl., 248 (1996), pp. 339–353.
- [7] R. A. HORN AND C. R. JOHNSON, *Matrix Analysis*, Cambridge University Press, New York, 1985.
- [8] R. A. HORN AND C. R. JOHNSON, *Topics in Matrix Analysis*, Cambridge University Press, New York, 1991.
- [9] T.-Z. HUANG, *A note on generalized diagonally dominant matrices*, Linear Algebra Appl., 225 (1995), pp. 237–242.
- [10] K. D. IKRAMOV, *Invariance of the Brauer diagonal dominance in Gaussian elimination*, Moscow Univ. Comput. Math. Cybernet., N2 (1989), pp. 91–94.
- [11] T. KOHNO, H. NIKI, H. SAWAMI, AND Y.-M. GAO, *An iterative test for H -matrix*, J. Comput. Appl. Math., 115 (2000), pp. 349–355.
- [12] B. LI, L. LI, M. MARADA, H. NIKI, AND M. J. TSATSOMEROS, *An iterative criterion for H -matrices*, Linear Algebra Appl., 271 (1998), pp. 179–190.
- [13] B. LI AND M. TSATSOMEROS, *Doubly diagonally dominant matrices*, Linear Algebra Appl., 261 (1997), pp. 221–235.
- [14] L. LI, H. NIKI, AND M. SASANABE, *A nonparameter criterion for generalized diagonally dominant matrices*, Int. J. Comput. Math., 71 (1999), pp. 267–275.
- [15] R.-C. LI, *Relations between the Field of Values of a Matrix and Those of Its Schur Complements*, Technical Report UCB/CSD-94-849, EECS Department, University of California, Berkeley, CA, 1994; available online at <http://www.eecs.berkeley.edu/Pubs/TechRpts/1994/5874.html>.
- [16] J. LIU, Y. HUANG, AND F. ZHANG, *The Schur complements of generalized doubly diagonally dominant matrices*, Linear Algebra Appl., 378 (2004), pp. 231–244.
- [17] K. OJIO, H. NIKI, AND M. USUI, *A new criterion for the H -matrix property*, J. Comput. Appl. Math., 150 (2003), pp. 293–302.
- [18] L. MIRSKY, *The spread of a matrix*, Mathematika, 3 (1956), pp. 127–130.
- [19] R. C. THOMPSON, *Root spreads for polynomials and Hermitian matrix pencils*, Linear Algebra Appl., 220 (1995), pp. 419–433.
- [20] F.-Z. ZHANG, *Matrix Theory: Basic Results and Techniques*, Springer, New York, 1999.
- [21] F.-Z. ZHANG, ED., *The Schur Complement and Its Applications*, Springer-Verlag, New York, 2005.

BEST APPROXIMATE SOLUTION OF MATRIX EQUATION

$$AXB + CYD = E^*$$

AN-PING LIAO[†], ZHONG-ZHI BAI[‡], AND YUAN LEI[†]

Abstract. We derive an analytical expression of the best approximate solution in the least-squares solution set $\mathbb{S}_E$ of the matrix equation $AXB + CYD = E$ to a given matrix pair (X_f, Y_f) , where A, B, C, D , and E are given matrices of suitable sizes. Our work is based on the projection theorem in the finite-dimensional inner product space, and we use the generalized singular value decomposition and the canonical correlation decomposition. Moreover, we establish a direct method for computing this best approximate solution. An algorithm for finding the best approximate solution is described in detail, and an example is used to show the feasibility and effectiveness of our algorithm.

Key words. matrix equation, least-squares solution, best approximate solution, generalized singular value decomposition, canonical correlation decomposition

AMS subject classifications. 15A14, 65F20, 65F22, 65K10

DOI. 10.1137/040615791

1. Introduction. Research on solving linear matrix equations has been actively ongoing for the past 50 years. For instance, Roth [21] derived a condition for the existence of a solution of the matrix equation $AX - YB = C$, where A, B , and C are given matrices of suitable sizes. Baksalary and Kala [1] presented a condition for the existence of a solution and derived a formula for the general solution of the matrix equation

$$(1.1) \quad AXB + CYD = E,$$

where A, B, C, D , and E are given matrices of suitable sizes defined over the real number field.

Many results have been obtained about the matrix equation (1.1). For example, Chu [4] gave the consistency conditions and the minimum-norm solution by making use of the generalized singular value decomposition (GSVD). Huang and Zeng [12] and Özgüler [25], respectively, gave the solvability conditions over a simple Artinian ring and a principal ideal domain by using the generalized inverse. However, we should point out that if the matrices A, B, C, D , and E are experimentally occurring in practice, they may not satisfy these solvability conditions because of the inconsistency of the matrix equation (1.1). Hence, we need to further study the least-squares solution

*Received by the editors September 26, 2004; accepted for publication (in revised form) by L. Elden June 10, 2005; published electronically December 7, 2005.

<http://www.siam.org/journals/simax/27-3/61579.html>

[†]College of Mathematics and Econometrics, Hunan University, Changsha 410082, People's Republic of China (liaoap@hnu.cn, e-ly1019@sina.com.cn). The work of the first author was supported in part by Natural Science Foundation of Hunan Province grant 03JJY6028, People's Republic of China.

[‡]Department of Mathematics, Fudan University, Shanghai 200433, People's Republic of China and State Key Laboratory of Scientific/Engineering Computing, Institute of Computational Mathematics and Scientific/Engineering Computing, Academy of Mathematics and Systems Science, Chinese Academy of Sciences, P.O. Box 2719, Beijing 100080, People's Republic of China (bzz@lsec.cc.ac.cn). The work of this author was subsidized by The Special Funds for Major State Basic Research Projects (grants G1999032803 and 2005CB321702), by China NNSF Outstanding Young Scientist Foundation grant 10525102, and by National Natural Science Foundation grant 10471146, People's Republic of China.

of the matrix equation (1.1). Xu, Wei, and Zheng [27] gave the least-squares solution of the matrix equation (1.1) by making use of the canonical correlation decomposition (CCD). In addition, Shim and Chen [24] presented the least-squares solution with the minimum norm by using the singular value decomposition (SVD) and the GSVD [8]. To our knowledge, however, no result exists about the best approximate least-squares solution of the matrix equation (1.1). We refer to [15, 16, 17] for discussions about some special cases of the matrix equation (1.1).

The problem that finds a nearest matrix in the least-squares solution set of a matrix equation to a given matrix, in the sense of the Frobenius norm, is called a matrix nearness problem. This kind of problem was initially proposed in the processes of testing and recovering linear systems with incomplete data and revising data. Some preliminary estimations of the unknown matrices can be obtained by experimental observation values and statistical distribution information. The matrix nearness problem associated with the matrix equation (1.1) can be described as follows.

PROBLEM 1.1. *Given matrices $A \in \mathbb{R}^{m \times n}$, $B \in \mathbb{R}^{p \times q}$, $C \in \mathbb{R}^{m \times k}$, $D \in \mathbb{R}^{l \times q}$, $E \in \mathbb{R}^{m \times q}$, $X_f \in \mathbb{R}^{n \times p}$, and $Y_f \in \mathbb{R}^{k \times l}$, let*

$$(1.2) \quad \mathbb{S}_E = \{[X, Y] \mid X \in \mathbb{R}^{n \times p}, Y \in \mathbb{R}^{k \times l}, \|AXB + CYD - E\|_F = \min\}.$$

Then, find $[X^*, Y^*] \in \mathbb{S}_E$ such that

$$\|[X^* - X_f, Y^* - Y_f]\| = \min_{[X, Y] \in \mathbb{S}_E} \|[X - X_f, Y - Y_f]\|,$$

where $\|\cdot\|_F$ denotes the Frobenius norm and $\|\cdot\|$ is defined by

$$\|[X, Y]\| = (\|X\|_F^2 + \|Y\|_F^2)^{\frac{1}{2}} \quad \forall [X, Y] \in \mathbb{R}^{n \times p} \times \mathbb{R}^{k \times l}.$$

It turns out that $\|[X, Y]\| \equiv \|[X, Y]\|_F$.

This problem may arise in many areas of scientific computing and in engineering applications. In section 2 we present a specific example to show a motivation for studying Problem 1.1.

Obviously, $\mathbb{S}_E$ defined in (1.2) is exactly the least-squares solution set of the matrix equation (1.1), and $[X^*, Y^*]$ is the best approximate least-squares solution to the given matrix pair $[X_f, Y_f]$. In particular, when $X_f = 0$ and $Y_f = 0$, the solution $[X^*, Y^*]$ of Problem 1.1 is just the least-squares solution with the minimum norm of the matrix equation (1.1); see [24].

There has been much discussion on the matrix nearness problem associated with other matrix equations, too. For details we refer to [10, 14, 20, 28, 29, 30, 18] and references therein. According to [4, 27], the solution of Problem 1.1 cannot be obtained by only applying GSVD and CCD. The difficulty is due to the fact that the invariance of the Frobenius norm holds only for orthogonal matrices but does not hold for nonsingular matrices that appear in GSVD and CCD used in [4, 27]. For details, see, for instance, (12) in [4] and (2.10) in [27].

In order to overcome the above-mentioned difficulty, in this paper we adopt a different approach from that used in [24] for solving Problem 1.1. Our approach is based on the projection theorem in an inner product space, and also based on GSVD and CCD of matrices. Specifically, it can be essentially divided into three parts. First, we find a least-squares solution $[X_0, Y_0]$ of the matrix equation (1.1) by using CCD; then by utilizing the solution $[X_0, Y_0]$ and the projection theorem, we transform Problem 1.1 into a problem of finding the best approximate solution of

a consistent matrix equation; finally, we find the best approximate solution of this consistent matrix equation by GSVD.

This paper is organized as follows. After describing a practical background from finite element model updating about Problem 1.1 in section 2, we introduce some necessary notation and concepts as well as several useful lemmas in section 3. An analytical expression of the solution of Problem 1.1 is derived in section 4. In section 5 we give an algorithm for computing the solution of Problem 1.1 and use an example to show feasibility and effectiveness of this algorithm. Finally, in section 6 we give some concluding remarks.

2. A practical background. *Finite element model updating* (FEMU) emerged in the 1990s as a significant subject in the design, construction, and maintenance of mechanical systems and civil engineering structures and has become a common and important method for improving the correlation between finite element models and measured data; see [7]. It is this class of FEMU problems that may present bounteous resources for Problem 1.1.

As is known, vibrating structures such as automobiles, bridges, highways, and buildings, etc., can often be characterized by a distributed parameter system of the form

$$(2.1) \quad \mathbf{M}(\mathbf{x}) \frac{\partial^2 \mathbf{u}(\mathbf{t}, \mathbf{x})}{\partial \mathbf{t}^2} + (\mathbf{D}(\mathbf{x}) + \mathbf{G}(\mathbf{x})) \frac{\partial \mathbf{u}(\mathbf{t}, \mathbf{x})}{\partial \mathbf{t}} + \mathbf{K}(\mathbf{x}) \mathbf{u}(\mathbf{t}, \mathbf{x}) = 0,$$

where $\mathbf{M}$, $\mathbf{D}$, $\mathbf{G}$, and $\mathbf{K}$ are given differential operators in the $\mathbf{x}$ -domain (spatial domain) of the displacement function $\mathbf{u}(\mathbf{t}, \mathbf{x})$, and where $\mathbf{u}(\mathbf{t}, \mathbf{x})$ for all $\mathbf{t}$ within the given time domain belong to some Hilbert space $\mathcal{H}$ that accounts for the boundary conditions of (2.1). The operators $\mathbf{M}$, $\mathbf{K}$, $\mathbf{D}$, and $\mathbf{G}$ are, respectively, mass, stiffness, damping, and gyroscopic operators. In many applications, $\mathbf{M}$ is self-adjoint and positive definite, $\mathbf{D}$ is self-adjoint, and $\mathbf{G}$ is skew-symmetric.

Though it is desirable to solve a vibration problem in its natural distributed-parameter settings, very often in practice such a system (2.1) is discretized to a matrix second-order system of the form

$$(2.2) \quad M_a \ddot{u}(\mathbf{t}) + C_a \dot{u}(\mathbf{t}) + K_a u(\mathbf{t}) = 0,$$

with $C_a = D_a + G_a$, by the finite element technique, because of the lack of viable and effective numerical methods to directly handle a distributed-parameter system; see [5]. Such a matrix second-order system (2.2) in vibration literature is called a *finite element model* (FEM). The matrices M_a , K_a , D_a , and G_a are known as mass, stiffness, damping, and gyroscopic matrices, respectively, and $\dot{u}(\mathbf{t})$ and $\ddot{u}(\mathbf{t})$ denote, respectively, the first and the second derivatives of the time-dependent vector $u(\mathbf{t})$. In practice, M_a , K_a , D_a , and G_a are large sparse real matrices, with, possibly, M_a being symmetric positive definite, D_a being symmetric, and G_a skew-symmetric.

By separation of variables [13], the FEM (2.2) leads to the *quadratic eigenvalue problem* (QEP): Find a complex number λ and a nonzero vector u such that

$$(2.3) \quad (\lambda^2 M_a + \lambda C_a + K_a)u = 0.$$

When a few frequencies and mode shapes of the FEM (2.2), or eigenvalues and eigenvectors of the QEP (2.3), are experimentally measured from a realized practical structure, these measured eigenvalues and eigenvectors (or frequencies and mode shapes) are then compared with those of the QEP (or FEM). Frequently, there is a

disagreement between these two sets. One then needs to tackle the correspondingly induced FEMU problem: Update the existing FEM with minimal changes so that the updated FEM has the following properties:

- (i) The measured eigenvalues and eigenvectors are also the eigenvalues and eigenvectors of the FEM;
- (ii) the remaining eigenvalues and eigenvectors remain unchanged; and
- (iii) the exploitable properties such as symmetry, definiteness, sparsity, and bandedness, etc. of the original model are preserved.

If the difference between the initial and the updated damping and stiffness matrices is minimized, subject to the constraints that the eigenvalue equation is satisfied and that the stiffness matrix is real, symmetric, and positive definite, then the penalty function to be minimized is given by

$$(2.4) \quad \mathcal{J}[K_u, C_u] := \frac{1}{2} \|N^{-1}(K_u - K_a)N^{-1}\|_F^2 + \frac{1}{2} \mu \|N^{-1}(C_u - C_a)N^{-1}\|_F^2,$$

and the constraints are given by

$$(2.5) \quad M_a \Phi \Lambda^2 + C_u \Phi \Lambda + K_u \Phi = 0,$$

where $N = M_a^{\frac{1}{2}}$; μ is a weighting parameter; M_a , K_a , and C_a are the initial analytical mass, stiffness, and damping matrices; K_u and C_u are the updated mass and stiffness matrices; and Φ and Λ are the measured eigenvector and eigenvalue matrices. See [2, 3, 6] for more details.

It may happen that the matrix equation (2.5) is inconsistent due to inaccuracies in the measured eigenvalues and eigenvectors. In this case, we may consider a solution of (2.5) in the least-squares sense; i.e., instead of (2.5) we adopt the constraint

$$(2.6) \quad \|M_a \Phi \Lambda^2 + C_u \Phi \Lambda + K_u \Phi\|_F = \min.$$

Now, with the matrix transformations

$$\begin{cases} X &= \frac{\sqrt{2}}{2} N^{-1} K_u N^{-1}, & X_f &= \frac{\sqrt{2}}{2} N^{-1} K_a N^{-1}, \\ Y &= \frac{\sqrt{2\mu}}{2} N^{-1} C_u N^{-1}, & Y_f &= \frac{\sqrt{2\mu}}{2} N^{-1} C_a N^{-1}, \end{cases}$$

we can equivalently express the above FEMU problem, i.e., (2.4) and (2.6), in the form of Problem 1.1, with

$$A = \sqrt{2}N, \quad B = N\Phi, \quad C = \frac{\sqrt{2\mu}}{\mu}N, \quad D = N\Phi\Lambda, \quad E = -M_a\Phi\Lambda^2.$$

3. Notation and lemmas. The set of $n \times n$ orthogonal matrices in $\mathbb{R}^{n \times n}$ is denoted by $\mathbb{O}^{n \times n}$, and I is used to represent the identity matrix. For a matrix $A \in \mathbb{R}^{n \times n}$, we denote its transpose, rank, and trace by A^T , $\text{rank}(A)$, and $\text{tr}(A)$, respectively. For two matrices $A = (a_{ij}) \in \mathbb{R}^{m \times n}$ and $B = (b_{ij}) \in \mathbb{R}^{m \times n}$, $A \circ B$ represents the Hadamard product of A and B , i.e., $A \circ B = (a_{ij}b_{ij})_{m \times n}$, and $\langle A, B \rangle$ represents the inner product of A and B , i.e., $\langle A, B \rangle = \text{tr}(B^T A)$; see [11]. Then we easily see that $\mathbb{R}^{m \times n}$ is an inner product space equipped with the Frobenius norm of matrices, which is induced from the inner product $\langle A, B \rangle$.

We first state the concepts of CCD and GSVD, which are essential tools for deriving the solution of Problem 1.1. See [19, 22, 4, 23, 9, 27] for details.

Let $A \in \mathbb{R}^{m \times n}$ and $C \in \mathbb{R}^{m \times k}$ satisfy $\text{rank}(A) \geq \text{rank}(C)$. Then the CCD of the matrix pair (A, C) is given by

$$(3.1) \quad A = U(\Sigma_A, 0)E_A^{-1} \quad \text{and} \quad C = U(\Sigma_C, 0)E_C^{-1},$$

where $U \in \mathbb{O}^{m \times m}$, $E_A \in \mathbb{R}^{n \times n}$, and $E_C \in \mathbb{R}^{k \times k}$ are nonsingular matrices, and

$$\Sigma_A = \begin{pmatrix} I_{r_1} & 0 & 0 \\ 0 & \Theta_A & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & \Omega_A & 0 \\ 0 & 0 & I_{t_1} \end{pmatrix}, \quad \Sigma_C = \begin{pmatrix} I_{h_1} \\ 0 \end{pmatrix}$$

are block matrices, with the diagonal matrices Θ_A and Ω_A given by

$$\begin{cases} \Theta_A &= \text{diag}(\theta_1^{(A)}, \theta_2^{(A)}, \dots, \theta_{s_1}^{(A)}) > 0, \\ \Omega_A &= \text{diag}(\omega_1^{(A)}, \omega_2^{(A)}, \dots, \omega_{s_1}^{(A)}) > 0. \end{cases}$$

Here,

$$\begin{cases} r_1 &= \text{rank}(A) + \text{rank}(C) - \text{rank}(A, C), \\ s_1 &= \text{rank}(A, C) + \text{rank}(C^T A) - \text{rank}(A) - \text{rank}(C), \\ t_1 &= \text{rank}(A) - \text{rank}(C^T A), \\ h_1 &= \text{rank}(C). \end{cases}$$

See [9, 23, 27] for details. We further partition the orthogonal matrix

$$(3.2) \quad U = (U_1, U_2, U_3, U_4, U_5, U_6)$$

compatibly with the block row partitioning of Σ_A ; i.e., it holds that

$$\begin{cases} U_1 \in \mathbb{R}^{m \times r_1}, & U_2 \in \mathbb{R}^{m \times s_1}, & U_3 \in \mathbb{R}^{m \times (h_1 - r_1 - s_1)}, \\ U_4 \in \mathbb{R}^{m \times (m - h_1 - s_1 - t_1)}, & U_5 \in \mathbb{R}^{m \times s_1}, & U_6 \in \mathbb{R}^{m \times t_1}. \end{cases}$$

The GSVD of the matrix pair (A^T, C^T) is given by

$$A^T = U_A \Pi_A M, \quad C^T = V_C \Pi_C M,$$

where $U_A \in \mathbb{O}^{n \times n}$ and $V_C \in \mathbb{O}^{k \times k}$; $M \in \mathbb{R}^{m \times m}$ is a nonsingular matrix; and

$$\Pi_A = \begin{pmatrix} I_r & 0 & 0 & 0 \\ 0 & \Lambda_A & 0 & 0 \\ 0 & 0 & 0_{(n-g) \times (t-g)} & 0 \end{pmatrix}, \quad \Pi_C = \begin{pmatrix} 0_{(k-t+r) \times r} & 0 & 0 & 0 \\ 0 & \Xi_C & 0 & 0 \\ 0 & 0 & I_{(t-g)} & 0 \end{pmatrix}$$

are block matrices, with the diagonal matrices Λ_A and Ξ_C given by

$$\begin{cases} \Lambda_A &= \text{diag}(\lambda_1^{(A)}, \lambda_2^{(A)}, \dots, \lambda_s^{(A)}) > 0, \\ \Xi_C &= \text{diag}(\xi_1^{(C)}, \xi_2^{(C)}, \dots, \xi_s^{(C)}) > 0. \end{cases}$$

Here,

$$\begin{cases} g &= \text{rank}(A), \\ r &= \text{rank}(A, C) - \text{rank}(C), \\ s &= \text{rank}(A) + \text{rank}(C) - \text{rank}(A, C), \\ t &= \text{rank}(A, C). \end{cases}$$

See [19, 4] for details. We further partition the nonsingular matrix

$$(3.3) \quad M^{-1} = (\tilde{M}_1, \tilde{M}_2, \tilde{M}_3, \tilde{M}_4)$$

compatibly with the block column partitioning of Π_A ; i.e., it holds that

$$\tilde{M}_1 \in \mathbb{R}^{m \times r}, \quad \tilde{M}_2 \in \mathbb{R}^{m \times s}, \quad \tilde{M}_3 \in \mathbb{R}^{m \times (t-g)}, \quad \tilde{M}_4 \in \mathbb{R}^{m \times (m-t)}.$$

The following lemmas are important for deriving an analytical formula of the solution of Problem 1.1.

LEMMA 3.1 (the projection theorem [26]). *Let $\mathbb{X}$ be an inner product space, $\mathbb{M}$ be a subspace of $\mathbb{X}$, and $\mathbb{M}^\perp$ be the orthogonal complement subspace of $\mathbb{M}$. For a given $x \in \mathbb{X}$, if there exists an $m_0 \in \mathbb{M}$ such that $\|x - m_0\| \leq \|x - m\|$ holds for any $m \in \mathbb{M}$, then m_0 is unique and $m_0 \in \mathbb{M}$ is the unique minimization vector in $\mathbb{M}$ if and only if $(x - m_0) \perp \mathbb{M}$, i.e., $(x - m_0) \in \mathbb{M}^\perp$.*

LEMMA 3.2. *Given matrices $\Phi = (\phi_{ij}) \in \mathbb{R}^{p \times q}$, $\Psi = (\psi_{ij}) \in \mathbb{R}^{p \times q}$, and $\Upsilon = (v_{ij}) \in \mathbb{R}^{p \times q}$, let*

$$\Theta = \text{diag}(\theta_1, \theta_2, \dots, \theta_p), \quad \Omega = \text{diag}(\omega_1, \omega_2, \dots, \omega_p)$$

and

$$\Gamma = \text{diag}(\gamma_1, \gamma_2, \dots, \gamma_q), \quad \Delta = \text{diag}(\delta_1, \delta_2, \dots, \delta_q)$$

be diagonal matrices of positive diagonal entries. Assume that Θ and Γ are nonsingular. Then there exists a unique matrix $\tilde{X} = (\tilde{x}_{ij}) \in \mathbb{R}^{p \times q}$ such that

$$\varphi(\tilde{X}) \equiv \|\Theta^{-1}(\Phi - \Omega\tilde{X}\Delta)\Gamma^{-1} - \Psi\|_F^2 + \|\tilde{X} - \Upsilon\|_F^2 = \min.$$

Moreover, the matrix $\tilde{X}$ possesses the analytical expression

$$(3.4) \quad \tilde{X} = K \circ [\Omega\Phi\Delta + \Theta^2\Upsilon\Gamma^2 - \Theta\Omega\Psi\Gamma\Delta],$$

where $K = (k_{ij}) \in \mathbb{R}^{p \times q}$ is defined by

$$k_{ij} = \frac{1}{\theta_i^2 \gamma_j^2 + \omega_i^2 \delta_j^2}, \quad i = 1, 2, \dots, p, \quad j = 1, 2, \dots, q.$$

Proof. We have

$$\varphi(\tilde{X}) = \sum_{i,j} \left[\left(\frac{\omega_i \delta_j}{\theta_i \gamma_j} \tilde{x}_{ij} - \frac{\phi_{ij}}{\theta_i \gamma_j} + \psi_{ij} \right)^2 + (\tilde{x}_{ij} - v_{ij})^2 \right].$$

Because $\varphi(\tilde{X})$ is a convex, continuous, and differentiable function with respect to the pq variables $\tilde{x}_{ij}$ ($i = 1, 2, \dots, p$, $j = 1, 2, \dots, q$), we easily see that $\varphi(\tilde{X}) = \min$ if and only if $\frac{\partial \varphi(\tilde{X})}{\partial \tilde{X}} = 0$. It then follows immediately from direct computations that

$$(3.5) \quad \tilde{x}_{ij} = \frac{\omega_i \phi_{ij} \delta_j + \theta_i^2 v_{ij} \gamma_j^2 - \theta_i \omega_i \psi_{ij} \gamma_j \delta_j}{\theta_i^2 \gamma_j^2 + \omega_i^2 \delta_j^2}, \quad i = 1, 2, \dots, p, \quad j = 1, 2, \dots, q.$$

By rewriting (3.5) in matrix form, we obtain (3.4). $\square$

4. The solution of Problem 1.1. In this section, we derive an analytical expression for the solution of Problem 1.1. Without loss of generality, we consider only the case

$$\text{rank}(A) \geq \text{rank}(C) \quad \text{and} \quad \text{rank}(B) \geq \text{rank}(D),$$

as all other possible cases can be treated in an analogous fashion with slight technical modifications.

To this end, we first transform the least-squares problem (1.2) with respect to the matrix equation (1.1) into a consistent matrix equation. This technique is precisely described in the following theorem.

THEOREM 4.1. *Given matrices $A \in \mathbb{R}^{m \times n}$, $B \in \mathbb{R}^{p \times q}$, $C \in \mathbb{R}^{m \times k}$, $D \in \mathbb{R}^{l \times q}$, and $E \in \mathbb{R}^{m \times q}$, let (X_0, Y_0) be one of the least-squares solutions of the matrix equation (1.1) and define*

$$(4.1) \quad E_0 = AX_0B + CY_0D.$$

Then the matrix equation

$$(4.2) \quad AXB + CYD = E_0$$

is consistent, and its solution set is the same as the least-squares solution set $\mathbb{S}_E$ of the matrix equation (1.1).

Proof. Let

$$\mathbb{S} = \{Z \mid Z = AXB + CYD, \quad X \in \mathbb{R}^{n \times p}, \quad Y \in \mathbb{R}^{k \times l}\}.$$

Then $\mathbb{S}$ is obviously a linear subspace of $\mathbb{R}^{m \times q}$. Because (X_0, Y_0) is a least-squares solution of the matrix equation (1.1), from (4.1) we see that $E_0 \in \mathbb{S}$ and

$$\begin{aligned} \|E_0 - E\|_F &= \|AX_0B + CY_0D - E\|_F \\ &= \min_{[X, Y] \in \mathbb{R}^{n \times p} \times \mathbb{R}^{k \times l}} \|AXB + CYD - E\|_F \\ &= \min_{Z \in \mathbb{S}} \|Z - E\|_F. \end{aligned}$$

Now, by Lemma 3.1 we have

$$(E_0 - E) \perp \mathbb{S} \quad \text{or} \quad (E_0 - E) \in \mathbb{S}^\perp.$$

For all $[X, Y] \in \mathbb{R}^{n \times p} \times \mathbb{R}^{k \times l}$, we know that

$$(AXB + CYD - E_0) \in \mathbb{S}.$$

It then follows that

$$\begin{aligned} \|AXB + CYD - E\|_F^2 &= \|(AXB + CYD - E_0) + (E_0 - E)\|_F^2 \\ &= \|AXB + CYD - E_0\|_F^2 + \|E_0 - E\|_F^2. \end{aligned}$$

Hence, the conclusion of this theorem holds true. $\square$

From Theorem 4.1, we easily see that the best approximate solution (X^*, Y^*) of the consistent matrix equation (4.2) to a given matrix pair (X_f, Y_f) is just the solution of Problem 1.1. Therefore, solving Problem 1.1 essentially reduces to finding E_0 or a least-squares solution (X_0, Y_0) of the matrix equation (1.1). In order to find such

an E_0 , we need to utilize the CCD of the matrix pair (B^T, D^T) , which is precisely described in the following.

Let matrices $B \in \mathbb{R}^{p \times q}$ and $D \in \mathbb{R}^{l \times q}$ satisfy $\text{rank}(B) \geq \text{rank}(D)$. Then the CCD of the matrix pair (B^T, D^T) is given by

$$(4.3) \quad B^T = V(\Sigma_B, 0)E_B^{-1}, \quad D^T = V(\Sigma_D, 0)E_D^{-1},$$

where $V \in \mathbb{O}^{q \times q}$; $E_B \in \mathbb{R}^{p \times p}$ and $E_D \in \mathbb{R}^{l \times l}$ are two nonsingular matrices; and

$$\Sigma_B = \begin{pmatrix} I_{r_2} & 0 & 0 \\ 0 & \Theta_B & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & \Omega_B & 0 \\ 0 & 0 & I_{t_2} \end{pmatrix}, \quad \Sigma_D = \begin{pmatrix} I_{h_2} \\ 0 \end{pmatrix}$$

are block matrices with diagonal matrices Θ_B and Ω_B given by

$$\begin{cases} \Theta_B &= \text{diag}(\theta_1^{(B)}, \theta_2^{(B)}, \dots, \theta_{s_2}^{(B)}) > 0, \\ \Omega_B &= \text{diag}(\omega_1^{(B)}, \omega_2^{(B)}, \dots, \omega_{s_2}^{(B)}) > 0. \end{cases}$$

See [9, 23, 27] and (3.1) for details. We further partition the orthogonal matrix

$$V = (V_1, V_2, V_3, V_4, V_5, V_6)$$

compatibly with the block row partitioning of Σ_B ; i.e., it holds that

$$\begin{cases} V_1 \in \mathbb{R}^{q \times r_2}, & V_2 \in \mathbb{R}^{q \times s_2}, & V_3 \in \mathbb{R}^{q \times (h_2 - r_2 - s_2)}, \\ V_4 \in \mathbb{R}^{q \times (q - h_2 - s_2 - t_2)}, & V_5 \in \mathbb{R}^{q \times s_2}, & V_6 \in \mathbb{R}^{q \times t_2}. \end{cases}$$

Denote

$$(4.4) \quad U^T E V = (\hat{E}_{ij})_{6 \times 6} \quad \text{with} \quad \hat{E}_{ij} = U_i^T E V_j, \quad i, j = 1, 2, \dots, 6,$$

where the matrices U_i ($i = 1, 2, \dots, 6$) are given by (3.2) and E is the right-hand-side matrix of the matrix equation (1.1).

Based on the CCDs of the matrix pairs (A, C) and (B^T, D^T) , we can give a matrix E_0 that is induced from a least-squares solution (X_0, Y_0) of the matrix equation (1.1) and satisfies (4.1).

THEOREM 4.2. *Let matrices A, B, C, D , and E given in Problem 1.1 satisfy*

$$\text{rank}(A) \geq \text{rank}(C) \quad \text{and} \quad \text{rank}(B) \geq \text{rank}(D).$$

Then for any least-squares solution (X_0, Y_0) of the matrix equation (1.1), the matrix E_0 defined in (4.1) can be expressed in the following form:

$$(4.5) \quad E_0 = U \begin{pmatrix} \hat{E}_{11} & \hat{E}_{12} & \hat{E}_{13} & 0 & \hat{E}_{15} & \hat{E}_{16} \\ \hat{E}_{21} & \hat{E}_{22} & \hat{E}_{23} & 0 & \Theta_A X_{22} \Omega_B & \Theta_A^2 \hat{E}_{26} + \Theta_A \Omega_A \hat{E}_{56} \\ \hat{E}_{31} & \hat{E}_{32} & \hat{E}_{33} & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ \hat{E}_{51} & \Omega_A X_{22} \Omega_B & 0 & 0 & \Omega_A X_{22} \Omega_B & \Omega_A \Theta_A \hat{E}_{26} + \Omega_A^2 \hat{E}_{56} \\ \hat{E}_{61} & \hat{E}_{62} \Theta_B^2 + \hat{E}_{65} \Omega_B \Theta_B & 0 & 0 & \hat{E}_{62} \Theta_B \Omega_B + \hat{E}_{65} \Omega_B^2 & \hat{E}_{66} \end{pmatrix} V^T,$$

where

$$(4.6) \quad X_{22} = W \circ (\Theta_A \hat{E}_{25} \Omega_B + \Omega_A \hat{E}_{55} \Omega_B + \Omega_A \hat{E}_{52} \Theta_B),$$

with

$$W = (w_{ij}) \in \mathbb{R}^{s_1 \times s_2}, \quad w_{ij} = \frac{1}{(\omega_j^{(B)})^2 + (\omega_i^{(A)})^2 (\theta_j^{(B)})^2},$$

$$i = 1, 2, \dots, s_1, \quad j = 1, 2, \dots, s_2.$$

Proof. From Theorem 2.1 in [27] we know that the least-squares solutions of the matrix equation (1.1) can be given by using CCDs of the matrix pairs (A, C) and (B^T, D^T) and are of the following expressions:

$$(4.7) \quad \begin{cases} X = E_A \begin{pmatrix} \hat{E}_{11} - Y_{11} & \hat{E}_{15} \Omega_B^{-1} & \hat{E}_{16} & X_{14} \\ \Omega_A^{-1} \hat{E}_{51} & X_{22} & \Theta_A \hat{E}_{26} + \Omega_A \hat{E}_{56} & X_{24} \\ \hat{E}_{61} & \hat{E}_{62} \Theta_B + \hat{E}_{65} \Omega_B & \hat{E}_{66} & X_{34} \\ X_{41} & X_{42} & X_{43} & X_{44} \end{pmatrix} E_B^T, \\ Y = E_C \begin{pmatrix} Y_{11} & \hat{E}_{12} - \hat{E}_{15} \Omega_B^{-1} \Theta_B & \hat{E}_{13} & Y_{14} \\ \hat{E}_{21} - \Theta_A \Omega_A^{-1} \hat{E}_{51} & \hat{E}_{22} - \Theta_A X_{22} \Theta_B & \hat{E}_{23} & Y_{24} \\ \hat{E}_{31} & \hat{E}_{32} & \hat{E}_{33} & Y_{34} \\ Y_{41} & Y_{42} & Y_{43} & Y_{44} \end{pmatrix} E_D^T, \end{cases}$$

where the matrix block X_{22} is defined by (4.6); X_{i4} , X_{4j} , Y_{11} , Y_{i4} , and Y_{4j} , $i, j = 1, 2, 3, 4$, are arbitrary matrix blocks.

By substituting (3.1), (4.3), and (4.7) into (4.1), after concrete manipulations we can immediately obtain (4.5). $\square$

Evidently, (4.5) shows that the matrix E_0 given in Theorem 4.2 is dependent only on the matrices A , B , C , D , and E but is independent of the least-squares solution (X_0, Y_0) of the matrix equation (1.1). Therefore, we can conclude that

$$\|E_0 - E\|_F = \min_{[X, Y] \in \mathbb{R}^{n \times p} \times \mathbb{R}^{k \times l}} \|AXB + CYD - E\|_F.$$

To derive the solution of Problem 1.1, we need to use the GSVD of the matrix pair (B, D) with $B \in \mathbb{R}^{p \times q}$ and $D \in \mathbb{R}^{l \times q}$. More precisely, the GSVD of the matrix pair (B, D) is given by

$$B = U_B \Pi_B N, \quad D = V_D \Pi_D N,$$

where $U_B \in \mathbb{O}^{p \times p}$ and $V_D \in \mathbb{O}^{l \times l}$; $N \in \mathbb{R}^{q \times q}$ is a nonsingular matrix; and

$$\Pi_B = \begin{pmatrix} I_{r'} & 0 & 0 & 0 \\ 0 & \Lambda_B & 0 & 0 \\ 0 & 0 & 0_{(p-g') \times (t'-g')} & 0 \end{pmatrix}, \quad \Pi_D = \begin{pmatrix} 0_{(l-t'+r') \times r'} & 0 & 0 & 0 \\ 0 & \Xi_D & 0 & 0 \\ 0 & 0 & I_{(t'-g')} & 0 \end{pmatrix}$$

are block matrices with the diagonal matrices Λ_B and Ξ_D given by

$$\begin{cases} \Lambda_B &= \text{diag}(\lambda_1^{(B)}, \lambda_2^{(B)}, \dots, \lambda_{s'}^{(B)}) > 0, \\ \Xi_D &= \text{diag}(\xi_1^{(D)}, \xi_2^{(D)}, \dots, \xi_{s'}^{(D)}) > 0. \end{cases}$$

See [19, 4] and (3.3) for details. We further partition the nonsingular matrix

$$(4.8) \quad N^{-1} = (\tilde{N}_1, \tilde{N}_2, \tilde{N}_3, \tilde{N}_4)$$

compatibly with the block column partitioning of Π_B ; i.e., it holds that

$$\tilde{N}_1 \in \mathbb{R}^{q \times r'}, \quad \tilde{N}_2 \in \mathbb{R}^{q \times s'}, \quad \tilde{N}_3 \in \mathbb{R}^{q \times (t' - g')}, \quad \tilde{N}_4 \in \mathbb{R}^{q \times (q - t')}.$$

Denote

$$(4.9) \quad M^{-T} E_0 N^{-1} = (\bar{E}_{ij}) \quad \text{with} \quad \bar{E}_{ij} = \tilde{M}_i^T E_0 \tilde{N}_j, \quad i, j = 1, 2, 3, 4,$$

where $\tilde{M}_i (i = 1, 2, 3, 4)$ and E_0 are given by (3.3) and (4.5), respectively.

Based on Theorems 4.1 and 4.2, we can obtain the analytical expression of the solution of Problem 1.1 using the GSVDs of the matrix pairs (A^T, C^T) and (B, D) .

THEOREM 4.3. *Let matrices A, B, C, D , and E given in Problem 1.1 satisfy*

$$\text{rank}(A) \geq \text{rank}(C) \quad \text{and} \quad \text{rank}(B) \geq \text{rank}(D).$$

Partition the matrices $U_A^T X_f U_B$ and $V_C^T Y_f V_D$ compatibly to the block row partitioning of Π_A, Π_B and Π_C, Π_D , respectively, into

$$(4.10) \quad U_A^T X_f U_B = (X_{ij}^{(f)})_{3 \times 3} \quad \text{and} \quad V_C^T Y_f V_D = (Y_{ij}^{(f)})_{3 \times 3}.$$

Then the unique solution (X^*, Y^*) of Problem 1.1 can be expressed as

$$(4.11) \quad \begin{cases} X^* = U_A \begin{pmatrix} \bar{E}_{11} & \bar{E}_{12} \Lambda_B^{-1} & X_{13}^{(f)} \\ \Lambda_A^{-1} \bar{E}_{21} & X_{22}^* & X_{23}^{(f)} \\ X_{31}^{(f)} & X_{32}^{(f)} & X_{33}^{(f)} \end{pmatrix} U_B^T, \\ Y^* = V_C \begin{pmatrix} Y_{11}^{(f)} & Y_{12}^{(f)} & Y_{13}^{(f)} \\ Y_{21}^{(f)} & Y_{22}^* & \Xi_C^{-1} \bar{E}_{23} \\ Y_{31}^{(f)} & \bar{E}_{32} \Xi_D^{-1} & \bar{E}_{33} \end{pmatrix} V_D^T, \end{cases}$$

where

$$\begin{cases} X_{22}^* = \Lambda_A^{-1} (\bar{E}_{22} - \Xi_C Y_{22}^* \Xi_D) \Lambda_B^{-1}, \\ Y_{22}^* = F \circ (\Xi_C \bar{E}_{22} \Xi_D + \Lambda_A^2 Y_{22}^{(f)} \Lambda_B^2 - \Lambda_A \Xi_C X_{22}^{(f)} \Xi_D \Lambda_B), \end{cases}$$

with

$$F = (f_{ij}) \in \mathbb{R}^{s \times s'}, \quad f_{ij} = \frac{1}{(\lambda_i^{(A)})^2 (\lambda_j^{(B)})^2 + (\xi_i^{(C)})^2 (\xi_j^{(D)})^2}, \\ i = 1, 2, \dots, s, \quad j = 1, 2, \dots, s'.$$

Proof. From Theorems 4.1 and 4.2 we know that the least-squares solution set of the matrix equation (1.1) is the same as the solution set of the consistent matrix equation (4.2) with the matrix E_0 given by (4.5). From [4] we see that the solutions of the consistent matrix equation (4.2) can be expressed as

$$(4.12) \quad \begin{cases} X = U_A \begin{pmatrix} \bar{E}_{11} & \bar{E}_{12} \Lambda_B^{-1} & X_{13} \\ \Lambda_A^{-1} \bar{E}_{21} & X_{22} & X_{23} \\ X_{31} & X_{32} & X_{33} \end{pmatrix} U_B^T, \\ Y = V_C \begin{pmatrix} Y_{11} & Y_{12} & Y_{13} \\ Y_{21} & Y_{22} & \Xi_C^{-1} \bar{E}_{23} \\ Y_{31} & \bar{E}_{32} \Xi_D^{-1} & \bar{E}_{33} \end{pmatrix} V_D^T, \end{cases}$$

where

$$(4.13) \quad X_{22} = \Lambda_A^{-1}(\bar{E}_{22} - \Xi_C Y_{22} \Xi_D) \Lambda_B^{-1},$$

and the other matrix blocks X_{ij} and Y_{ij} , $i, j = 1, 2, 3, 4$, are arbitrary.

It follows from (4.10) and (4.12) that

$$\begin{aligned} \|[X - X^{(f)}, Y - Y^{(f)}]\|^2 &= \|X - X^{(f)}\|_F^2 + \|Y - Y^{(f)}\|_F^2 \\ &= \|U_A^T X U_B - U_A^T X^{(f)} U_B\|_F^2 + \|V_C^T Y V_D - V_C^T Y^{(f)} V_D\|_F^2 \\ &= \left\| \begin{pmatrix} \bar{E}_{11} - X_{11}^{(f)} & \bar{E}_{12} \Lambda_B^{-1} - X_{12}^{(f)} & X_{13} - X_{13}^{(f)} \\ \Lambda_A^{-1} \bar{E}_{21} - X_{21}^{(f)} & X_{22} - X_{22}^{(f)} & X_{23} - X_{23}^{(f)} \\ X_{31} - X_{31}^{(f)} & X_{32} - X_{32}^{(f)} & X_{33} - X_{33}^{(f)} \end{pmatrix} \right\|_F^2 \\ &\quad + \left\| \begin{pmatrix} Y_{11} - Y_{11}^{(f)} & Y_{12} - Y_{12}^{(f)} & Y_{13} - Y_{13}^{(f)} \\ Y_{21} - Y_{21}^{(f)} & Y_{22} - Y_{22}^{(f)} & \Xi_C^{-1} \bar{E}_{23} - Y_{23}^{(f)} \\ Y_{31} - Y_{31}^{(f)} & \bar{E}_{32} \Xi_D^{-1} - Y_{32}^{(f)} & \bar{E}_{33} - Y_{33}^{(f)} \end{pmatrix} \right\|_F^2. \end{aligned}$$

Hence,

$$\|[X - X^{(f)}, Y - Y^{(f)}]\| = \min_{\forall (X, Y) \in \mathbb{S}_E}$$

(see (1.2) for the definition of the matrix set $\mathbb{S}_E$) if and only if

$$\begin{cases} X_{13} = X_{13}^{(f)}, & X_{23} = X_{23}^{(f)}, \\ X_{31} = X_{31}^{(f)}, & X_{32} = X_{32}^{(f)}, & X_{33} = X_{33}^{(f)}, \\ Y_{11} = Y_{11}^{(f)}, & Y_{12} = Y_{12}^{(f)}, & Y_{13} = Y_{13}^{(f)}, \\ Y_{21} = Y_{21}^{(f)}, & Y_{31} = Y_{31}^{(f)}, \end{cases}$$

and

$$(4.14) \quad \|X_{22} - X_{22}^{(f)}\|_F^2 + \|Y_{22} - Y_{22}^{(f)}\|_F^2 = \min_{\forall Y_{22} \in \mathbb{R}^{s \times s'}}.$$

By using (4.13) and Lemma 3.2 we know that the solution of (4.14) is

$$Y_{22}^* = F \circ (\Xi_C \bar{E}_{22} \Xi_D + \Lambda_A^2 Y_{22}^{(f)} \Lambda_B^2 - \Lambda_A \Xi_C X_{22}^{(f)} \Xi_D \Lambda_B).$$

Now, after substituting Y_{22}^* into (4.13), we immediately get X_{22}^* . $\square$

5. Numerical verification. Based on Theorem 4.3, we can establish an algorithm for finding the solution of Problem 1.1 when

$$\text{rank}(A) \geq \text{rank}(C) \quad \text{and} \quad \text{rank}(B) \geq \text{rank}(D).$$

To this end, we let $A \in \mathbb{R}^{m \times n}$, $B \in \mathbb{R}^{p \times q}$, $C \in \mathbb{R}^{m \times k}$, $D \in \mathbb{R}^{l \times q}$, $E \in \mathbb{R}^{m \times q}$, and $X^{(f)} \in \mathbb{R}^{n \times p}$, $Y^{(f)} \in \mathbb{R}^{k \times l}$. Then the following algorithm computes the solution (X^*, Y^*) of Problem 1.1:

1. Input matrices A , B , C , D , E , and $X^{(f)}$, $Y^{(f)}$.

2. Make CCDs of the matrix pairs (A, C) and (B^T, D^T) , and partition the matrix $U^T E V = (\hat{E}_{ij})_{6 \times 6}$ by (4.4).
3. Compute the matrix X_{22} by (4.6).
4. Compute the matrix E_0 by (4.5).
5. Make GSVDs of the matrix pairs (A^T, C^T) and (B, D) , and partition the matrices M^{-1} and N^{-1} by (3.3) and (4.8), respectively.
6. Compute the matrix blocks $\bar{E}_{ij}$ ($i, j = 1, 2, 3, 4$) by (4.9).
7. Compute the solution (X^*, Y^*) of Problem 1.1 by (4.11).

Example 5.1. Let

$$\begin{aligned} A &= \begin{pmatrix} \text{hilb}(4) & \text{zeros}(4, 3) \\ \text{eye}(4) & \text{ones}(4, 3) \end{pmatrix}, & B &= \begin{pmatrix} \text{ones}(5, 5) & \text{zeros}(5, 4) \\ \text{zeros}(4, 5) & \text{pasc}(4) \end{pmatrix}, \\ C &= \begin{pmatrix} \text{magic}(4) \\ \text{ones}(4, 4) \end{pmatrix}, & D &= \begin{pmatrix} \text{hank}(1 : 4) & \text{zeros}(4, 5) \\ \text{zeros}(5, 4) & \text{zeros}(5, 5) \end{pmatrix}, \\ E &= (\text{toep}(1 : 8) \quad \text{ones}(8, 1)), \end{aligned}$$

and

$$X^{(f)} = (\text{eye}(7) \quad \text{zeros}(7, 2)), \quad Y^{(f)} = (\text{ones}(4, 5) \quad \text{zeros}(4, 4)),$$

where $\text{hilb}(n)$, $\text{pasc}(n)$, and $\text{magic}(n)$ denote Hilbert matrix, Pascal matrix, and Magic matrix of order n , respectively; and $\text{toep}(1 : n)$ and $\text{hank}(1 : n)$ denote Toeplitz matrix and Hankel matrix of order n , respectively, with their first rows being $(1, 2, \dots, n)$.

By using MATLAB 6.5, we obtain

$$E_0 = \begin{pmatrix} 1.7592 & 2.5500 & 3.0339 & 3.8110 & 4.3976 & 6.9218 & 8.4774 & 9.9590 & 1.4016 \\ 3.8154 & 2.4062 & 2.0901 & 2.6673 & 2.9698 & 4.3193 & 5.1198 & 5.9611 & 0.8807 \\ 1.6467 & 0.8375 & 0.9215 & 2.0986 & 2.2900 & 3.2241 & 3.7572 & 4.3494 & 0.6576 \\ 3.7029 & 2.6937 & 1.9777 & 0.9548 & 1.8746 & 2.5922 & 2.9892 & 3.4466 & 0.5283 \\ 2.9843 & 2.9127 & 2.8990 & 2.9434 & 3.0022 & 1.7791 & 2.6296 & 3.4990 & 0.8901 \\ 4.0431 & 3.9714 & 3.9577 & 4.0021 & 4.0609 & 1.0415 & 2.0675 & 3.0901 & 1.0189 \\ 5.0566 & 4.9849 & 4.9712 & 5.0156 & 5.0744 & 2.0861 & 1.1446 & 2.1958 & 1.0431 \\ 6.0601 & 5.9884 & 5.9748 & 6.0191 & 6.0780 & 3.0933 & 2.1582 & 1.2151 & 1.0479 \end{pmatrix}$$

and

$$\begin{aligned} X^* &= \begin{pmatrix} 1.0312 & 0.0312 & 0.0312 & 0.0312 & 0.0312 & 7.0270 & -13.9569 & 15.5052 & -5.2827 \\ 0.2430 & 1.2430 & 0.2430 & 0.2430 & 0.2430 & 5.6849 & -12.5166 & 14.2627 & -4.8759 \\ 0.4457 & 0.4457 & 1.4457 & 0.4457 & 0.4457 & 11.7990 & -21.7942 & 19.5768 & -5.9819 \\ 0.6464 & 0.6464 & 0.6464 & 1.6464 & 0.6464 & 5.8184 & -2.8446 & 2.6343 & -1.0013 \\ 0.0564 & 0.0564 & 0.0564 & 0.0564 & 1.0564 & -1.1941 & 1.0927 & -1.6649 & 0.5952 \\ 0.0564 & 0.0564 & 0.0564 & 0.0564 & 0.0564 & -0.1941 & 1.0927 & -1.6649 & 0.5952 \\ 0.0564 & 0.0564 & 0.0564 & 0.0564 & 0.0564 & -1.1941 & 2.0927 & -1.6649 & 0.5952 \end{pmatrix}, \\ Y^* &= \begin{pmatrix} -0.0209 & 0.0201 & -0.0056 & -0.0440 & 1.0000 & 0 & 0 & 0 & 0 \\ 0.0009 & -0.0057 & 0.0059 & 0.0266 & 1.0000 & 0 & 0 & 0 & 0 \\ -0.0117 & 0.0162 & 0.0082 & 0.0045 & 1.0000 & 0 & 0 & 0 & 0 \\ 0.0170 & -0.0454 & -0.0125 & 0.0224 & 1.0000 & 0 & 0 & 0 & 0 \end{pmatrix}. \end{aligned}$$

By concrete computations, we have

$$AX^*B + CY^*D = E_0$$

and

$$\min_{[X,Y] \in \mathbb{S}_E} \|[X - X^{(f)}, Y - Y^{(f)}]\| = \|[X^* - X^{(f)}, Y^* - Y^{(f)}]\| = 45.2607.$$

In addition, we also have

$$\min_{[X,Y] \in \mathbb{R}^{n \times p} \times \mathbb{R}^{k \times l}} \|AXB + CYD - E\|_F = \|E_0 - E\|_F = 8.4265$$

and

$$\langle E_0, E_0 - E \rangle = \text{tr}(E_0^T(E_0 - E)) = 6.7704 \times 10^{-13}.$$

Therefore, this example clearly shows that the above algorithm is feasible for solving Problem 1.1.

6. Concluding remarks. We have discussed a class of nearness problems in the least-squares solution set associated with the linear matrix equation $AXB + CYD = E$, where A, B, C, D , and E are given matrices of suitable sizes. A new algorithm has been presented based on the projection theorem and on GSVD and CCD for finding its solution. An example has been used to show the feasibility of our new algorithm.

In fact, the least-squares solution of a matrix equation with minimum norm may be found by simultaneously using both SVD and GSVD; see, e.g., [24]. The very powerful tools GSVD and CCD have also been individually used to find the solution of a linear matrix equation in the literature, but to our knowledge, there has been no simultaneous use of both GSVD and CCD to tackle a linear matrix equation. This work could be considered the first realization of this technical approach.

Acknowledgments. The authors are very much indebted to the referees for their constructive and valuable comments and suggestions which greatly improved this paper. The authors sincerely thank Professor Shu-Fang Xu for offering references [5, 6, 7, 13], which give a good motivation for the problem studied here. The second author thanks Professor Yang-Feng Su for his kind invitation to the Shanghai Key Laboratory of Contemporary Applied Mathematics of Fudan University in the winter of 2004.

REFERENCES

- [1] J. K. BAKSALARY AND R. KALA, *The matrix equation $AXB + CYD = E$* , Linear Algebra Appl., 30 (1980), pp. 141–147.
- [2] M. BARUCH, *Optimization procedures to correct stiffness and flexibility matrices using vibration data*, AIAA J., 16 (1978), pp. 1208–1210.
- [3] M. BARUCH AND I. Y. BAR-ITZACK, *Optimal weighted orthogonalization of measured modes*, AIAA J., 16 (1978), pp. 346–351.
- [4] K. E. CHU, *Singular value and generalized singular value decomposition and the solution of linear matrix equations*, Linear Algebra Appl., 88/89 (1987), pp. 83–98.
- [5] B. N. DATTA, *Finite element model updating, eigenstructure assignment, and eigenvalue embedding techniques for vibrating systems*, Mech. Syst. Signal Process., 16 (2002), pp. 83–96.
- [6] M. I. FRISWELL, D. J. INMAN, AND D. F. PILKEY, *Direct updating of damping and stiffness matrices*, AIAA J., 36 (1998), pp. 491–493.
- [7] M. I. FRISWELL AND J. E. MOTTERSHEAD, *Finite Element Model Updating in Structural Dynamics*, Kluwer Academic Publishers, Dordrecht, Boston, London, 1995.
- [8] G. H. GOLUB AND C. F. VAN LOAN, *Matrix Computations*, 3rd ed., The Johns Hopkins University Press, Baltimore, London, 1996.
- [9] G. H. GOLUB AND H. ZHA, *Perturbation analysis of the canonical correlations of matrix pairs*, Linear Algebra Appl., 210 (1994), pp. 3–28.

- [10] N. J. HIGHAM, *Computing a nearest symmetric positive semidefinite matrix*, Linear Algebra Appl., 103 (1988), pp. 103–118.
- [11] R. J. HORN AND C. R. JOHNSON, *Topics in Matrix Analysis*, Cambridge University Press, Cambridge, UK, 1991.
- [12] L.-P. HUANG AND Q.-G. ZENG, *The matrix equation $AXB + CYD = E$ over a simple Artinian ring*, Linear and Multilinear Algebra, 38 (1995), pp. 225–232.
- [13] D. J. INMAN AND J. DANIEL, *Vibration with Control, Measurement, and Stability*, Prentice-Hall, Englewood Cliffs, NJ, 1989.
- [14] Z.-X. JIANG AND Q.-S. LU, *On optimal approximation of a matrix under spectral restriction*, Math. Numer. Sinica, 8 (1986), pp. 47–52 (in Chinese).
- [15] A.-P. LIAO AND Z.-Z. BAI, *The constrained solutions of two matrix equations*, Acta Math. Sinica (Engl. Ser.), 18 (2002), pp. 671–678.
- [16] A.-P. LIAO AND Z.-Z. BAI, *Least-squares solutions of the matrix equation $A^*XA = D$ in bisymmetric matrix set*, Math. Numer. Sinica, 24 (2002), pp. 9–20 (in Chinese).
- [17] A.-P. LIAO AND Z.-Z. BAI, *Least-squares solution of $AXB = D$ over symmetric positive semidefinite matrices X* , J. Comput. Math., 21 (2003), pp. 175–182.
- [18] A.-P. LIAO AND Z.-Z. BAI, *Least squares symmetric and skew-symmetric solutions of the matrix equation $AXA^T + BYB^T = C$ with the least norm*, Math. Numer. Sinica, 27 (2005), pp. 81–95 (in Chinese).
- [19] C. C. PAIGE AND M. A. SAUNDERS, *Towards a generalized singular value decomposition*, SIAM J. Numer. Anal., 18 (1981), pp. 398–405.
- [20] R. PENROSE, *On best approximate solutions of linear matrix equations*, Proc. Cambridge Philos. Soc., 52 (1956), pp. 17–19.
- [21] W. E. ROTH, *The equations $AX - YB = C$ and $AX - XB = C$ in matrices*, Proc. Amer. Math. Soc., 3 (1952), pp. 392–396.
- [22] G. W. STEWART, *Computing the CS-decomposition of a partitioned orthogonal matrix*, Numer. Math., 40 (1982), pp. 297–306.
- [23] G. W. STEWART AND J.-G. SUN, *Matrix Perturbation Theory*, Academic Press, New York, 1990.
- [24] S.-Y. SHIM AND Y. CHEN, *Least squares solution of matrix equation $AXB^* + CYD^* = E$* , SIAM J. Matrix Anal. Appl., 24 (2003), pp. 802–808.
- [25] A. B. ÖZGÜLER, *The equation $AXB + CYD = E$ over a principal ideal domain*, SIAM J. Matrix Anal. Appl., 12 (1991), pp. 581–591.
- [26] R.-S. WANG, *Functional Analysis and Optimization Theory*, Beijing University of Aeronautics and Astronautics Press, Beijing, 2003 (in Chinese).
- [27] G.-P. XU, M.-S. WEI, AND D.-S. ZHENG, *On solutions of matrix equation $AXB + CYD = F$* , Linear Algebra Appl., 279 (1998), pp. 93–109.
- [28] Y.-X. YUAN, *The minimum norm solutions of two classes of matrix equations*, Numer. Math. J. Chinese Univ., 24 (2002), pp. 127–134 (in Chinese).
- [29] Y.-X. YUAN, *The optimal solution of linear matrix equation by using matrix decompositions*, Math. Numer. Sinica, 24 (2002), pp. 165–176 (in Chinese).
- [30] L. ZHANG, *Approximation on a closed convex cone and its numerical application*, Hunan Ann. Math., 6 (1986), pp. 16–22 (in Chinese).

NEW CRITERIA FOR THE SEMICONVERGENCE OF INTERVAL MATRICES*

HANS-ROBERT ARNDT[†] AND GÜNTHER MAYER[†]

Dedicated to Professor Dr. G. Alefeld, Karlsruhe, on the occasion of his 65th birthday

Abstract. We derive new criteria for the convergence of powers of interval matrices $[A]$ to a limit which may differ from O . The absolute value $||[A]||$ of $[A]$ is allowed to be reducible with slight additional restrictions.

Key words. powers of interval matrices, convergence of matrices, semiconvergence of matrices, interval analysis

AMS subject classification. 65G10

DOI. 10.1137/05062264X

1. Introduction. Consider the following initial value problem:

$$(1.1) \quad y'(t) = \{A + B(t)\}y(t) + \{a + b(t)\}, \quad t > t_0,$$

$$(1.2) \quad y(t_0) = y^0,$$

where $t, t_0 \in \mathbb{R}$; $a, b(t), y^0, y(t) \in \mathbb{R}^n$; $A, B(t) \in \mathbb{R}^{n \times n}$; and B and b are sufficiently smooth periodic functions with the same period $T > 0$, i.e., $B(t+T) = B(t)$, $b(t+T) = b(t)$. (Mathieu's differential equation

$$y''(t) + cy'(t) + \{a + b \cos(2t)\}y(t) = 0, \quad c > 0, \quad a, b \in \mathbb{R},$$

illustrates (1.1) when transformed to a linear system of the first order.) Let the matrix valued function $\Phi(t)$ denote the fundamental matrix associated with the homogeneous system

$$(1.3) \quad y'(t) = \{A + B(t)\}y(t), \quad t \in \mathbb{R},$$

with the initial values

$$(1.4) \quad \Phi(t_0) = I \quad (I = \text{identity matrix}).$$

Since the functions $\Phi(t+T)$ and $\Phi(t) \cdot \Phi(t_0+T)$ both satisfy (1.3) with the same initial value $\Phi(t_0+T)$, they are identical, and an inductive argument shows

$$(1.5) \quad \Phi(t+kT) = \Phi(t) \cdot \{\Phi(t_0+T)\}^k, \quad t_0 \leq t \leq t_0+T, \quad k \in \mathbb{N}_0.$$

Since (1.1), (1.2) is an inhomogeneous linear initial value problem, stability can be studied by means of (1.3) (cf., for instance, [7, pp. 226, 227]). By virtue of (1.4) every solution y_h of (1.3) with initial value $y_h(t_0) = y^0$ can be written as $y_h(t) = \Phi(t)y^0$.

*Received by the editors January 13, 2005; accepted for publication (in revised form) by A. Frommer May 31, 2005; published electronically December 30, 2005.

<http://www.siam.org/journals/simax/27-3/62264.html>

[†]Institut für Mathematik, Universität Rostock, D-18051 Rostock, Germany (hans-robert.arndt@uni-rostock.de, guenter.mayer@uni-rostock.de). The first author was supported by a grant (Landesgraduiertenförderung) of the German state Mecklenburg-Vorpommern.

Hence by (1.5) the solution of (1.1), (1.2) is stable if and only if the powers of the matrix $\Phi(t_0 + T)$ are bounded, and asymptotically stable if and only if these powers tend to zero. Enclosing $\Phi(t_0 + T)$ by using a software package such as AWA (see [8], [14]) yields interval matrices whose powers can be used to derive sufficient criteria for stability and asymptotic stability. To this end, criteria have to be provided to decide whether $[A]^\infty = \lim_{k \rightarrow \infty} [A]^k = O$ holds for a given interval matrix $[A]$ or whether the sequence of powers $[A]^k$ is at least bounded. Such necessary and sufficient criteria were first given in [9] for the case $[A]^\infty = O$ and in [10] for the case $[A]^\infty \neq O$, provided that the absolute value $||[A]||$ is irreducible in the latter case. In [3] the assumption of irreducibility was dropped, but additional conditions for $[A]$ had to be prescribed. A rigorous proof of stability *by means of a computer* was first given in [5] by using AWA combined with powers of interval matrices. Of course, stability of a solution of (1.1), (1.2) can be completely handled by inspecting the eigenvalues of $\Phi(t_0 + T)$, but normally neither the matrix $\Phi(t_0 + T)$ nor its eigenvalues are known exactly.

In the present paper we generalize the results in [3]. We also study the more general case, where the matrices $[A]^k$, $k = 0, 1, \dots$, are merely bounded, although our main goal is to obtain their convergence. Note that we do not take into account conversion or rounding errors or overestimations arising from practical problems such as the inclusion of $\Phi(t_0 + T)$ by interval methods for initial value problems. In particular, we do not intend to illustrate our theory by enclosing $\Phi(t_0 + T)$ —a task which was part of an entire Ph.D. thesis [5].

We have organized our paper as follows. In section 2 we list some notation and basic facts which we will use later on. Section 3 contains some auxiliary results for real or interval matrices. In section 4 we present our new results, among them a far-reaching necessary and sufficient criterion (Theorem 4.9) for the semiconvergence (defined in section 2) of $[A]$. This criterion uses a companion matrix $B \in \mathbb{R}^{n \times n}$ which has already been introduced in [9]. In contrast to [9], where $\lim_{k \rightarrow \infty} B^k = O$ was equivalent to $[A]^\infty = O$, this equivalence holds in the case $[A]^\infty \neq O$ only under some additional conditions which are weaker than those in [3]. Theorem 4.9 is specialized such that $||[A]||$ can replace or equal B . Moreover, two open cases are listed which are not covered by Theorem 4.9. The difficulties caused by them are illustrated by examples. Criteria for the semiconvergence in the two open cases are given for restricted subclasses of matrices.

2. Notation. By $I(\mathbb{R}), I(\mathbb{R}^n), I(\mathbb{R}^{n \times n})$ we denote the set of intervals, the set of interval vectors with n components, and the set of $n \times n$ interval matrices, respectively. By “interval” we always mean a real compact interval. We write interval quantities in brackets with the exception of point quantities (i.e., degenerate interval quantities) which we identify with the element which they contain. Examples are the null matrix O , the identity matrix I , and the vector $e = (1, 1, \dots, 1)^T$. We use the notation $[A] = [\underline{A}, \bar{A}] = ([a]_{ij}) = ([\underline{a}_{ij}, \bar{a}_{ij}]) \in I(\mathbb{R}^{n \times n})$ simultaneously without further reference, and we proceed similarly for the elements of $\mathbb{R}^n, \mathbb{R}^{n \times n}, I(\mathbb{R})$, and $I(\mathbb{R}^n)$. If $[A]$ is partitioned in blocks $[A]_{ij}$, $i, j = 1, \dots, s$, we also write $[A] = ([A]_{ij}) = ([A]_{ij})_{i,j=1,\dots,s}$, and we use an analogous notation for real matrices if a block partition is emphasized. We call $[a] \in I(\mathbb{R})$ *symmetric* if $[a] = -[a]$, i.e., if $[a] = [-r, r]$ with some real number $r \geq 0$. For intervals $[a], [b]$ we introduce the *midpoint* $\hat{a} := (\underline{a} + \bar{a})/2$, the *absolute value* $||[a]|| := \max\{|\underline{a}|, |\bar{a}|\}$, the *radius* $\text{rad}([a]) := (\bar{a} - \underline{a})/2$, and the (*Hausdorff*) *distance* $q([a], [b]) := \max\{|\underline{a} - \underline{b}|, |\bar{a} - \bar{b}|\}$. For interval vectors and interval matrices these quantities are defined entrywise, for instance, $||[A]|| := (||[a]_{ij}||) \in \mathbb{R}^{n \times n}$. We assume some familiarity when working with these definitions. For details see, e.g., the

introductory chapters of [1] and [13].

Following [9] and [10] we define

$$[A]^0 := I, \quad [A]^{k+1} = ([a]_{ij}^{(k+1)}) := [A]^k \cdot [A], \quad k = 0, 1, \dots,$$

and

$${}^0[A] := I, \quad {}^{k+1}[A] = ({}^{(k+1)}[a]_{ij}) := [A] \cdot {}^k[A], \quad k = 0, 1, \dots$$

The example $[A] = \begin{pmatrix} 1 & [0, 1] \\ 1 & -1 \end{pmatrix}$ shows that $[A]^3$ can differ from ${}^3[A]$. If $\lim_{k \rightarrow \infty} [A]^k$ exists (with respect to the Hausdorff distance q), then we write $[A]^\infty$ for this limit, and A^∞ if $[A] \equiv A \in \mathbb{R}^{n \times n}$. Similarly we write ${}^\infty[A] := \lim_{k \rightarrow \infty} {}^k[A]$. As in the degenerate case, we call $[A] \in I(\mathbb{R}^{n \times n})$ *semiconvergent* if $[A]^\infty$ exists and *convergent* if it is semiconvergent with $[A]^\infty = O$. If $[A]$ is represented by blocks $[A]_{ij}$, i.e., $[A] = ([A]_{ij})$, we write $[A]_{ij}^{(k)}$ for the blocks of $[A]^k$ corresponding to $[A]_{ij}$.

The (*directed*) graph $G(A) = (X, E)$ of a matrix $A \in \mathbb{R}^{n \times n}$ consists of the sets $X = \{1, 2, \dots, n\}$ and $E = \{(i, j) \mid a_{ij} \neq 0\} \subseteq X \times X$. The elements of X are called *nodes*; those of E are called (*directed*) *edges*. We write $i \rightarrow j$ if $(i, j) \in E$. A sequence

$$(2.1) \quad (i, i_1), (i_1, i_2), \dots, (i_{l-2}, i_{l-1}), (i_{l-1}, j)$$

of edges is called a *path of length l* , which connects the node i with the node j . We abbreviate (2.1) by

$$(2.2) \quad i \rightarrow i_1 \rightarrow i_2 \rightarrow \dots \rightarrow i_{l-2} \rightarrow i_{l-1} \rightarrow j.$$

A matrix $A \in \mathbb{R}^{n \times n}$ is called *irreducible* if and only if for every two nodes i, j there is a path which connects i with j . Otherwise A is called *reducible*. It is well known (see, e.g., [15]) that a reducible matrix A can be transformed into the block form

$$(2.3) \quad R(A) = PAP^T = \begin{pmatrix} A_{11} & A_{12} & \dots & A_{1s} \\ O & A_{22} & \dots & A_{2s} \\ \vdots & \ddots & \ddots & \vdots \\ O & \dots & O & A_{ss} \end{pmatrix}$$

by means of an appropriate permutation matrix P . Here, A_{ii} are square block matrices which are either irreducible or 1×1 zero matrices. The matrix $R(A) = (A_{ij})$ is called the *reducible normal form* of A . It is unique up to some additional permutations of rows and corresponding columns. We say that $[A] = ([A]_{ij}) \in I(\mathbb{R}^{n \times n})$ is in reducible normal form if $(|[A]_{ij}|)$ is the reducible normal form of $|[A]|$. For such matrices we define the index set

$$\text{ind}([A]_{ii}) := \{l \mid [a]_{il} \text{ is an entry of } [A]_{ii}\},$$

where l is the index with respect to the matrix $[A]$.

A graph $G_B(A) = (X_B, E_B)$ is called a *block graph* of a block matrix $A = (A_{ij})_{i,j=1,\dots,s} \in \mathbb{R}^{n \times n}$ with blocks A_{ij} if $X_B = \{1, 2, \dots, s\}$ and $E_B = \{(i, j) \mid A_{ij} \neq O\} \subseteq X_B \times X_B$. We use the notation $i \xrightarrow{B} j$ and $i \xrightarrow{B} i_1 \xrightarrow{B} i_2 \xrightarrow{B} \dots \xrightarrow{B} i_{m-1} \xrightarrow{B} j$ analogously to (2.2). In the latter case we say that i is connected with j in $G_B(A)$.

As usual we call the matrix $A \in \mathbb{R}^{m \times n}$ *nonnegative* if $a_{ij} \geq 0$ for $i = 1, \dots, m$, $j = 1, \dots, n$, writing $A \geq O$ in this case. By $A > O$ we denote nonnegative matrices

whose entries all are positive. We call them *positive*. For $A, B \in \mathbb{R}^{m \times n}$ the inequality $A \leq B$ means $B - A \geq O$, and $A \geq B$ is equivalent to $B \leq A$. For vectors we apply these definitions analogously.

According to the theorem of Perron and Frobenius for irreducible nonnegative matrices A the spectral radius $\rho(A)$ is a positive simple eigenvalue of A , and there are two positive eigenvectors x, y such that

$$(2.4) \quad Ax = \rho(A)x, \quad y^T A = \rho(A)y^T, \quad y^T x = 1$$

hold (see, e.g., [15, p. 30] and [6, p. 500]). We call such vectors (right and left, respectively) *Perron vectors* of A . In our paper we will use x, y exclusively for such vectors. Note that we do not require the normalization $\sum_{i=1}^n x_i = 1$ or $\sum_{i=1}^n y_i = 1$, as was done in [6, p. 497], in order to make Perron vectors unique.

In matrix theory one often divides nonnegative irreducible matrices into two classes: the *primitive* matrices, which have, by definition, only $\rho(A)$ as a (simple) eigenvalue λ with $|\lambda| = \rho(A)$, and the *cyclic* matrices of index $h > 1$ with the (simple) eigenvalues $\lambda_j = \rho(A) e^{\frac{2\pi i}{h} j}$, $j = 0, 1, \dots, h-1$. The theory guarantees that other cases cannot occur for such matrices. By means of some appropriate permutation matrix P , cyclic matrices A of index h can be brought into the so-called *cyclic normal form*

$$(2.5) \quad PAP^T = \begin{pmatrix} O & A_{12} & O & O & \dots & O \\ O & O & A_{23} & O & \dots & O \\ \vdots & \vdots & \ddots & \ddots & \ddots & \vdots \\ O & O & \dots & O & A_{h-2,h-1} & O \\ O & O & \dots & O & O & A_{h-1,h} \\ A_{h,1} & O & \dots & O & O & O \end{pmatrix}$$

with square diagonal blocks $A_{ii} = O$. This shows that $A^k > O$ can never occur for cyclic matrices, in contrast to primitive ones for which there is a smallest integer k_0 (the so-called *index of primitivity*) such that $A^k > O$ holds for all $k \geq k_0 = k_0(A)$. (See, e.g., [4, p. 46] and [15, p. 42].)

By $D = \text{diag}(\sigma_1, \dots, \sigma_n) \in \mathbb{R}^{n \times n}$ we denote the diagonal matrix whose diagonal entries are $\sigma_1, \dots, \sigma_n$. If $|D| = I$ holds, then we call $D \in \mathbb{R}^{n \times n}$ a *signature matrix*. Note that $D^{-1} = D$ for such matrices.

A vector $x \in \mathbb{C}^n \setminus \{0\}$ is called a (*right*) *principal vector of degree* $m \geq 2$ associated with an eigenvalue λ of $A \in \mathbb{R}^{n \times n}$ if $(A - \lambda I)^m x = 0$ while $(A - \lambda I)^{m-1} x \neq 0$. Generalizing this definition, we consider eigenvectors as principal vectors of degree 1.

3. Auxiliary results. In this section we present some results which are used later on and which are helpful to understanding the subsequent section.

LEMMA 3.1. *Let v be a right principal vector of $A \in \mathbb{R}^{n \times n}$ of degree $l \in \mathbb{N}$ associated with an eigenvalue λ , and let w be a left eigenvector of A with respect to an eigenvalue $\mu \neq \lambda$. Then $w^T v = 0$.*

Proof. The assertion follows immediately from $\lambda \neq \mu$ and $0 = w^T((A - \lambda I)^l v) = (w^T(A - \lambda I)^l)v = (\mu - \lambda)^l w^T v$. $\square$

We next recall properties of real matrices which are equivalent to semiconvergence.

THEOREM 3.2 (cf. [4, p. 152]). *The matrix $A \in \mathbb{R}^{n \times n}$ is semiconvergent if and only if the following conditions hold:*

- (i) $\rho(A) \leq 1$.
- (ii) *If $\rho(A) = 1$ and if λ is an eigenvalue of A with $|\lambda| = 1$, then $\lambda = 1$ and every Jordan block associated with $\lambda = 1$ is of dimension 1×1 .*

THEOREM 3.3. *Let $A \in \mathbb{R}^{n \times n}$ be in reducible normal form (2.3) with blocks A_{ij} , $i, j = 1, \dots, s$. Then A is semiconvergent if and only if the following two properties hold:*

- (i) A_{ii} is semiconvergent for all $i = 1, \dots, s$.
- (ii) *The sequence (A^k) is bounded.*

Proof. The implication “ $\Rightarrow$ ” is trivial. In order to prove the converse we first note $\rho(A) = \max_{1 \leq i \leq s} \rho(A_{ii}) \leq 1$ by (i). If A has an eigenvalue λ such that $|\lambda| = 1$, then it must be 1 again by (i). Using (ii) and the Jordan normal form $J = S^{-1}AS$ of A shows that the powers of the Jordan blocks are bounded; hence they must be 1×1 for the eigenvalues $\lambda = 1$. Now Theorem 3.2 finishes the proof. $\square$

THEOREM 3.4 (cf. [6, Thm. 8.2.11, p. 500] or [10, Lem. 3]). *Let $A \in \mathbb{R}^{n \times n}$ be irreducible and nonnegative with $\rho(A) = 1$. Then A is semiconvergent if and only if A is primitive. In this case we have*

$$A^\infty = xy^T, \quad x, y \text{ as in (2.4).}$$

Our next result shows a close relation between connectivity between two nodes in the block graph $G_B(A)$ of a matrix A which is in reducible normal form and connectivity between two nodes in the ordinary graph $G(A)$ of A .

LEMMA 3.5 (cf. [3, Thm. 4]). *Let $A \in \mathbb{R}^{n \times n}$ be in reducible normal form (2.3) with blocks A_{ij} , $i, j = 1, \dots, s$. Then i is connected to i' in the block graph $G_B(A)$ of $A = (A_{ij})$ if and only if each $k \in \text{ind}(A_{ii})$ is connected in $G(A)$ with each $k' \in \text{ind}(A_{i'i'})$.*

We can use the definition of block connectivity in a criterion for semiconvergence for nonnegative reducible matrices.

THEOREM 3.6 (cf. [3, Thm. 6]). *Let $A \in \mathbb{R}^{n \times n}$ be nonnegative and in reducible normal form (2.3) with blocks A_{ij} , $i, j = 1, \dots, s$. Then A is semiconvergent if and only if the following two properties hold:*

- (i) A_{ii} is semiconvergent for all $i = 1, \dots, s$.
- (ii) *If $\rho(A_{ii}) = \rho(A_{i'i'}) = 1$ holds for some i, i' with $i < i'$, then i is not connected to i' in the block graph $G_B(A)$.*

Now we address cyclic matrices. To this end we introduce some notation which we will use throughout the paper.

DEFINITION 3.7. *Let $0 \leq A \in \mathbb{R}^{n \times n}$ be irreducible and cyclic of index $h \in \mathbb{N} \setminus \{1\}$. Assume that A is already in cyclic normal form (2.5) with blocks A_{ij} , spectral radius $\rho := \rho(A)$, and the eigenvalues λ_j , $j = 1, \dots, n$, (counted according to their multiplicity), and the corresponding right and left eigen- and principal vectors $v^{(j)}$ and $w^{(j)}$, respectively, $j = 1, \dots, n$, which form a basis of $\mathbb{R}^n$. W.l.o.g. let $\lambda_j = \rho e^{i\theta(j-1)}$, $\theta = 2\pi/h$, for $j = 1, \dots, h$, and $|\lambda_j| < \rho$ for $j = h+1, \dots, n$, as is guaranteed by the theory. Let $v^{(1)}$ and $w^{(1)}$ be the Perron vectors x, y from (2.4) for which $y^T x = 1$ holds and which are partitioned into blocks according to (2.5).*

Denote by $x^{(j)}, y^{(j)} \in \mathbb{R}^n$, $j = 1, \dots, h$, those nonnegative vectors which have the positive j th block in common with x and y , respectively, and which are zero otherwise. (Note that the position of the zero entries coincides in $x^{(j)}$ and $y^{(j)}$ for the same index j .)

Define the matrices D_j , $j = 1, \dots, h$, by

$$D_j = A_{j+1}A_{j+2} \times \dots \times A_h A_1 \times \dots \times A_j \quad (h \text{ factors}).$$

LEMMA 3.8. *With the assumptions and notation of Definition 3.7 the following assertions hold:*

- (a) $\sum_{j=1}^h x^{(j)} = x$, $\sum_{j=1}^h y^{(j)} = y$.
- (b) $Ax^{(1)} = \rho x^{(h)}$, $Ax^{(j)} = \rho x^{(j-1)}$, $j = 2, \dots, h$,
 $(y^{(h)})^T A = \rho (y^{(1)})^T$, $(y^{(j)})^T A = \rho (y^{(j+1)})^T$, $j = 1, \dots, h-1$.
 $(y^{(j)})^T x^{(l)} = \delta_{jl}/h$, $j, l = 1, \dots, h$ (δ_{jl} Kronecker symbol).
- (c) The matrices D_j , $j = 1, \dots, h$, are irreducible and primitive with spectral radius $\rho(D_j) = \rho^h$. Moreover, $A^h = \text{diag}(D_1, \dots, D_h)$, $A^h x^{(j)} = \rho^h x^{(j)}$, $(y^{(j)})^T A = \rho^h (y^{(j)})^T$, $j = 1, \dots, h$.
- (d) If $\rho = 1$, then the limit $A^{h \cdot \infty} := \lim_{k \rightarrow \infty} A^{h \cdot k}$ exists and satisfies

$$A^{h \cdot \infty} = h \cdot \sum_{j=1}^h x^{(j)} (y^{(j)})^T.$$

- (e) If v , respectively, w , is a right, respectively, left, principal vector of A of degree l associated with an eigenvalue λ such that $|\lambda| < \rho$, then $(y^{(j)})^T v = 0$, $w^T x^{(j)} = 0$, $j = 1, \dots, h$.
- (f) The vectors $x^{(j)}$, $j = 1, \dots, h$, together with $v^{(j)}$, $j = h+1, \dots, n$, form a basis of $\mathbb{R}^n$. Another basis of $\mathbb{R}^n$ is formed by the vectors $y^{(j)}$, $j = 1, \dots, h$, together with $w^{(j)}$, $j = h+1, \dots, n$.
- (g) Let $R \in \mathbb{R}^{m \times n}$. Then

$$(3.1) \quad R = zy^T + \sum_{l=h+1}^n z^{(l)} (w^{(l)})^T$$

holds if and only if

$$(3.2) \quad Rx^{(j)} = c, \quad j = 1, \dots, h,$$

where $c, z, z^{(l)} \in \mathbb{R}^m$ are appropriate vectors. (Note that (3.1) means that no left eigenvalue $w^{(j)}$, $j = 2, \dots, h$, is needed to represent a row of R , and (3.2) means that the product $Rx^{(j)}$ is independent of j .)

Proof.

- (a) This proof follows directly from the definition of $x^{(j)}$ and $y^{(j)}$.
- (b) The first four formulae follow immediately from $Ax = \rho x$, $y^T A = \rho y^T$, and (2.5). The orthogonality of $y^{(j)}$ and $x^{(l)}$ for $j \neq l$ results from the position of the zeros in both vectors. (See also the note in Definition 3.7.) In order to prove the result in the case $j = l$, proceed inductively as follows:

$$\begin{aligned} (y^{(h)})^T x^{(h)} &= ((y^{(h-1)})^T A) x^{(h)} / \rho = (y^{(h-1)})^T (Ax^{(h)}) / \rho \\ &= (y^{(h-1)})^T x^{(h-1)} = \dots = (y^{(1)})^T x^{(1)}. \end{aligned}$$

Using (a) and the orthogonality, we get

$$1 = y^T x = \sum_{j=1}^h (y^{(j)})^T x^{(j)} = h(y^{(1)})^T x^{(1)},$$

whence $(y^{(j)})^T x^{(j)} = 1/h$, $j = 1, \dots, h$.

(c) The assertions follow essentially from Theorem 2.6 in [15]. The eigenvalue/eigenvector relation follows by applying (b) iteratively h times.

(d) This proof follows directly from (b), (c), and Theorem 3.4.

(e) We have $(y^{(j)})^T (A^h - \lambda I)^l v = (y^{(j)})^T (A^{h-1} + \lambda A^{h-2} + \lambda^2 A^{h-3} + \dots + \lambda^{h-2} A + \lambda^{h-1} I)^l ((A - \lambda I)^l v) = 0$, and with (c) we get $((y^{(j)})^T (A^h - \lambda I)^l) v = (\rho^h - \lambda)^l (y^{(j)})^T v$. Putting both results together yields $(y^{(j)})^T v = 0$. The remaining part can be shown analogously.

(f) Multiply the equation

$$\sum_{j=1}^h \alpha_j x^{(j)} + \sum_{j=h+1}^n \alpha_j v^{(j)} = 0$$

by $(y^{(l)})^T$ from the left and take into account (b) and (e) to end up with $\alpha_l/h = 0$. Hence $\alpha_l = 0$, $l = 1, \dots, h$, and the linear independence of $v^{(j)}$ yields $\alpha_l = 0$, $l = h+1, \dots, n$. This proves the basis in the first case. The second one is treated analogously.

(g) Let (3.1) hold. Multiplying it by $x^{(j)}$ from the right and applying (e) and (b) yields $Rx^{(j)} = zy^T x^{(j)} = z(y^{(j)})^T x^{(j)} = z/h = c$, $j = 1, \dots, h$.

Conversely, let (3.2) hold. By virtue of (f) we can represent R as

$$(3.3) \quad R = \sum_{l=1}^h z^{(l)} (y^{(l)})^T + \sum_{l=h+1}^n z^{(l)} (w^{(l)})^T.$$

Multiply this equation from the right by $x^{(j)}$ and apply (e) and (b) in order to get $Rx^{(j)} = z^{(j)}/h$. Together with (3.2) one obtains $z^{(j)} = h \cdot c$. Plugging this result into (3.3) and applying (a) results in

$$\begin{aligned} R &= \sum_{l=1}^h h \cdot c (y^{(l)})^T + \sum_{l=h+1}^n z^{(l)} (w^{(l)})^T = hc \sum_{l=1}^h (y^{(l)})^T + \sum_{l=h+1}^n z^{(l)} (w^{(l)})^T \\ &= hcy^T + \sum_{l=h+1}^n z^{(l)} (w^{(l)})^T \end{aligned}$$

as in (3.1) with $z = h \cdot c$. $\square$

Now we present some results from interval analysis. The first was proved in [11], and the second one is a generalization of a theorem due to O. Mayer [12] on the global convergence of the total step method supplemented by some result on degenerate input data and degenerate starting vectors. The third, to our knowledge, is new.

We start with a result on the series $\sum_{k=0}^{\infty} |[A]^k|$.

LEMMA 3.9 (cf. [11, Thm. 2 and Lem. 3]). *Let $[A] \in \mathbf{I}(\mathbb{R}^{n \times n})$ be convergent. Then $\sum_{k=0}^{\infty} |[A]^k|$ is convergent.*

In our next lemma we study some nonstationary iteration.

LEMMA 3.10 (cf. [3, Lem. 5]). *Let $[A] \in \mathbf{I}(\mathbb{R}^{n \times n})$, $[C]^{(k)}$, $[C]^* \in \mathbf{I}(\mathbb{R}^{m \times n})$, $\rho([A]) < 1$, $\lim_{k \rightarrow \infty} [C]^{(k)} = [C]^*$.*

Then for any matrix $[X]^{(0)} \in \mathbf{I}(\mathbb{R}^{m \times n})$ the iterates

$$(3.4) \quad [X]^{(k+1)} = [C]^{(k)} + [X]^{(k)}[A], \quad k = 0, 1, \dots,$$

converge to some limit $[X]^*$ which satisfies

$$(3.5) \quad [X]^* = [C]^* + [X]^*[A].$$

This limit $[X]^*$ is independent of the starting matrix $[X]^{(0)}$. It is the unique matrix which satisfies (3.5).

The assertions hold analogously for $[C]^{(k)}$, $[C]^*$, $[X]^{(0)} \in \mathbf{I}(\mathbb{R}^{n \times m})$, and

$$(3.6) \quad [X]^{(k+1)} = [C]^{(k)} + [A][X]^{(k)}.$$

If all matrices are degenerate (in particular, $X^{(0)} \in \mathbb{R}^{m \times n}$; respectively, $X^{(0)} \in \mathbb{R}^{n \times m}$), then all assertions remain true if one replaces the assumption $\rho(|A|) < 1$ by the weaker one, $\rho(A) < 1$.

Proof. We prove only the last statement. The proof of the other ones can be found in [3]. From the analogue of (3.5) we get

$$X^* = C^*(I - A)^{-1} = \sum_{j=0}^{\infty} C^* A^j.$$

The analogue of (3.4) yields

$$X^{(k+1)} = \sum_{j=0}^k C^{(k-j)} A^j + X^{(0)} A^{k+1}.$$

Combining both equations results in

$$(3.7) \quad X^{(k+1)} - X^* = \sum_{j=0}^{k_0} (C^{(k-j)} - C^*) A^j + \sum_{j=k_0+1}^k (C^{(k-j)} - C^*) A^j + X^{(0)} A^{k+1} - \sum_{j=k+1}^{\infty} C^* A^j$$

for any $k_0 < k$. Since $\lim_{j \rightarrow \infty} C^{(j)} = C^*$, the sequence $(|C^{(j)} - C^*|)$ is bounded by some nonnegative matrix K . Analogously, $|A^j| \leq L$ for some nonnegative matrix L and all $j \in \mathbb{N}_0$. Given $\varepsilon > 0$, choose k_0, k_1 such that $\sum_{j=k_0+1}^{\infty} |A^j| \leq \varepsilon e e^T$ (note Lemma 3.9!) and $|C^{(j)} - C^*| \leq \varepsilon e e^T$ for $j \geq k_1$. For any k with $k - k_0 > k_1$ we get from (3.7)

$$\begin{aligned} |X^{(k+1)} - X^*| &\leq \varepsilon e e^T \sum_{j=0}^{k_0} |A^j| + K \sum_{j=k_0+1}^k |A^j| + |X^{(0)}| \varepsilon e e^T + |C^*| \sum_{j=k+1}^{\infty} |A^j| \\ &\leq \varepsilon \left(e e^T (k_0 + 1) L + K e e^T + |X^{(0)}| e e^T + |C^*| e e^T \right). \end{aligned}$$

By choosing $\varepsilon > 0$ sufficiently small, the right-hand side can be made arbitrarily small, which finishes the proof. $\square$

The next lemma concerns a situation similar to Lemma 3.10. Now we allow $\rho(|A|) = 1$ with stronger restrictions on $[C]^{(k)}$.

LEMMA 3.11. *Consider the iteration*

$$(3.8) \quad [X]^{(k+1)} = [C]^{(k)} + [X]^{(k)}[A], \quad k = 0, 1, \dots,$$

with $[A] \in \mathbf{I}(\mathbb{R}^{n \times n})$, $[A]$ irreducible, and $[C]^{(k)} \in \mathbf{I}(\mathbb{R}^{m \times n})$. Let the following conditions hold:

(i) $[C]^* = \lim_{k \rightarrow \infty} [C]^{(k)}$ exists with convergent matrix series $\sum_{j=0}^{\infty} q([C]^{(j)}, [C]^*) =: T \in \mathbb{R}^{m \times n}$.

(ii) There exists a solution $[X]^* \in \mathbf{I}(\mathbb{R}^{m \times n})$ of

$$[X] = [C]^* + [X][A].$$

(iii) $|[A]|$ and $[A]$ are semiconvergent.

Then for any matrix $[X]^{(0)} \in \mathbf{I}(\mathbb{R}^{m \times n})$ the iterates of (3.8) are convergent.

The assertion holds analogously for $[C]^{(k)}, [X]^{(0)} \in \mathbf{I}(\mathbb{R}^{n \times m})$, and

$$[X]^{(k+1)} = [A][X]^{(k)} + [C]^{(k)}.$$

Proof. Since $|[A]|$ is semiconvergent, the sequence $(|[A]|^k)$ is bounded by some positive real matrix K . The sequence $([X]^{(k)})$ is bounded because of

$$\begin{aligned} q([X]^{(k)}, [X]^*) &= q([C]^{(k-1)} + [X]^{(k-1)}[A], [C]^* + [X]^*[A]) \\ &\leq q([C]^{(k-1)}, [C]^*) + q([X]^{(k-1)}, [X]^*) \cdot |[A]| \leq \dots \\ &\leq \sum_{j=0}^{k-1} q([C]^{(j)}, [C]^*) |[A]|^{k-j-1} + q([X]^{(0)}, [X]^*) \cdot |[A]|^k \\ &\leq TK + q([X]^{(0)}, [X]^*) \cdot K. \end{aligned}$$

So there exists a subsequence $([X]^{(k_l)})$ which is convergent to some $[X]^*$. Now we take such an $[X]^*$ as a starting matrix $[X]^{(0)}$ in the iteration

$$[X]^{(m+1)} = [C]^* + [X]^{(m)}[A]$$

and denote the resulting iterates by $[X]^{*m}$. Condition (iii) ensures that these iterates converge to some matrix $[X]^{**}$ (cf. [2, Thms. 6 and 7]); hence there is some integer $M(\varepsilon)$ such that

$$(3.9) \quad q([X]^{*m}, [X]^{**}) \leq \frac{\varepsilon}{2} \quad \text{for all } m > M(\varepsilon).$$

Furthermore we have

$$\begin{aligned} q([X]^{(k_l+m)}, [X]^{*m}) &= q([C]^{(k_l+m-1)} + [X]^{(k_l+m-1)}[A], [C]^* + [X]^{*m-1}[A]) \\ &\leq q([C]^{(k_l+m-1)}, [C]^*) + q([X]^{(k_l+m-1)}, [X]^{*m-1}) \cdot |[A]| \leq \dots \\ &\leq \sum_{j=k_l}^{k_l+m-1} q([C]^{(j)}, [C]^*) |[A]|^{k_l+m-1-j} + q([X]^{(k_l)}, [X]^*) \cdot |[A]|^m \\ &\leq \sum_{j=k_l}^{\infty} q([C]^{(j)}, [C]^*) \cdot K + q([X]^{(k_l)}, [X]^*) \cdot K. \end{aligned}$$

The right-hand side becomes infinitely small if l tends to infinity; hence there is some integer $L(\varepsilon)$ such that

$$(3.10) \quad q([X]^{(k_l+m)}, [X]^{*m}) \leq \frac{\varepsilon}{2} \quad \text{for all } l > L(\varepsilon).$$

The inequalities (3.9) and (3.10) lead to

$$q([X]^{(k_l+m)}, [X]^{**}) \leq q([X]^{(k_l+m)}, [X]^{*m}) + q([X]^{*m}, [X]^{**}) \leq \varepsilon$$

for all $l > L(\varepsilon)$, $m > M(\varepsilon)$, and so the sequence $([X]^{(k)})$ is convergent. $\square$

4. New results. In this section we present new results on semiconvergent interval matrices $[A]$. In order to prove them we repeat some results of [3], [9], and [10]. We first characterize those columns of an interval matrix $[A]$ which are nondegenerate in at least one of the powers $[A]^k$, $k = 1, 2, \dots$. (Cf. Definition 3 in [9].)

DEFINITION 4.1. *The j th column of $[A] \in I(\mathbb{R}^{n \times n})$ has the $*$ -property if there exists a power $[A]^k$ containing in the same j th column at least one nondegenerate interval.*

The following results concerning the $*$ -property and the convergence of interval matrices were proven in [9] and can be seen immediately.

LEMMA 4.2. *Let $[A] \in I(\mathbb{R}^{n \times n})$.*

(a) *The j th column of $[A]$ has the $*$ -property if and only if there is a path $i = i_0 \rightarrow i_1 \rightarrow i_2 \rightarrow \dots \rightarrow i_{m-1} \rightarrow i_m = j$ in the graph $G(|[A]|)$ which contains two neighboring nodes i_l, i_{l+1} such that $[a]_{i_l, i_{l+1}}$ is nondegenerate.*

(b) *If the absolute value $|[A]|$ of a nondegenerate interval matrix $[A]$ is irreducible, then all columns of $[A]$ have the $*$ -property.*

(c) *If the matrix $[A] = ([A]_{ij})$ is in reducible normal form and if its l th column has the $*$ -property with $l \in \text{ind}([A]_{ii})$ for some block index i , then all columns in $\text{ind}([A]_{ii})$ have the $*$ -property. In this case we also say that the block column with index i has the $*$ -property.*

Note that in Lemma 4.2 (c) the particular case $[A]_{ii} = O$ implies $\text{ind}([A]_{ii}) = \{l\}$.

DEFINITION 4.3. *Let $[A] \in I(\mathbb{R}^{n \times n})$. Then the companion matrix $B \in \mathbb{R}^{n \times n}$ of $[A]$ is defined by*

$$b_{ij} := \begin{cases} |[a]_{ij}| & \text{if the } j\text{th column of } [A] \text{ has the } *\text{-property,} \\ [a]_{ij} \equiv a_{ij} & \text{otherwise.} \end{cases}$$

Note that B generally differs from the well-known Frobenius companion matrix of a polynomial, which we do not use throughout this paper. Thus we do not expect mix-ups by introducing the terminology in Definition 4.3.

THEOREM 4.4. *The matrix $[A] \in I(\mathbb{R}^{n \times n})$ is convergent if and only if its companion matrix B satisfies $\rho(B) < 1$.*

If the absolute value $|[A]|$ of a nondegenerate interval matrix $[A]$ is irreducible, then all columns of $[A]$ have the $*$ -property by Lemma 4.2(b); hence $[A]$ is convergent in this case if and only if $\rho(|[A]|) < 1$. Moreover, Theorem 4.4 in connection with Corollary 4 in [9] shows that for an irreducible absolute value $|[A]|$ the matrix $[A]$ is convergent if and only if the limit ${}^\infty[A]$ exists and equals O . For general matrices $|[A]|$ this becomes false as is illustrated by

$$[A] = \begin{pmatrix} 1 & -1 & 0 \\ 1 & -1 & 0 \\ 0 & [0, \frac{1}{2}] & 0 \end{pmatrix},$$

where ${}^\infty[A] = O$ exists while the sequence $([A]^k)$ is divergent.

For $|[A]|$ being irreducible, Theorem 4.4 can be generalized in the following way (cf. [10]), where now $[A]^\infty \neq O$ is possible.

THEOREM 4.5. *Let $|[A]| \in \mathbb{R}^{n \times n}$ be irreducible with $\text{rad}([A]) \neq O$. Then $[A]$ is semiconvergent if and only if the following two conditions hold:*

(i) *The matrix $|[A]|$ is semiconvergent.*

(ii) *If $\rho(|[A]|) = 1$ and if $[A]$ contains only one matrix $\dot{A}$ with $|\dot{A}| = |[A]|$, then $\dot{A} \neq -D|[A]|D$ for any signature matrix D .*

Now we study semiconvergence for matrices $[A]$ in reducible normal form. We start with two negative results.

THEOREM 4.6. *Let $[A] = ([A]_{ij})_{i,j=1,\dots,s} \in \mathbf{I}(\mathbb{R}^{n \times n})$ be in reducible normal form with at least one block column i which has the $*$ -property and which fulfills $\rho := \rho(|[A]_{ii}|) > 1$. Then the sequence $([A]^k)$ is unbounded. In particular, the matrix $[A]$ cannot be semiconvergent.*

Proof. By virtue of the $*$ -property there is some integer m and some block row p such that $\text{rad}([A]_{pi}^{(m)}) \neq O$. Let v be a Perron vector of $|[A]_{ii}|$. Then

$$\begin{aligned} \text{rad}([A]_{pi}^{(k+m)})v &= \sum_{j=1}^i \text{rad}([A]_{pj}^{(k+m-1)}[A]_{ji})v \geq \text{rad}([A]_{pi}^{(k+m-1)}[A]_{ii})v \\ &\geq \text{rad}([A]_{pi}^{(k+m-1)})|A_{ii}|v \geq \dots \geq \text{rad}([A]_{pi}^{(m)})|A_{ii}|^k v \\ &= \rho^k \text{rad}([A]_{pi}^{(m)})v. \end{aligned}$$

Since the right-hand side contains at least one positive component, this component tends to infinity with k by the factor ρ^k . Therefore, the left-hand side is unbounded and so is $([A]^k)$. $\square$

THEOREM 4.7. *Let $[A] = ([A]_{ij})_{i,j=1,\dots,s} \in \mathbf{I}(\mathbb{R}^{n \times n})$ be in reducible normal form with a block column i which has the $*$ -property (cf. Lemma 4.2(c)). Let i be connected to some node $i' > i$ in the block graph $G_B(|[A]|)$. If $\rho(|[A]_{ii}|) = \rho(|[A]_{i'i'}|) = 1$, then the sequence $([A]^k)$ is unbounded. In particular, the matrix $[A]$ cannot be semiconvergent.*

Proof. Let w^T be a left Perron vector of $|[A]_{ii}|$ and v be a right Perron vector of $|[A]_{i'i'}|$. Choose any indices i_0, i'_0 with $i_0 \in \text{ind}(|[A]_{ii}|)$, $i'_0 \in \text{ind}(|[A]_{i'i'}|)$. According to Lemma 3.5 there is a path in $G(|[A]|)$ connecting i_0 with i'_0 . Let c be the length of this path. Then $|[A]_{ii'}|^{(c)} \neq O$ and we get

$$w^T |[A]_{ii'}|^{(k+c)} v \geq w^T \cdot \sum_{l=0}^k |[A]_{ii}|^l |[A]_{ii'}|^{(c)} |[A]_{i'i'}|^{k-l} \cdot v \geq k \cdot w^T |[A]_{ii'}|^{(c)} v > 0.$$

Hence there are some indices $r \in \text{ind}(|[A]_{ii}|)$, $r' \in \text{ind}(|[A]_{i'i'}|)$ for which the sequence $(|[a]_{rr'}|^{(k)})$ is unbounded.

Since the block column i has the $*$ -property, there are integers m, p, q with $q \in \text{ind}(|[A]_{ii}|)$ such that $\text{rad}([a]_{pq}^{(m)}) > 0$. Since $|[A]_{ii}|$ is irreducible, there is a path from q to r in $G(|[A]|)$ whose additional nodes we denote by $q_1, \dots, q_{l-1}$. Now we have

$$\text{rad}([a]_{pr'}^{(m+l+k)}) \geq \text{rad}([a]_{pq}^{(m)}) \cdot |[a]_{qq_1}| \times \dots \times |[a]_{q_{l-1}r}| \cdot |[a]_{rr'}|^{(k)}.$$

All factors in front of $|[a]_{rr'}|^{(k)}$ are positive and independent of k . Therefore, the sequence $(\text{rad}([a]_{pr'}^{(m+l+k)}))$ is unbounded and $[A]$ cannot be semiconvergent. $\square$

In our next theorem we present a condition which is sufficient for the semiconvergence of $[A]$. It uses the semiconvergence of the companion matrix B .

THEOREM 4.8. *Let $[A] = ([A]_{ij})_{i,j=1,\dots,s} \in \mathbf{I}(\mathbb{R}^{n \times n})$ be in reducible normal form and let $B = (B_{ij})_{i,j=1,\dots,s}$ be the companion matrix of $[A]$ with the following property:*

(i) *If $\rho(B_{i'i'}) = \rho(B_{ii}) = 1$ for some $i' < i$ and if the block column with index i has the $*$ -property, then i' is not connected to i in $G_B(B)$.*

Then the matrix $[A]$ is semiconvergent if the matrices B and $[A]_{ii}$, $i = 1, \dots, s$, are semiconvergent.

Proof. We proceed by induction on the number s of block rows; respectively, of block columns.

If $s = 1$, then the assertion is true by the assumption. Now assume that the theorem holds for matrices in reducible normal form with fewer than s block rows and corresponding block columns, and let

$$[A] = \begin{pmatrix} [Q] & [R] \\ O & [A]_{ss} \end{pmatrix}, \quad [R] = \begin{pmatrix} [A]_{1s} \\ \vdots \\ [A]_{s-1,s} \end{pmatrix} =: [R]^{(1)},$$

$$[A]^k = ([A]_{ij}^{(k)}) = \begin{pmatrix} [Q]^k & [R]^{(k)} \\ O & [A]_{ss}^k \end{pmatrix}.$$

Then $[Q]^\infty := \lim_{k \rightarrow \infty} [Q]^k$ and $[A]_{ss}^\infty$ exist by the induction hypothesis and by the hypothesis of the theorem. It remains to show that $[R]^\infty := \lim_{k \rightarrow \infty} [R]^{(k)}$ exists.

In block columns s without the $*$ -property (cf. Lemma 3.1) the semiconvergence follows from $[A]_{is}^{(k)} = B_{is}^{(k)}$, $i = 1, \dots, s$ (cf. [9, Lem. 3]). From now on let the s th block column have the $*$ -property. Then by our hypothesis the matrix $[A]_{ss}$ is semiconvergent.

If $\rho([A]_{ss}) < 1$, the semiconvergence of $[R]^{(k)}$ follows from Lemma 3.10 applied to

$$[R]^{(k+1)} = [Q]^k [R] + [R]^{(k)} [A]_{ss}, \quad k = 1, 2, \dots$$

If $\rho([A]_{ss}) = \rho(B_{ss}) = 1$, then no block index $i' < s$ with $\rho(B_{i'i'}) = 1$ is connected to s because of (i). Thus for the calculation of $[R]^{(k)}$ we can delete in $[A]$ all block rows and block columns i' with $\rho(B_{i'i'}) = 1$. They do not influence the values of the entries of $[R]^{(k)}$. In particular, $[R]_{i's}^{(k)} = [A]_{i's}^{(k)} = O$, $k = 1, 2, \dots$, in this case. Having finished the process of cancellation, one ends up with some reduced matrix

$$[A_r] = \begin{pmatrix} [Q_r] & [R_r] \\ O & [A]_{ss} \end{pmatrix}.$$

Let B_r be the companion matrix of $[Q_r]$. If j is a block column of $[Q]$ without the $*$ -property, then $(B_r)_{jj} = B_{jj}$. If j is a block column of $[Q]$ with the $*$ -property, it might be a block column of $[Q_r]$ without this property because of the cancellations. Therefore, we can state only $|(B_r)_{jj}| = B_{jj}$ in this case, where we used the numbering of B also for B_r . Hence $\rho((B_r)_{jj}) \leq \rho(B_{jj})$ for all block indices j which occur in B_r . Since for B_r we have deleted all blocks $B_{i'i'}$ with $\rho(B_{i'i'}) = 1$, we finally get $\rho(B_r) < 1$, and with Theorem 4.4 we can conclude $\lim_{k \rightarrow \infty} [Q_r]^k = O$; i.e., $[Q_r]$ is convergent. By virtue of Lemma 3.9 the matrix series $\sum_{k=0}^{\infty} |[Q_r]^k|$ is convergent and so is $\sum_{k=0}^{\infty} |[Q_r]^k| \cdot |[R_r]|$. This series is a convergent majorant of the nonnegative matrix series $\sum_{k=0}^{\infty} |[Q_r]^k [R_r]| = \sum_{k=0}^{\infty} q([Q_r]^k [R_r], O)$; hence the latter series is convergent, too. Moreover, $O = O + O[A]_{ss}$, where O and $[R_r]$ have the same size. After these considerations it is easy to see that all assumptions of Lemma 3.11 are fulfilled for the iteration

$$[R_r]^{(k+1)} = [Q_r]^k [R_r] + [R_r]^{(k)} [A]_{ss}, \quad k = 1, 2, \dots$$

Hence $[R]^\infty$ exists and $[A]$ is semiconvergent. $\square$

Note that we cannot omit the assumption (i) in Theorem 4.8. This can be seen by Example 4.13 below. The converse implication in Theorem 4.8 does not hold, as can be seen by Example 4.14. Adding, however, a weak additional condition yields the expected semiconvergence of B . Applying Theorem 4.8 for the necessity finally results in the following necessary and sufficient condition for the semiconvergence of $[A]$.

THEOREM 4.9. *Let $[A] = ([A]_{ij})_{i,j=1,\dots,s} \in \mathbf{I}(\mathbb{R}^{n \times n})$ be in reducible normal form and let $B = (B_{ij})_{i,j=1,\dots,s}$ be the companion matrix of $[A]$. Moreover, let the following two properties hold:*

(i) *If $\rho(B_{i'i'}) = \rho(B_{ii}) = 1$ for some $i' < i$ and if the block column with index i has the $*$ -property, then i' is not connected with i in $G_B(B)$.*

(ii) *If $\text{rad}([A]_{ii}) = O$ and $\rho(B_{ii}) = 1$ and if the block column with index i has the $*$ -property, then B_{ii} is primitive.*

Then $[A]$ is semiconvergent if and only if the matrices B and $[A]_{ii}$, $i = 1, \dots, s$, are semiconvergent.

Proof. By virtue of Theorem 4.8 it remains to prove the implication “ $\Rightarrow$.” Therefore, let $[A]$ be semiconvergent. Then, trivially, $[A]_{ii}$ is semiconvergent for $i = 1, \dots, s$. In block columns i without the $*$ -property the sequence $(B_{pi}^{(k)})$ is convergent for $p = 1, \dots, s$ because of $B_{pi}^{(k)} = [A]_{pi}^{(k)}$ (cf. [9, Lem. 3]).

Now let us consider a block column i with the $*$ -property. If $\text{rad}([A]_{ii}) \neq O$, then $B_{ii} = |[A]_{ii}|$ is semiconvergent by Theorem 4.5. If $\text{rad}([A]_{ii}) = O$, there exists a block index i_0 such that $\text{rad}([A]_{i_0 i}^{(m)}) \neq O$ holds for some m . Let v be a right Perron vector of $[A]_{ii}$. From

$$\text{rad}([A]_{i_0 i}^{(m+k)})v \geq \text{rad}([A]_{i_0 i}^{(m)})|[A]_{ii}|^k v \geq \text{rad}([A]_{i_0 i}^{(m)})\rho([A]_{ii})^k v$$

we can deduce that $\rho([A]_{ii}) \leq 1$. So, together with condition (ii) the matrix $B_{ii} = |[A]_{ii}|$ turns out to be semiconvergent.

It remains to show that the sequences $(B_{pi}^{(k)})$ are convergent for $p = 1, \dots, i-1$. Consider

$$B' = \begin{pmatrix} B_{i-1} & B_{*i} \\ O & B_{ii} \end{pmatrix}, \quad B_{*i} = \begin{pmatrix} B_{1i} \\ \vdots \\ B_{i-1,i} \end{pmatrix} =: B^{(1)}.$$

From $(B')^{k+1} = (B')^k B' = B'(B')^k$ we get

$$(4.1) \quad B_{*i}^{(k+1)} = B_{i-1}^k B_{*i} + B_{*i}^{(k)} B_{ii} = B_{i-1} B_{*i}^{(k)} + B_{*i} B_{ii}^k.$$

We proceed by induction on block columns. Therefore, we can assume that B_{i-1} is semiconvergent. If $\rho(B_{ii}) < 1$, then the sequence $(B_{*i}^{(k)})$ is convergent by Lemma 3.10. If $\rho(B_{ii}) = 1$, then without loss of generality we can assume $\rho(B_{i-1}) < 1$. Otherwise we can delete all block rows and block columns $i' \neq i$ in B' for which $\rho(B_{i'i'}) = 1$ holds because there is no connection of block column i' with block column i by condition (i). Now Lemma 3.10 applied to the second part of (4.1) shows that the sequence $(B_{*i}^{(k)})$ is convergent. $\square$

In our next two theorems we replace the companion matrix B in Theorems 4.8 and 4.9 by the absolute value $|[A]|$ in order to get similar results.

THEOREM 4.10. *Let $[A] = ([A]_{ij})_{i,j=1,\dots,s}$ be in reducible normal form and let the following two properties hold:*

- (i) The matrix $|[A]|$ is semiconvergent.
- (ii) The matrices $[A]_{ii}$, $i = 1, \dots, s$, are semiconvergent.

Then both the matrix $[A]$ and its companion matrix B are semiconvergent.

Proof. The companion matrix B of $[A]$ satisfies $B^k \leq |[A]|^k$; hence the sequence (B^k) is bounded. The diagonal blocks of B are semiconvergent because they are equal to the diagonal blocks either of $[A]$ or of $|[A]|$, which are both semiconvergent. So with Theorem 3.3 the companion matrix B turns out to be semiconvergent. Moreover, B fulfills assumption (i) of Theorem 4.8. This can be seen by applying Theorem 3.6 to $|[A]|$ and by taking into account $\rho(B_{ii}) \leq \rho(|[A]|_{ii}) \leq 1$. Since we already know that B is semiconvergent, Theorem 4.8 concludes the proof. $\square$

Now we present a situation where the semiconvergence of $|[A]|$ follows from the semiconvergence of $[A]$. Combined with Theorem 4.10 this yields the following necessary and sufficient criterion for the semiconvergence of $[A]$.

THEOREM 4.11. *Let $[A] = ([A]_{ij})_{i,j=1,\dots,s}$ be in reducible normal form and let the following two properties hold:*

- (i) *If $\rho(|[A]|_{ii}) = 1$, then $|[A]|_{ii}$ is primitive.*
- (ii) *All columns of $[A]$ for which $\rho(|[A]|_{ii}) \geq 1$ have the *-property.*

Then $[A]$ is semiconvergent if and only if the matrices $|[A]|$ and $[A]_{ii}$, $i = 1, \dots, s$, are semiconvergent.

Proof. Let $[A]$ be semiconvergent. Then, trivially, $[A]_{ii}$ is semiconvergent for $i = 1, \dots, s$. If $\text{rad}([A]_{ii}) \neq O$, the semiconvergence of $|[A]|_{ii}$ follows from Theorem 4.5. In case of $\text{rad}([A]_{ii}) = O$ we get $\rho(|[A]|_{ii}) \leq 1$ by (ii) and Theorem 4.6, and the semiconvergence of $|[A]|_{ii}$ follows from (i) and Theorem 3.4.

Since every block column i with $\rho(|[A]|_{ii}) = 1$ has the *-property and since Theorem 4.7 holds, there is no connection of i with a node $i' > i$ in the block graph $G_B(|[A]|)$ such that $\rho(|[A]|_{i'i'}) = 1$. Now Theorem 3.6 shows that $|[A]|$ is semiconvergent, which concludes the implication “ $\Rightarrow$ ” of the proof.

The converse implication follows from Theorem 4.10, which does not need assumptions (i), (ii) of the present Theorem 4.11. $\square$

THEOREM 4.12. *Let $[A] = ([A]_{ij})_{i,j=1,\dots,s}$ be in reducible normal form with*

$$(4.2) \quad \rho(|[A]|_{ii}) < 1 \quad \text{if} \quad \text{rad}([A]_{ii}) = O.$$

Then $[A]$ is semiconvergent if and only if the matrices $|[A]|$ and $[A]_{ii}$, $i = 1, \dots, s$, are semiconvergent.

Although this theorem was already published in [3] we prove it once more now by applying the preceding theorems. This shortens the original proof drastically.

Proof of Theorem 4.12. Let $[A]$ be semiconvergent. We will show that (4.2) implies conditions (i) and (ii) of Theorem 4.11. If $\rho(|[A]|_{ii}) \geq 1$, then $\text{rad}([A]_{ii}) \neq O$ by (4.2); i.e., block column i has the *-property. Furthermore $|[A]|_{ii}$ is semiconvergent by Theorem 4.5; hence in the case $\rho(|[A]|_{ii}) = 1$ it is primitive. Therefore, the conditions (i) and (ii) of Theorem 4.11 are fulfilled, which proves the necessity. The sufficiency follows from Theorem 4.10 without using (4.2). $\square$

As with assumptions (i) and (ii) in Theorem 4.9, the additional assumptions (i), (ii) in Theorem 4.11 and (4.2) in Theorem 4.12 cannot be dropped. This can be seen by the example

$$[A] \equiv A = \begin{pmatrix} 1 & -1 \\ 1 & -1 \end{pmatrix},$$

where $\rho(A) = 0$, while $\rho(|A|) = 2$. Moreover, this example shows that the companion matrix B of $[A]$ can be semiconvergent, while $|[A]|$ is not.

Note that in Theorems 4.8–4.12 the assumption

$$(4.3) \quad \text{the matrices } [A]_{ii}, \ i = 1, \dots, s, \text{ are semiconvergent}$$

can be replaced by the following:

$$(4.4) \quad \text{If } \rho(B_{ii}) = 1 \text{ and if } [A]_{ii} \text{ contains only one matrix } \dot{A}_{ii} \text{ with } |\dot{A}_{ii}| = |[A]_{ii}|, \text{ then} \\ \dot{A}_{ii} \neq -D|[A]_{ii}|D \text{ for any signature matrix } D \text{ of the size of } |[A]_{ii}|.$$

If $\text{rad}([A]_{ii}) \neq O$, this follows directly from Theorems 4.4 and 4.5. If $\text{rad}([A]_{ii}) = O$, i.e., $[A]_{ii} \equiv A_{ii}$, we first consider the implication $(4.3) \Rightarrow (4.4)$. To this end let A_{ii} be semiconvergent and assume $\rho(B_{ii}) = 1$, $A_{ii} = -D|A_{ii}|D$, $|D| = I$. Then A_{ii} must have an eigenvalue λ , which satisfies $\lambda = -\rho(|A_{ii}|) \leq -\rho(B_{ii}) = -1$. Hence A_{ii} cannot be semiconvergent by virtue of Theorem 3.2. This is a contradiction; hence $A_{ii} \neq -D|A_{ii}|D$.

In order to prove the implication $(4.4) \Rightarrow (4.3)$ in the degenerate case $[A]_{ii} \equiv A_{ii}$ we need the corresponding additional condition of our theorems; i.e., we assume that B and $|[A]|$ are semiconvergent. Together with

$$(4.5) \quad \rho(A_{ii}) \leq \rho(B_{ii}) \leq \rho(|A_{ii}|)$$

this implies $\rho(A_{ii}) \leq 1$ and

$$(4.6) \quad \rho(B_{ii}) \leq 1 \quad \text{or} \quad \rho(|A_{ii}|) \leq 1.$$

If $\rho(A_{ii}) < 1$, then the matrix A_{ii} is semiconvergent. Now let $\rho(A_{ii}) = 1$ until the end of this proof. Then $\rho(B_{ii}) = 1$ by (4.5) and (4.6). Hence

$$(4.7) \quad A_{ii} \neq -D|A_{ii}|D, \quad |D| = I,$$

by assumption (4.4). If $B_{ii} = |A_{ii}|$ and if B or $|[A]|$ are semiconvergent, then $|[A]_{ii}|$ is semiconvergent, whence

$$(4.8) \quad \rho(|A_{ii}|) = \rho(B_{ii}) = 1.$$

Therefore, by virtue of Lemma 5 in [10], together with (4.7), we obtain $A_{ii} = D|A_{ii}|D$, $|D| = I$, D real. Hence A_{ii} is semiconvergent. If $B_{ii} \neq |[A]_{ii}|$ and if B is semiconvergent, then trivially $A_{ii} = B_{ii}$ has this latter property. If $B_{ii} \neq |[A]_{ii}|$ and if $|[A]|$ is semiconvergent, then (4.8) holds again and the semiconvergence is proved as in the lines below (4.8). This shows the remaining part of the equivalence.

We now formulate the situations still to be studied in view of the semiconvergence of $[A]$. To this end we assume that at least one of assumptions (i), (ii) of Theorem 4.9 is violated. If (i) is false, although its hypotheses hold, then there must be two indices i' , i such that $\rho(B_{i'i'}) = \rho(B_{ii}) = 1$ for two diagonal blocks of the companion matrix B of $[A]$, the block column with index i has the $*$ -property, and $i' < i$ is connected to i in $G_B(B)$. Since we want to find *criteria for the semiconvergence* of $[A]$ we can exclude the case where the block column with index i' has the $*$ -property, too, since then $\rho(|[A]_{ii}|) = \rho(B_{ii}) = 1 = \rho(B_{i'i'}) = \rho(|[A]_{i'i'}|)$, which prevents $[A]$ from being semiconvergent by Theorem 4.7. For the remaining case we have $B_{ii} = |[A]_{ii}|$ and $B_{i'i'} = [A]_{i'i'} \equiv A_{i'i'}$, in particular, $\rho(|[A]_{ii}|) = 1 = \rho(A_{i'i'})$. Now $B_{ii} = |[A]_{ii}|$ can be primitive or cyclic of some index h . The first case yields (A) of the two assumptions

below, taking into account $G_B(B) = G_B(|[A]|)$. In the second case, $B_{ii} = |[A]_{ii}|$ cannot be semiconvergent by Theorem 3.4. Therefore, if $[A]$ is semiconvergent, we must have $\text{rad}([A]_{ii}) = O$, since otherwise Theorem 4.5 applies and prevents $[A]_{ii}$ from being semiconvergent. But if $[A]_{ii} \equiv A_{ii}$ with $\rho(B_{ii}) = \rho(|[A]_{ii}|) = 1$, assumption (ii) of Theorem 4.9 is violated, which yields (B) of our subsequent assumptions.

Summarizing and bearing in mind Theorem 4.9, there remain the following situations to be studied in view of the semiconvergence of $[A]$.

ASSUMPTIONS. Let $[A] = ([A]_{ij})_{i,j=1,\dots,s}$ be in reducible normal form with at least one block column i which has the $*$ -property and which fulfills at least one of the following two conditions:

(A) $\rho(|[A]_{ii}|) = 1$ and $[A]_{ii}$ is primitive and there is at least one additional block column with index $i' < i$ which does not have the $*$ -property but which is connected to i in $G_B(|[A]|)$ and fulfills $\rho(A_{i'i'}) = 1$.

(B) $[A]_{ii} \equiv A_{ii}$ and $\rho(|A_{ii}|) = 1$ and $|A_{ii}|$ is cyclic of some index h .

Moreover, assume that all diagonal blocks A_{ii} , $i = 1, \dots, s$, are semiconvergent.

Both situations can lead to convergence or to divergence of the sequence $([A]^k)$ as will be shown by the examples below. If, however, none of the new assumptions holds and if the sequence $([A]^k)$ should not be divergent in advance (cf. Theorems 4.6 and 4.7) then the arguments above show that assumptions (i), (ii) of Theorem 4.9 must be fulfilled, whence the semiconvergence of $[A]$ is completely clarified.

In both cases still to be studied the semiconvergence of the companion matrix B of $[A]$ is not necessary for the semiconvergence of $[A]$. In case (B) the cyclicity of $B_{ii} = |[A]_{ii}|$, together with $\rho(B_{ii}) = 1$, even prevents it from being semiconvergent. The following examples illustrate both assumptions. Here the matrices

$$(4.9) \quad C = \begin{pmatrix} 1/2 & -1/2 \\ 1/2 & -1/2 \end{pmatrix} \quad \text{and} \quad [C] = [-|C|, |C|]$$

play a crucial role. It is easy to see that the relations

$$C^2 = O, \quad C|C| = O, \quad |C|C = C, \quad |C|C^T = O, \\ C^T|C| = C^T, \quad |C|^2 = |C|, \quad [C]^2 = [C]$$

hold and, in particular, $\rho(C) = 0 < 1 = \rho(|C|)$.

Example 4.13. Let

$$[A] = \begin{pmatrix} |C| & C^T \\ O & [C] \end{pmatrix} \in \mathbb{I}(\mathbb{R}^{4 \times 4})$$

with C , $[C]$ as in (4.9). Then

$$[A]^k = \begin{pmatrix} |C| & [C] \\ O & [C] \end{pmatrix}, \quad k = 2, 3, \dots,$$

while its companion matrix

$$B = \begin{pmatrix} |C| & |C| \\ O & |C| \end{pmatrix} \in \mathbb{R}^{4 \times 4}$$

yields

$$B^k = \begin{pmatrix} |C| & k|C| \\ O & |C| \end{pmatrix}, \quad k = 2, 3, \dots$$

Hence $[A]$ fulfills assumption (A) above and is semiconvergent, while B is not.

In contrast to $[A]$ the matrix

$$[\tilde{A}] = \begin{pmatrix} |C| & [C] \\ O & [C] \end{pmatrix} \in \mathbf{I}(\mathbb{R}^{4 \times 4})$$

still satisfies (A) but has divergent powers

$$[\tilde{A}]^k = \begin{pmatrix} |C| & k[C] \\ O & [C] \end{pmatrix}, \quad k = 2, 3, \dots$$

In passing we note that the matrix

$$[\hat{A}] = \begin{pmatrix} D|C|D & [C] \\ O & |C| \end{pmatrix} \in \mathbf{I}(\mathbb{R}^{4 \times 4})$$

with $D = \text{diag}(1, -1)$ implies

$$[\hat{A}]^k = \begin{pmatrix} D|C|D & k[C] \\ O & |C| \end{pmatrix},$$

while its companion matrix $\hat{B}$ satisfies

$$\hat{B}^k = \begin{pmatrix} D|C|D & |C| \\ O & |C| \end{pmatrix}, \quad k = 1, 2, \dots$$

Hence $\hat{B}$ is semiconvergent, while $[\hat{A}]$ is not. If, however,

$$[\tilde{A}] = \begin{pmatrix} D|C|D & |C| \\ O & [C] \end{pmatrix} \in \mathbf{I}(\mathbb{R}^{4 \times 4})$$

with the same diagonal matrix D as in $[\hat{A}]$, then

$$[\tilde{A}]^k = \begin{pmatrix} D|C|D & [C] \\ O & [C] \end{pmatrix},$$

while the companion matrix $\tilde{B}$ satisfies $\tilde{B} = \hat{B}$, whence $[\tilde{A}]$ and $\tilde{B}$ are semiconvergent simultaneously. Therefore, with assumption (A) all cases can occur: Both $[A]$ and its companion matrix B are semiconvergent, exactly one of them is semiconvergent, or none is.

Example 4.14. Let

$$[A] = \begin{pmatrix} 0 & [-1, 1] & [-1, 1] & [a] & [a] \\ 0 & & O & C & \\ 0 & & C & O & \end{pmatrix} \in \mathbf{I}(\mathbb{R}^{5 \times 5})$$

with C as in (4.9). If $[a] = [-1, 1]$, then

$$[A]^k = \begin{pmatrix} 0 & [-1, 1] & [-1, 1] & [-1, 1] & [-1, 1] \\ 0 & & O & O & \end{pmatrix} \in \mathbf{I}(\mathbb{R}^{5 \times 5}), \quad k = 2, 3, \dots,$$

while its companion matrix

$$B = \begin{pmatrix} 0 & 1 & 1 & 1 & 1 \\ 0 & O & |C| & & \\ 0 & |C| & O & & \end{pmatrix} \in \mathbb{R}^{5 \times 5}$$

yields

$$B^k = \begin{pmatrix} 0 & 1 & 1 & 1 & 1 \\ 0 & |C| & & O & \\ 0 & O & & |C| & \end{pmatrix}$$

if k is even and $B^k = B$ if k is odd. Again $[A]$ is semiconvergent, while B is not. Note that $[A]$ fulfills assumption (B) above with $h = 2$.

The choice $[a] = 0$ yields

$$[A]^k = \begin{pmatrix} 0 & 0 & 0 & [-1, 1] & [-1, 1] \\ 0 & O & & O & \end{pmatrix} \in \mathbb{I}(\mathbb{R}^{5 \times 5})$$

if $k \geq 2$ is even and

$$[A]^k = \begin{pmatrix} 0 & [-1, 1] & [-1, 1] & 0 & 0 \\ 0 & O & O & O & \end{pmatrix} \in \mathbb{I}(\mathbb{R}^{5 \times 5})$$

if $k \geq 2$ is odd. Hence this time $[A]$ is not semiconvergent although it satisfies assumption (B) again.

Both examples show that the situation becomes difficult if at least one of assumptions (A) and (B) holds. Therefore, we will take a closer look at the class of matrices which fulfill one of them. However, we are not able to discuss the problem of semiconvergence fully. We list only some ideas which may help in finding criteria that guarantee the semiconvergence in these critical cases. We first treat a situation with assumption (A), which may be helpful in an inductive procedure.

THEOREM 4.15. *Let*

$$[A] = \begin{pmatrix} A_{11} & [R] \\ O & [Q] \end{pmatrix} = ([A]_{ij})_{i,j=1,\dots,s}$$

be in reducible normal form with semiconvergent diagonal blocks $[A]_{ii}$, $i = 1, \dots, s$. Moreover, let $\rho(A_{11}) = 1$, $\rho([A]_{ss}) = 1$, $[A]_{ss}$ primitive. Let the s th block column of $[A]$ have the $*$ -property. If each row of $[A]_{1s}$ has at least one nondegenerate entry, then $([A]^k)$ is unbounded; hence $[A]$ cannot be semiconvergent.

Proof. Define the vector $\begin{pmatrix} 0 \\ x \end{pmatrix}$ such that its number of entries coincides with the number of columns of $[R]$ and that x is a Perron vector of the matrix $[A]_{ss}$. In particular, $[A]_{ss}x = x$ and $[Q]z \geq z$. From $[R]^{(k+1)} = A_{11}^k[R] + [R]^{(k)}[Q]$ (with notation similar to that in the proof of Theorem 4.8) one gets

$$\text{rad}([R]^{(k+1)}) \geq |A_{11}^k| \text{rad}([R]) + \text{rad}([R]^{(k)})|[Q]|,$$

whence

$$\text{rad}([R]^{(k+1)})z \geq |A_{11}^k| \text{rad}([R]) \begin{pmatrix} 0 \\ x \end{pmatrix} + \text{rad}([R]^{(k)})z = |A_{11}^k| \text{rad}([A]_{1s})x + \text{rad}([R]^{(k)})z.$$

By induction one obtains

$$\text{rad}([R]^{(k+1)})z \geq \sum_{j=0}^k |A_{11}^j| \text{rad}([A]_{1s})x.$$

Since $(|A_{11}^k|)$ does not tend to the zero matrix and since $\text{rad}([A]_{1s})x$ is positive by the assumptions, the summands in the series do not tend to zero as k approaches infinity.

Therefore, the series is divergent, and hence the sequence $(\text{rad}([R]^{(k)}))$ is unbounded and the theorem is proved. $\square$

Note that the assumption “ $[A]_{ss}$ primitive” was only stated in Theorem 4.15 in order to be compatible with assumption (A) above. In fact it is not needed for proving the assertion of this theorem. The indices 1 and s play the roles of i' and i in assumption (A). The connection of i' with i is realized by the edge $1 \rightarrow s$.

We next consider assumption (B). In view of an inductive procedure we deal with the following simple situation: Let

$$(4.10) \quad [A] = \begin{pmatrix} [Q] & [R] \\ O & [A]_{ss} \end{pmatrix},$$

where the s th block column has the $*$ -property. Moreover, assume that $[Q]$ and $[A]_{ss}$ are semiconvergent with convergent matrix $[Q]$ and $\rho(|[A]_{ss}|) = 1$. Hence $[Q]^\infty = O$ and assumption (A) cannot hold. Therefore, let $[A]_{ss} \equiv A_{ss}$ with $|A_{ss}|$ being cyclic of index h such that (B) is true. Example 4.14 illustrates this situation. Since A_{ss} is assumed to be semiconvergent the cyclicity of $|A_{ss}|$ implies $\rho(A_{ss}) < 1$; hence $A_{ss}^\infty = O$. Therefore, $[A]$ is here semiconvergent if and only if $[R]^\infty = \lim_{k \rightarrow \infty} [R]^{(k)}$ exists. Assume for the moment that $[A]^\infty$ exists and use notation similar to that in the proof of Theorem 4.8. Then the recursion

$$(4.11) \quad [R]^{(k+1)} = [Q]^k [R] + [R]^{(k)} A_{ss}$$

implies $[R]^\infty = [R]^\infty A_{ss}$, whence

$$(4.12) \quad \check{R}^\infty = \check{R}^\infty A_{ss}$$

for the midpoint $\check{R}^\infty$ of $[R]^\infty$ and

$$(4.13) \quad \text{rad}([R]^\infty) = \text{rad}([R]^\infty |A_{ss}|)$$

for its radius. From (4.12) and (4.13) we get $\check{R}^\infty = \check{R}^\infty A_{ss}^k \rightarrow O$ for $k \rightarrow \infty$ and

$$(4.14) \quad \text{rad}([R]^\infty) = zy^T$$

with a fixed left Perron vector y of $|A_{ss}|$ and an appropriate vector $z \geq 0$. Therefore, in case of semiconvergence of $[A]$ we finally obtain

$$[R]^\infty = zy^T[-1, 1], \quad z \geq 0, \quad y > 0, \quad y^T |A_{ss}| = y^T.$$

Unfortunately, Example 4.14 shows that $[A]^\infty$ does not always exist under the assumptions above. Lemma 3.11 cannot be applied to (4.11) since $|A_{ss}|$ is not semiconvergent. However, the last part of Lemma 3.10 can be used for the midpoint relation

$$\check{R}^{(k+1)} = \text{mid}([Q]^k [R]) + \check{R}^{(k)} A_{ss}$$

of (4.11). It shows that

$$\lim_{k \rightarrow \infty} \check{R}^{(k)} = O$$

remains true even if $[A]^\infty$ does not exist. Therefore, $[A]$ is semiconvergent if and only if the sequence $(\text{rad}([R]^{(k)}))$ is convergent. We also note that the sequence $(|[A]^k|)$ is

bounded. This can be deduced by repeated application of (4.11) and of the absolute value

$$\begin{aligned}
 (4.15) \quad |[R]^{(k+1)}| &\leq |[Q]^k| \cdot |[R]| + |[R]^{(k)}| \cdot |A_{ss}| \leq \cdots \\
 &\leq \sum_{j=0}^k |[Q]^{k-j}| \cdot |[R]| \cdot |A_{ss}|^j \leq \sum_{j=0}^k |[Q]^{k-j}| \cdot |[R]| \cdot K \\
 &\leq \left(\sum_{j=0}^{\infty} |[Q]^j| \right) \cdot |[R]| \cdot K.
 \end{aligned}$$

Here we exploited the fact that the h eigenvalues λ of $|A_{ss}|$ with $|\lambda| = \rho(A_{ss})$ are simple. Hence the powers $|A_{ss}|^k$ are bounded by some constant matrix $K \geq O$. The series in the last expression is convergent according to Lemma 3.9. This proves that $(|[R]^{(k)}|)$ is bounded and so are the powers $|[A]^k|$. Moreover, it guarantees that $([R]^{(k)})$ has a convergent subsequence $([R]^{(k_i)})$ with some limit $[\tilde{R}]^\infty$.

Now we study the very particular case $[Q] \equiv Q$ of (4.10).

THEOREM 4.16. *Let*

$$[A] = \begin{pmatrix} Q & [R] \\ O & A_{ss} \end{pmatrix} = ([A]_{ij})_{i,j=1,\dots,s}$$

be in reducible normal form with

$$\text{rad}([R]) \neq O, \quad \rho(Q) < 1, \quad \rho(A_{ss}) < 1, \quad \rho(|A_{ss}|) = 1, \quad |A_{ss}| \text{ cyclic of index } h.$$

Let x be a fixed right Perron vector of $|A_{ss}|$ and construct with it the vectors $x^{(j)}$, $j = 1, \dots, h$, of Definition 3.7 (with $A = |A_{ss}|$).

If

$$(4.16) \quad \text{rad}([R]) x^{(j)} = c, \quad j = 1, \dots, h,$$

with some constant vector c , then $[A]^\infty$ exists.

Conversely, if $[A]^\infty$ exists and if, in addition,

$$\begin{aligned}
 (4.17) \quad Q &= e^{i\theta} D |Q| D^{-1} \in \mathbb{R}^{n \times n}, \quad D = \text{diag}(e^{i\theta_1}, \dots, e^{i\theta_n}) \in \mathbb{C}^{n \times n}, \\
 \theta, \theta_j &\in [0, 2\pi), \quad i^2 = -1,
 \end{aligned}$$

then condition (4.16) holds.

Proof. Starting with (4.11) we get

$$\begin{aligned}
 (4.18) \quad \text{rad}([R]^{(k+1)}) &= |Q|^k \text{rad}([R]) + \text{rad}([R]^{(k)}) |A_{ss}| = \cdots = \sum_{j=0}^k |Q|^j \text{rad}([R]) |A_{ss}|^{k-j} \\
 &= \sum_{j=0}^{k_0} |Q|^j \text{rad}([R]) |A_{ss}|^{k-j} + \sum_{j=k_0+1}^k |Q|^j \text{rad}([R]) |A_{ss}|^{k-j},
 \end{aligned}$$

where $k_0 < k$ is some positive integer. The last sum in (4.18) can be made arbitrarily small if one chooses k_0 sufficiently large since

$$\sum_{j=k_0+1}^k |Q|^j \text{rad}([R]) |A_{ss}|^{k-j} \leq \left(\sum_{j=k_0+1}^{\infty} |Q|^j \right) \text{rad}([R]) K$$

with K as in (4.15). Assuming k_0 is chosen, let $k = k_0 + lh + r$ with $r \in \{0, 1, \dots, h-1\}$. Fix r and let k tend to infinity. With the notation of Definition 3.7, with Lemma 3.8 (d) and with (4.16) we obtain

$$\begin{aligned} & \lim_{k \rightarrow \infty} \left(\sum_{j=0}^{k_0} |Q^j| \operatorname{rad}([R]) |A_{ss}|^{k-j} \right) \\ &= \lim_{l \rightarrow \infty} \left(\sum_{j=0}^{k_0} |Q^j| \operatorname{rad}([R]) |A_{ss}|^{lh} |A_{ss}|^{k_0+r-j} \right) = \sum_{j=0}^{k_0} |Q^j| \operatorname{rad}([R]) |A_{ss}|^{h \cdot \infty} |A_{ss}|^{k_0+r-j} \\ &= \sum_{j=0}^{k_0} |Q^j| \operatorname{rad}([R]) h \sum_{i=1}^h x^{(i)} (y^{(i)})^T |A_{ss}|^{k_0+r-j} = h \sum_{j=0}^{k_0} |Q^j| \sum_{i=1}^h c(y^{(i)})^T |A_{ss}|^{k_0+r-j} \\ &= h \sum_{j=0}^{k_0} |Q^j| cy^T |A_{ss}|^{k_0+r-j} = h \left(\sum_{j=0}^{k_0} |Q^j| \right) cy^T. \end{aligned}$$

Since the last expression is independent of r and since k_0 was large but arbitrary, we finally get

$$\lim_{k \rightarrow \infty} \operatorname{rad}([R]^{(k)}) = h \left(\sum_{j=0}^{\infty} |Q^j| \right) cy^T;$$

in particular, $[A]^\infty$ exists.

In order to prove the converse, we first note that (4.17) implies $|Q^k| = |Q|^k$, $k = 1, 2, \dots$. We get

$$\begin{aligned} \operatorname{rad}([R]^{(k+1)}) &= \sum_{j=0}^k |Q^j| \operatorname{rad}([R]) |A_{ss}|^{k-j} \\ &= \operatorname{rad}([R]) |A_{ss}|^k + \sum_{j=1}^k |Q| |Q|^{j-1} \operatorname{rad}([R]) |A_{ss}|^{k-j+1-1} \\ &= \operatorname{rad}([R]) |A_{ss}|^k + |Q| \sum_{j=0}^{k-1} |Q^j| \operatorname{rad}([R]) |A_{ss}|^{k-1-j} \\ &= \operatorname{rad}([R]) |A_{ss}|^k + |Q| \operatorname{rad}([R]^{(k)}). \end{aligned}$$

Choose $k = lh$, let l tend to infinity, and apply Lemma 3.8. This results in

$$\begin{aligned} \operatorname{rad}([R]^\infty) &= \operatorname{rad}([R]) |A_{ss}|^{h \cdot \infty} + |Q| \operatorname{rad}([R]^\infty) \\ &= \operatorname{rad}([R]) h \sum_{i=1}^h x^{(i)} (y^{(i)})^T + |Q| \operatorname{rad}([R]^\infty), \end{aligned}$$

and after a multiplication with $x^{(j)}$, we obtain

$$(I - |Q|) \operatorname{rad}([R]^\infty) x^{(j)} = \operatorname{rad}([R]) x^{(j)}.$$

Using (4.14) and Lemma 3.8 once more finally yields

$$\operatorname{rad}([R]) x^{(j)} = (I - |Q|) z y^T x^{(j)} = (I - |Q|) z (y^{(j)})^T x^{(j)} = \frac{1}{h} (I - |Q|) z =: c.$$

Since j was arbitrary this concludes the proof. $\square$

Note that according to Lemma 3.8(g), assumption (4.16) can be reformulated equivalently in the following way:

No left eigenvector $w^{(j)}$, $j = 2, \dots, h$, of $|A_{ss}|$ (cf. Definition 3.7) is needed in order to represent the rows of $\text{rad}([R])$ in the basis of left eigenvectors and left principal vectors given in Definition 3.7 for $A := |A_{ss}|$.

At the moment we do not know whether the additional assumption (4.17) can be dropped for the converse part of Theorem 4.16. We suppose so, but cannot prove it. Trivially, this assumption implies $\rho(Q) = \rho(|Q|)$. Note that $D \in \mathbb{C}^{n \times n} \setminus \mathbb{R}^{n \times n}$ can occur in (4.17); consider the example

$$Q = \begin{pmatrix} 0 & -1/2 \\ 1/2 & 0 \end{pmatrix} = i \begin{pmatrix} i & 0 \\ 0 & 1 \end{pmatrix} \begin{pmatrix} 0 & 1/2 \\ 1/2 & 0 \end{pmatrix} \begin{pmatrix} -i & 0 \\ 0 & 1 \end{pmatrix}.$$

If $\text{rad}([Q]) \neq O$, things are much more difficult, as can be seen by the following example.

Example 4.17. Let

$$[A] = \begin{pmatrix} [0, 1/2] & [2, 4] & [-4, -2] & [-1, 1] & [-1, 1] \\ 0 & O & C & C & O \\ 0 & C & O & O & O \end{pmatrix} \in \mathbb{I}(\mathbb{R}^{5 \times 5})$$

with C as in (4.9). Then

$$[a]_{11}^{(2)} = [0, 1/4], \quad [a]_{12}^{(2)} = -[a]_{13}^{(2)} = [-1, 3], \quad [a]_{14}^{(2)} = [a]_{15}^{(2)} = [-3/2, 3/2].$$

It can easily be seen by induction that

$$(4.19) \quad O \in [A]^k, \quad [a]_{11}^{(k)} = [0, 1/2^k], \quad [a]_{12}^{(k)} = -[a]_{13}^{(k)}, \quad [a]_{14}^{(k)} = -[a]_{15}^{(k)} = [a]_{15}^{(k)}$$

holds for $k \geq 1$ with the exception of $O \in [A]^k$, which is true for $k \geq 2$. The other entries of $[A]^k$ are zero for $k \geq 2$. Therefore, $[A]^k$ is completely determined for $k \geq 2$ if we know the entries $\underline{a}_{12}^{(k)}$, $\bar{a}_{12}^{(k)}$, and $\bar{a}_{14}^{(k)}$. With (4.19) we get the recursions

$$(4.20) \quad \underline{a}_{12}^{(k+1)} = -\bar{a}_{14}^{(k)}, \quad k = 1, 2, \dots,$$

$$(4.21) \quad \bar{a}_{12}^{(k+1)} = 4/2^k + \bar{a}_{14}^{(k)}, \quad k = 1, 2, \dots,$$

$$(4.22) \quad \bar{a}_{14}^{(k+1)} = 1/2^k + (\bar{a}_{12}^{(k)} - \underline{a}_{12}^{(k)})/2 = 5/2^k + \bar{a}_{14}^{(k-1)}, \quad k = 2, 3, \dots,$$

where we used (4.20), (4.21) with $k-1$ instead of k for the last equality in (4.22). Applying (4.22) repeatedly yields

$$\bar{a}_{14}^{(k)} = \{5/2^{k-1} + 5/2^{k-3} + 5/2^{k-5} + \dots + 5/2^{r+1}\} + \bar{a}_{14}^{(r)}$$

with $r = 2$ if $k = 2l$, $l \geq 2$, and $r = 1$ if $k = 2l + 1$, $l \geq 1$. Hence

$$\lim_{l \rightarrow \infty} \bar{a}_{14}^{(2l)} = 7/3 = 2.\bar{3}, \quad \lim_{l \rightarrow \infty} \bar{a}_{14}^{(2l+1)} = 8/3 = 2.\bar{6}.$$

This shows that $[A]^\infty$ does not exist although (4.16) holds with $x = e$ and although assumption (B) is satisfied.

It is one of our future projects to find criteria which are more general than those in Theorems 4.15 and 4.16 so that semiconvergence of $[A]$ is guaranteed under assumptions (A) and (B).

5. Conclusion. Under mild assumptions on an interval matrix $[A]$, necessary and sufficient criteria are presented for the semiconvergence of $[A]$. In particular, for matrices in reducible normal form such criteria could be proved. Here, the semiconvergence of the diagonal blocks, together with that of some comparison matrices, plays a crucial role. Only two cases remained open of which some simpler subclasses could be studied in greater detail. Examples illustrate the various difficulties which can occur.

REFERENCES

- [1] G. ALEFELD AND J. HERZBERGER, *Introduction to Interval Computations*, Academic Press, New York, 1983.
- [2] G. ALEFELD AND G. MAYER, *Enclosing solutions of singular interval systems iteratively*, Reliable Computing, 11 (2005), pp. 165–190.
- [3] H.-R. ARNDT AND G. MAYER, *On the semi-convergence of interval matrices*, Linear Algebra Appl., 393 (2004), pp. 15–37.
- [4] A. BERMAN AND R. J. PLEMMONS, *Nonnegative Matrices in the Mathematical Sciences*, Classics in Appl. Math. 9, SIAM, Philadelphia, 1994.
- [5] D. CORDES, *Verifizierter Stabilitätsnachweis für Lösungen von Systemen periodischer Differentialgleichungen auf dem Rechner mit Anwendungen*, Ph.D. Thesis, Universität Karlsruhe, Karlsruhe, Germany, 1987.
- [6] R. A. HORN AND C. R. JOHNSON, *Matrix Analysis*, Cambridge University Press, Cambridge, UK, 1994.
- [7] D. W. JORDAN AND P. SMITH, *Nonlinear Differential Equations*, 2nd ed., Clarendon Press, Oxford, UK, 1987.
- [8] R. LOHNER, *Einschließung der Lösung gewöhnlicher Anfangs—und Randwertaufgaben und Anwendungen*, Ph.D. Thesis, Universität Karlsruhe, Karlsruhe, Germany, 1988.
- [9] G. MAYER, *On the convergence of powers of interval matrices*, Linear Algebra Appl., 58 (1984), pp. 201–216.
- [10] G. MAYER, *On the convergence of powers of interval matrices (2)*, Numer. Math., 46 (1985), pp. 69–83.
- [11] G. MAYER, *On the convergence of the Neumann series in interval analysis*, Linear Algebra Appl., 65 (1985), pp. 63–70.
- [12] O. MAYER, *Über die in der Intervallrechnung auftretenden Räume und einige Anwendungen*, Ph.D. Thesis, Universität Karlsruhe, Karlsruhe, Germany, 1968.
- [13] A. NEUMAIER, *Interval Methods for Systems of Equations*, Cambridge University Press, Cambridge, UK, 1990.
- [14] R. RIHM, *Interval methods for initial value problems in ODEs*, in Topics in Validated Computations, J. Herzberger, ed., Elsevier, Amsterdam, 1994, pp. 173–207.
- [15] R. S. VARGA, *Matrix Iterative Analysis*, 2nd ed., Springer, Berlin, 2000.

MEANS OF POSITIVE NUMBERS AND MATRICES*

DÉNES PETZ[†] AND RÓBERT TEMESI[†]

Dedicated to Professor Pál Rózsa on the occasion of his 80th birthday

Abstract. It is shown that any two-variable-mean of positive matrices (studied by Kubo and Ando) can be extended to more variables. The n -variable-mean $M_n(A_1, A_2, \dots, A_n)$ is defined by a symmetrization procedure when the n -tuple $(A_1, A_2, \dots, A_n)$ is ordered, is monotone in each variable, and satisfies the transformer inequality. This approach is motivated by the paper of Ando, Li, and Mathias on geometric means [*Linear Algebra Appl.*, 385 (2004), pp. 305–334]. Special attention is paid to the logarithmic mean. It is conjectured that for matrix means the symmetrization procedure converges for all triplets.

Key words. matrix mean, geometric mean, logarithmic mean

AMS subject classifications. 47A64, 15A45

DOI. 10.1137/050621906

1. Introduction. A function $M : \mathbb{R}^+ \times \mathbb{R}^+ \rightarrow \mathbb{R}^+$ may be called a *mean* of positive numbers if the following hold:

- (i) $M(x, x) = x$ for every $x \in \mathbb{R}^+$.
- (ii) $M(x, y) = M(y, x)$ for every $x, y \in \mathbb{R}^+$.
- (iii) If $x < y$, then $x < M(x, y) < y$.
- (iv) If $x < x'$ and $y < y'$, then $M(x, y) < M(x', y')$.
- (v) $M(x, y)$ is continuous.

The *geometric mean* $\sqrt{xy}$, the *arithmetic mean* $(x + y)/2$, and the *harmonic mean* $2/(x^{-1} + y^{-1})$ are the most well-known examples, sometimes called Pythagorean means. There are many other means as well, such as

$$(1) \quad A_\alpha(x, y) := \frac{1}{2}(x^\alpha y^{1-\alpha} + x^{1-\alpha} y^\alpha) \quad (0 \leq \alpha \leq 1)$$

and

$$(2) \quad B_\alpha(x, y) := \left(\frac{x^\alpha + y^\alpha}{2} \right)^{1/\alpha} \quad (-\infty \leq \alpha \leq \infty).$$

In order to have a more treatable class of means, we impose the extra condition of *homogeneity*:

- (vi) $M(tx, ty) = tM(x, y)$ ($t, x, y \in \mathbb{R}^+$).

Note that all the above examples are homogeneous means.

A two-variable function $M(x, y)$ satisfying condition (vi) can be reduced to a one-variable function $f(x) := M(1, x)$. Namely, $M(x, y)$ is recovered from f as

$$(3) \quad M(x, y) = xf\left(\frac{y}{x}\right).$$

What are the properties of f which imply conditions (i)–(v)? They are as follows.

*Received by the editors January 3, 2005; accepted for publication (in revised form) by R. Bhatia June 7, 2005; published electronically December 30, 2005. This work was partially supported by Hungarian grant OTKA T032662.

<http://www.siam.org/journals/simax/27-3/62190.html>

[†]Department for Mathematical Analysis, Budapest University of Technology and Economics, H-1521 Budapest XI., Hungary (petz@math.bme.hu, temesi@math.bme.hu).

- (i)' $f(1) = 1$.
- (ii)' $tf(t^{-1}) = f(t)$.
- (iii)' $f(t) > 1$ if $t > 1$ and $f(t) < 1$ if $0 < t < 1$.
- (iv)' f is monotone increasing.
- (v)' f is continuous.

Then a homogeneous and continuous mean is uniquely described by a function f satisfying the properties (i)'–(v)'.

A function f determines a mean for positive matrices (or operators) when f is matrix monotone. This property is a strengthening of (iv)'. Then the formula

$$(4) \quad M(A, B) = A^{1/2} f(A^{-1/2} B A^{-1/2}) A^{1/2}$$

replaces (3). In this paper the two-variable matrix means of Kubo and Ando [6] are extended to n -tuples $(A_1, A_2, \dots, A_n)$ of positive matrices such that $A_1 \leq A_2 \leq \dots \leq A_n$. The latter restriction is not required for every mean—the Pythagorean means work very well without the ordering. The geometric mean was discussed by Ando, Li, and Mathias [1], and we present an elementary proof of their result. We conjecture that there are other means such that the more-variable-extension works without the restrictive ordering condition. For example, the logarithmic mean seems to be in this class. Although we discuss briefly the means for positive real variables, our emphasis is on the matrix case.

2. Means for three variables. The geometric mean, the arithmetic mean, and the harmonic mean are extended to three (and more) variables as

$$\sqrt{xyz}, \quad (x + y + z)/3, \quad 3/(x^{-1} + y^{-1} + z^{-1}).$$

Our target is to extend a general mean $M_2(x, y)$ of two variables to three variables. In order to do this, we assume that $0 < x \leq y \leq z$ and define a recursion

$$(5) \quad x_1 = x, \quad y_1 = y, \quad z_1 = z,$$

$$(6) \quad x_{n+1} = M_2(x_n, y_n), \quad y_{n+1} = M_2(x_n, z_n), \quad z_{n+1} = M_2(y_n, z_n).$$

Below we refer to this recursion as the *symmetrization procedure*.

One can show by induction that $x_n \leq y_n \leq z_n$; moreover, the sequence (x_n) is increasing and (z_n) is decreasing. Therefore, the limits

$$L := \lim_{n \rightarrow \infty} x_n \quad \text{and} \quad U = \lim_{n \rightarrow \infty} z_n$$

exist. Our next goal is to show that $L = U$.

Assume that $L < U$. By the continuity of the mean, $y_n \rightarrow M := M_2(L, U)$ and $L < M < U$. Taking the limit of the relation $M_2(y_n, z_n) = z_{n+1}$, we obtain

$$M_2(M, U) = U.$$

This contradicts (iii), and $L = U$ is proven.

Given a mean $M_2(x, y)$ of two variables, we define $M_3(x, y, z)$ as the limit $\lim_n x_n = \lim_n y_n = \lim_n z_n$ in recursions (5) and (6) for any $x, y, z \in \mathbb{R}^+$. Note that the existence of the limit does not require the symmetry of the $M_2(x, y)$.

Example 1. For the arithmetic, geometric, and harmonic means the above symmetrization yields the customary three-variable-mean from the corresponding two-variable-means.

Example 2. For the logarithmic mean, different extensions have been proposed in the literature. An extension can be given by the formula [9, 10]

$$L_1(x_1, x_2, \dots, x_n) = (n-1) \frac{L_1(x_2, x_3, \dots, x_n) - L_1(x_1, x_2, \dots, x_{n-1})}{\log x_n - \log x_1}.$$

This gives

$$L_1(x, y, z) := \frac{x}{(\log x - \log y)(\log x - \log z)} + \frac{y}{(\log y - \log x)(\log y - \log z)} + \frac{z}{(\log z - \log x)(\log z - \log y)}.$$

Note that this logarithmic mean appears in the scalar curvature of a certain Riemannian metric on matrices [7].

Another possibility is

$$L_2(x, y, z) := \frac{1}{2} \frac{(x-y)(x-z)(y-z)}{xy(\log x - \log y) + xz(\log z - \log x) + yz(\log y - \log z)},$$

see [12]. Since

$$L_i(x, y, z) = L_i(L(x, y), L(x, z), L(y, z)) \quad (i = 1, 2)$$

does not hold, our above symmetrization does not give $L_i(x, y, z)$ from $L(x, y)$ ($i = 1, 2$).

Given a mean M_2 of two variables, the corresponding mean M_3 of three variables (constructed above) satisfies the equation

$$(7) \quad M_3(x, y, z) = M_3(M_2(x, y), M_2(x, z), M_2(y, z))$$

and has the property $M_3(u, u, u) = u$. Conversely, when M_3 is continuous and has the above two properties, then it must be the mean obtained from M_2 by the symmetrization procedure. Indeed, (7) implies

$$M_3(x, y, z) = M_3(x_n, y_n, z_n).$$

Since the sequences (x_n) , (y_n) , and (z_n) converge to some u , the right-hand side converges to $M_3(u, u, u) = u$. Therefore, M_3 is the mean provided by the symmetrization procedure.

Example 3. Let

$$M_f(x, y) = f^{-1} \left(\frac{f(x) + f(y)}{2} \right)$$

be a *quasi-arithmetic mean* defined by means of a strictly monotone function f . For these means,

$$M_3(x, y, z) = f^{-1} \left(\frac{f(x) + f(y) + f(z)}{3} \right),$$

as is easy to check from (7). Remember that arithmetic, geometric, and harmonic means belong to this class.

3. Means of two positive matrices. A theory of means of positive operators was developed by Kubo and Ando [6]. The theory has many interesting applications. Here we do not go into the details concerning operator means but instead confine ourselves to the essentials. *Operator means* are binary operations on positive operators acting on a Hilbert space, and they satisfy the following conditions, which are similar to the above (i)–(v), where the notation $A < B$ means $B - A$ is positive semidefinite and $A \neq B$:

- (a) $M(A, A) = A$ for every A .
- (b) $M(A, B) = M(B, A)$ for every A and B .
- (c) If $A < B$, then $A < M(A, B) < B$.
- (d) If $A < A'$ and $B < B'$, then $M(A, B) < M(A', B')$.
- (e) M is continuous.

An important further requirement is the *transformer inequality* for all operators C :

- (f) $C M(A, B) C^* \leq M(C A C^*, C B C^*)$.

This is a stronger version of condition (vi), since for an invertible C , equality holds in (f). (Note that another approach to operator means is in [5].)

The key result in the theory is that operator means are in a 1-to-1 correspondence with *operator monotone functions* satisfying conditions (i)'–(v)'. The correspondence is given by

$$(8) \quad M(A, B) = A^{1/2} f(A^{-1/2} B A^{-1/2}) A^{1/2}$$

when A is invertible. Note that an operator monotone function is automatically continuous and even smooth. See [2] for the theory of operator monotone functions. For the sake of simplicity we restrict our discussion to invertible positive matrices. (Generalization to positive operators is immediate.)

An important example is the *geometric mean*,

$$(9) \quad A \# B = A^{1/2} (A^{-1/2} B A^{-1/2})^{1/2} A^{1/2},$$

which has the special property

$$(10) \quad (\lambda A) \# (\mu B) = \sqrt{\lambda \mu} (A \# B)$$

for positive numbers λ and μ .

Example 4. Consider

$$f_\alpha(x) = \frac{2x^{\alpha+1/2}}{1+x^{2\alpha}} \quad \left(0 \leq \alpha \leq \frac{1}{2}\right).$$

Then

$$f_\alpha(x)^{-1} = \frac{1}{2} x^{-\alpha-1/2} + \frac{1}{2} x^{\alpha-1/2}$$

is clearly operator monotone decreasing, so f_α is operator monotone.

The operator monotone function

$$(11) \quad f(x) := \frac{x-1}{\log x}$$

gives the *logarithmic mean*.

Example 5. From the arithmetic and geometric means, one can arrive at the logarithmic mean also in the matrix case; cf. [4]. We define an operation W on the pairs of positive matrices as

$$(12) \quad W(A, B) := ((A + B)/2, ((A + B)/2) \# B).$$

By induction, it can be shown that the coordinates of W , $W_{n,1}$, and $W_{n,2}$ satisfy conditions (a) and (c)–(f) but not (b), so these are not symmetric. Even in this case $W_{n,1}(A, B)$ can be written in the form of (8):

$$W_{n,1}(A, B) = A^{1/2} f_n(A^{-1/2} B A^{-1/2}) A^{1/2}.$$

We have $f_n(x) = W_{n,1}(x, 1)$, which converges to (11) according to [4].

This fact implies that the iteration works also for matrices: the sequence

$$A^{1/2} f_n(A^{-1/2} B A^{-1/2}) A^{1/2}$$

converges to the logarithmic mean of A and B .

The recursive way of mixing two means as in the previous example can produce new means, or new operator monotone functions, from arbitrary initial operator monotone data g_1 and h_1 . Recursively

$$g_n := g_1(g_{n-1}, h_{n-1}) \quad \text{and} \quad h_n := h_1(g_{n-1}, h_{n-1}).$$

Example 6. Mixing of the operator monotone functions

$$g_1(x) = \sqrt{x}, \quad h_1(x) = \frac{2x}{x+1}$$

(or geometric and harmonic means) yields the (joint) limit

$$\frac{x-1}{\log x} \frac{2\sqrt{x}}{1+x}.$$

Carrying out the recursion with

$$g_1(x) = \frac{x-1}{\log x} \quad \text{and} \quad h_1(x) = \frac{x-1}{\log x} \frac{2\sqrt{x}}{1+x},$$

we arrive at the operator monotone function

$$(13) \quad \left(\frac{x-1}{\log x} \right)^2 \frac{2}{1+x}.$$

The operator mean corresponding to the function (13) looks a bit unusual.

In the paper [11] a 1-to-1 correspondence was established between means and certain Riemannian metrics on a space of matrices.

4. Means for three matrices. The mean of three positive matrices can be obtained by a symmetrization procedure from the two-variable-mean.

THEOREM 1. *Let $A, B, C \in M_n(\mathbb{C})$ be positive definite matrices and let M_2 be an operator mean. Assume that $A \leq B \leq C$. Set a recursion as*

$$(14) \quad A_1 = A, \quad B_1 = B, \quad C_1 = C,$$

$$(15) \quad A_{n+1} = M_2(A_n, B_n), \quad B_{n+1} = M_2(A_n, C_n), \quad C_{n+1} = M_2(B_n, C_n).$$

Then the limits

$$(16) \quad M_3(A, B, C) := \lim_n A_n = \lim_n B_n = \lim_n C_n$$

exist.

Proof. By the monotonicity of M_2 and mathematical induction, we see that $A_n \leq B_n \leq C_n$. It follows that the sequence (A_n) is increasing and (C_n) is decreasing. Therefore, the limits

$$L := \lim_{n \rightarrow \infty} A_n \quad \text{and} \quad U = \lim_{n \rightarrow \infty} C_n$$

exist. We claim that $L = U$.

Assume that $L \neq U$. By continuity, $B_n \rightarrow M_2(L, U) =: M$, where $L < M < U$. Since

$$M_2(B_n, C_n) = C_{n+1},$$

the limit $n \rightarrow \infty$ gives $M_2(M, U) = U$, which contradicts $M < U$. $\square$

By the symmetrization procedure any operator mean of two variables has an extension to those triplets (A, B, C) which can be ordered. In a few cases, the latter restriction can be skipped.

Example 7. If $M_2(A, B) = (A + B)/2$ is the arithmetic mean, then the above recursion can be written explicitly and the convergence is easily verified without the conditions $A \leq B \leq C$. The limit gives

$$M_3(A, B, C) = \frac{1}{3}(A + B + C).$$

Example 8. The case of the harmonic mean $M_2(A, B) = 2/(A^{-1} + B^{-1})$ can be reduced to the arithmetic mean by means of the inverse, and the recursion gives

$$M_3(A, B, C) = \frac{3}{A^{-1} + B^{-1} + C^{-1}}.$$

The following theorem shows this for the geometric mean.

THEOREM 2. Let $A, B, C \in M_n(\mathbb{C})$ be positive definite matrices. Set a recursion as

$$(17) \quad A_1 = A, \quad B_1 = B, \quad C_1 = C,$$

$$(18) \quad A_{n+1} = A_n \# B_n, \quad B_{n+1} = A_n \# C_n, \quad C_{n+1} = B_n \# C_n.$$

Then the limits

$$(19) \quad G(A, B, C) := \lim_n A_n = \lim_n B_n = \lim_n C_n$$

exist.

Proof. Choose positive numbers λ and μ such that

$$A' := A < B' := \lambda B < C' := \mu C.$$

Start the recursion with these matrices. By Theorem 1 the limits

$$G(A', B', C') := \lim_n A'_n = \lim_n B'_n = \lim_n C'_n$$

exist. For the numbers

$$a := 1, \quad b := \lambda, \quad \text{and} \quad c := \mu$$

the recursion provides a convergent sequence (a_n, b_n, c_n) of triplets,

$$(\lambda\mu)^{1/3} = \lim_n a_n = \lim_n b_n = \lim_n c_n.$$

Since

$$A_n = A'_n/a_n, B_n = B'_n/b_n \quad \text{and} \quad C_n = C'_n/c_n$$

due to property (10) of the geometric mean, the limits stated in the theorem must exist and equal $G(A', B', C')/(\lambda\mu)^{1/3}$. $\square$

The result of Theorem 2 was obtained in [1], but our proof is different and completely elementary. The path

$$(20) \quad \gamma(t) = A^{1/2} (A^{-1/2} B A^{-1/2})^t A^{1/2} \quad (0 \leq t \leq 1)$$

connects A with B . In [3, 8], a Riemannian geometry is defined on the set of positive definite matrices such that the path (20) is geodesic. In this approach the geometric mean of two (or three) matrices has a nice geometric interpretation (similar to the arithmetic mean in the flat space).

THEOREM 3. *The mean $M_3(A, B, C)$ defined in Theorem 1 under the condition $A \leq B \leq C$ is a symmetric continuous function of all variables, is monotone in all variables, and satisfies the transformer inequality*

$$X M_3(A, B, C) X^* \leq M_3(X A X^*, X B X^*, X C X^*)$$

for any matrix X .

Proof. Symmetry and monotonicity are obvious; we prove the transformer inequality. Let A_n, B_n , and C_n be the matrices recursively defined in the construction of $M_3(A, B, C)$, and let A'_n, B'_n, C'_n be those defined in the construction of $M_3(X A X^*, X B X^*, X C X^*)$. By repeated use of the transformer inequality (f), we see that

$$X A_n X^* \leq A'_n,$$

and the limit $n \rightarrow \infty$ gives our statement.

To prove the continuity it is essential to check that

$$\lim_{\varepsilon \rightarrow +0} M_3(A - \varepsilon I, B - \varepsilon I, C - \varepsilon I) = M_3(A, B, C) = \lim_{\varepsilon \rightarrow +0} M_3(A + \varepsilon I, B + \varepsilon I, C + \varepsilon I).$$

Both limits exist by monotonicity, and $M_3(A, B, C)$ must be in between. The different limits would lead to a contradiction.

If $\|A - A'\|, \|B - B'\|, \|C - C'\| \leq \varepsilon$, then $A - \varepsilon I \leq A' \leq A + \varepsilon I$ and similarly $B - \varepsilon I \leq B' \leq B + \varepsilon I, C - \varepsilon I \leq C' \leq C + \varepsilon I$. By monotonicity

$$M_3(A - \varepsilon I, B - \varepsilon I, C - \varepsilon I) \leq M_3(A', B', C') \leq M_3(A + \varepsilon I, B + \varepsilon I, C + \varepsilon I).$$

Taking limit $\varepsilon \rightarrow 0$, continuity follows. $\square$

In the case of the geometric mean, the transformer inequality holds without the restriction on ordering. This mean has several other properties, as shown in [1].

The next theorem is straightforward from the construction of three-variable-means.

THEOREM 4. *An inequality between the two-variable-means M_2 and M'_2 is inherited by the corresponding three-variable-means M_3 and M'_3 .*

In particular, this theorem applies to the inequality between the arithmetic, logarithmic, and geometric means.

5. Discussion. Given a positive operator mean M_2 , formula (7) defines a corresponding three-variable-mean, which certainly makes sense for ordered triplets.

The approach extends to more variables in an obvious way. In this case we start with an ordered n -tuple, form the $(n-1)$ -variable-mean of all possible subsets, obtain an ordered n -tuple, and continue the procedure. The limit gives the n -variable-mean (provided that the $(n-1)$ -variable-mean is already at our disposal). The transformer inequality can be proved as in Theorem 3.

Restricting the discussion to three variables, we observed that ordering the triplet of the matrices is not necessary for the construction of some means. Computer simulations (programmed in MAPLE) show that recursion for the logarithmic mean seems to be convergent without the ordering condition on the three input matrices (see Figure 1). We do not have a proof for this.

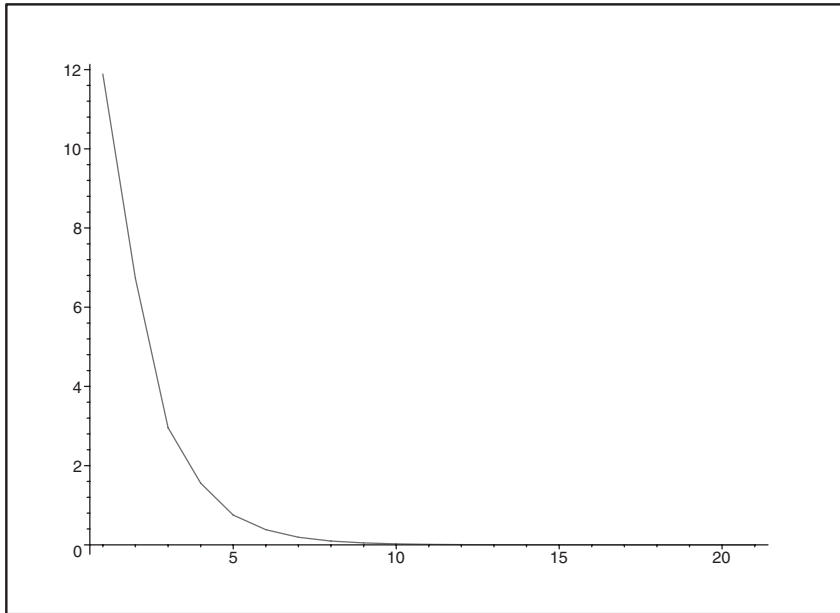


FIG. 1. Iteration for the logarithmic mean. The 3×3 matrices A_1 , B_1 , and C_1 are chosen randomly and the symmetrization procedure is carried out in 20 steps. The maximum of the diameter $D(A_n, B_n, C_n)$ of 100 tests is shown vertically, and the number of iterations, from 1 to 20, is on the horizontal axis.

It seems that there are more three-variable-means without the ordering constraint than the known arithmetic, geometric, and harmonic means. (Computer simulation has been carried out for some other means as well.)

An operator monotone function f associated with a matrix mean has the property

$f(1) = 1$ and $f'(1) = 1/2$. The latter formula follows from (ii)'. From the power series expansion around 1, one can deduce that there are some positive numbers ε and δ such that

$$\|A^{1/2}f(A^{-1/2}B_1A^{-1/2})A^{1/2} - A^{1/2}f(A^{-1/2}B_2A^{-1/2})A^{1/2}\| \leq (1-\delta)\|B_1 - B_2\|$$

whenever $I - \varepsilon \leq A, B_1, B_2 \leq I + \varepsilon$. This estimate equivalently means

$$(21) \quad \|M_f(A, B_1) - M_f(A, B_2)\| \leq (1-\delta)\|B_1 - B_2\|,$$

where $\|\cdot\|$ denotes the operator norm.

If the *diameter* of a triplet (A, B, C) is defined as

$$D(A, B, C) := \max\{\|A - B\|, \|B - C\|, \|A - C\|\},$$

then we have

$$D(A_{n+1}, B_{n+1}, C_{n+1}) \leq (1-\delta)D(A_n, B_n, C_n),$$

provided that the condition $I - \varepsilon \leq A_1, B_1, C_1 \leq I + \varepsilon$ holds. Therefore in a small neighborhood of the identity the symmetrization procedure converges exponentially fast for any triplet of matrices and for any matrix mean.

Acknowledgments. The authors thank T. Ando, M. Pálfi, J. Réffy, and R. Szentpéteri for useful conversations on the subject and for some remarks, and thank the referee for information on [3, 8].

REFERENCES

- [1] T. ANDO, C.-K. LI, AND R. MATHIAS, *Geometric means*, Linear Algebra Appl., 385 (2004), pp. 305–334.
- [2] R. BHATIA, *Matrix Analysis*, Springer-Verlag, New York, 1996.
- [3] R. BHATIA AND J. HOLBROOK, *Geometry and means*, Linear Algebra Appl., to appear.
- [4] B. C. CARLSON, *The logarithmic mean*, Amer. Math. Monthly, 79 (1972), pp. 615–618.
- [5] F. HIAI AND H. KOSAKI, *Means of Hilbert Space Operators*, Lecture Notes in Math. 1820, Springer-Verlag, Berlin, 2003.
- [6] F. KUBO AND T. ANDO, *Means of positive linear operators*, Math. Ann., 246 (1980), pp. 205–224.
- [7] P. W. MICHOR, D. PETZ, AND A. ANDAI, *On the curvature of a certain Riemannian space of matrices*, Infin. Dimens. Anal. Quantum Probab. Relat. Top., 3 (2000), pp. 199–212.
- [8] M. MOAKHER, *A differential geometric approach to the geometric mean of symmetric positive-definite matrices*, SIAM J. Matrix Anal. Appl., 26 (2005), pp. 735–747.
- [9] S. MUSTONEN, *Logarithmic Mean for Several Arguments*, 2002. Available online at <http://www.survo.fi/papers/logmean.pdf>
- [10] E. NEUMAN, *The weighted logarithmic mean*, J. Math. Anal. Appl., 188 (1994), pp. 885–900.
- [11] D. PETZ, *Monotone metrics on matrix spaces*, Linear Algebra Appl., 244 (1996), pp. 81–96.
- [12] A. O. PITTENGER, *The logarithmic mean in n variables*, Amer. Math. Monthly, 92 (1985), pp. 99–104.
- [13] M. K. VAMANAMURTHY AND M. VUORINEN, *Inequalities for means*, J. Math. Anal. Appl., 183 (1994), pp. 155–166.

COMPLEX POLYTOPE EXTREMALITY RESULTS FOR FAMILIES OF MATRICES*

N. GUGLIELMI[†], F. WIRTH[‡], AND M. ZENNARO[§]

Abstract. In this paper we consider finite families of complex $n \times n$ -matrices. In particular, we focus on those families that satisfy the so-called *finiteness conjecture*, which was recently disproved in its more general formulation. We conjecture that the validity of the finiteness conjecture for a finite family of *nondefective* type is equivalent to the existence of an *extremal norm* in the class of *complex polytope norms*. However, we have not been able to prove this *complex polytope extremality conjecture*, but we are able to prove the *small complex polytope extremality theorem* under some more restrictive hypotheses on the underlying family of matrices. In addition, our theorem assures a certain finiteness property on the number of vertices of the unit ball of the extremal complex polytope norm, which could be very useful for the construction of suitable algorithms aimed at the actual computation of the *spectral radius* of the family.

Key words. families of matrices, joint spectral radius, extremal norms, complex polytope norms, finiteness property

AMS subject classifications. 14A99, 34D08, 93D09

DOI. 10.1137/040606818

1. Introduction. To a finite (or more generally bounded) set of real or complex matrices a growth rate may be associated by considering the maximal growth of all products that can be formed by taking matrices in the set. This quantity is often called the joint or generalized spectral radius of this set. Interestingly, it provides relevant information in many applications where the problem can be described by several matrices. For instance it characterizes the regularity of certain wavelets [DauLag1, Mae1, Mae2], capacity of codes [MoisOrli01], and the stability of hybrid systems [Bara88, Wirt98] and of the numerical solution of ordinary differential equations [GugZen2]. Further references to applications can be found in [DauLag2, Wirt02].

The precise definition of the joint spectral radius is as follows. We consider a bounded family $\mathcal{F} = \{A^{(i)}\}_{i \in \mathcal{I}}$ of complex $n \times n$ -matrices, where $\mathcal{I}$ is a set of indices, possibly infinite.

Let $\|\cdot\|$ be a given norm on the vector space $\mathbb{C}^n$ and let the same symbol $\|\cdot\|$ denote also the corresponding induced $n \times n$ -matrix norm. Then, for each $k = 0, 1, \dots$, consider the set $\Sigma_k(\mathcal{F})$ of all possible products of length k whose factors are elements of $\mathcal{F}$, that is,

$$\Sigma_k(\mathcal{F}) = \{A^{(i_1)} \cdot \dots \cdot A^{(i_k)} \mid i_1, \dots, i_k \in \mathcal{I}\},$$

with the convention that $\Sigma_0(\mathcal{F}) = \{I\}$, I the identity matrix. Moreover, for each

*Received by the editors April 15, 2004; accepted for publication (in revised form) by P. Van Dooren June 9, 2005; published electronically December 30, 2005. This work was supported by M.I.U.R., by INdAM - G.N.C.S., and by Science Foundation Ireland grant 00/PI.1/C067.

<http://www.siam.org/journals/simax/27-3/60681.html>

[†]Dipartimento di Matematica Pura e Applicata, Università de L'Aquila, I-67100 L'Aquila, Italy (guglielm@univaq.it).

[‡]Hamilton Institute, NUI Maynooth, Maynooth, Co. Kildare, Ireland (Fabian.Wirth@nuim.ie).

[§]Dipartimento di Matematica e Informatica, Università di Trieste, I-34100 Trieste, Italy (zennaro@univ.trieste.it).

$k = 0, 1, \dots$, consider the number

$$(1.1) \quad \hat{\rho}_k(\mathcal{F}) = \sup_{P \in \Sigma_k(\mathcal{F})} \|P\|$$

and, finally, define the *joint spectral radius* of $\mathcal{F}$ as

$$(1.2) \quad \hat{\rho}(\mathcal{F}) = \limsup_{k \rightarrow \infty} \hat{\rho}_k(\mathcal{F})^{1/k}$$

(see Rota and Strang [RotStr]). Note that the numbers $\hat{\rho}_k(\mathcal{F})$ depend on the particular norm $\|\cdot\|$ used in (1.1), whereas, by the equivalence of all the norms in finite-dimensional spaces, it turns out that $\hat{\rho}(\mathcal{F})$ is independent of it.

Analogously, let $\rho(\cdot)$ denote the spectral radius of an $n \times n$ -matrix and then, for each $k = 0, 1, \dots$, consider the number

$$\bar{\rho}_k(\mathcal{F}) = \sup_{P \in \Sigma_k(\mathcal{F})} \rho(P)$$

and define the *generalized spectral radius* of $\mathcal{F}$ as

$$\bar{\rho}(\mathcal{F}) = \limsup_{k \rightarrow \infty} \bar{\rho}_k(\mathcal{F})^{1/k}$$

(see Daubechies and Lagarias [DauLag1]). It is not difficult to see [DauLag1] that

$$(1.3) \quad \bar{\rho}_k(\mathcal{F}) \leq \bar{\rho}(\mathcal{F})^k \leq \hat{\rho}(\mathcal{F})^k \leq \hat{\rho}_k(\mathcal{F}) \quad \text{for all } k \geq 0,$$

and it was later shown that

$$\hat{\rho}(\mathcal{F}) = \bar{\rho}(\mathcal{F})$$

(see Berger and Wang [BerWan], Elsner [Els], Shih, Wu, and Pang [ShWuPa], and Shih [Shi]). In light of the above equality, the joint and the generalized spectral radii of $\mathcal{F}$ are the same number, which we shall simply call the *spectral radius* of the (bounded) family of matrices $\mathcal{F}$ and denote by $\rho(\mathcal{F})$.

The above definitions and results are nice generalizations of the well-known situation for *single* families. In particular, the equality $\hat{\rho}(\mathcal{F}) = \bar{\rho}(\mathcal{F})$ is the generalization of the so-called Gelfand limit.

In practical applications, the actual computation of $\rho(\mathcal{F})$ is very important, but, unfortunately, if the family $\mathcal{F}$ is not just a single matrix, this is not an easy task at all; see, e.g., [Kozy90, TsitBlon97]. Nonetheless there are several approaches to the computation, the most recent being that of [BlonNest05]; see also the references therein.

In this paper we try to give a contribution in this direction for special classes of families. More precisely, we find conditions on the family which are sufficient to guarantee the existence of an *extremal norm* in the class of *complex polytope norms*. In addition, under such sufficient conditions, we prove that the unit ball of the extremal norm is a *balanced complex polytope* with a finite *essential system of vertices*. Such a finiteness property is very useful in view of the construction of suitable algorithms aimed at the actual computation and approximation of $\rho(\mathcal{F})$ via the detection of an extremal norm. In section 2 we review some of the most important definitions and results available in the literature, which are useful for our subsequent developments. In section 3 we illustrate a few simple results concerning the extremal norms.

In section 4 we recall the definitions of a *balanced complex polytope*, an *adjoint balanced complex polytope*, and the corresponding *complex polytope norms* and *adjoint complex polytope norms*, reviewing a few of their most important properties. The main results of the paper are presented in section 5, where we state the *complex polytope extremality conjecture*, which is our guess that, for a finite *nondefective* family $\mathcal{F}$, the validity of the finiteness conjecture (recently disproved in its more general formulation) is equivalent to the existence of an *extremal norm* in the class of *complex polytope norms*. Indeed, we are able to prove only the *small complex polytope extremality theorem* under some more restrictive hypotheses on the underlying family of matrices.

Finally, in section 6 we present some examples that prove the necessity of the particular hypotheses of the *small complex polytope extremality theorem* in order to guarantee some specific finiteness properties for the unit ball of the extremal norm.

While our results are mostly theoretical in nature, they have potential impact in applications. This potential lies in the fact that, if there is prior knowledge that a certain set $\mathcal{F}$ has an extremal complex polytope norm, then one could devise algorithms for the computation of $\rho(\mathcal{F})$ that rely on the computation of the extremal points of the unit ball of the norm. Furthermore, part of the results of this paper have already been successfully applied by Guglielmi and Zennaro [GugZen2] in the analysis of the *zero stability* of some linear multistep methods for the numerical solution of ordinary differential equations. At that time, however, the small complex polytope extremality theorem was still at the stage of a conjecture, and [GugZen2] makes reference to an early, unpublished version of the present paper under another title, namely, “Polytope norms for families of matrices.”

2. Preliminary results from the literature. In this section we briefly review some results from the literature.

In what follows, for the bounded family $\mathcal{F} = \{A^{(i)}\}_{i \in \mathcal{I}}$ of complex $n \times n$ -matrices, if $\|\cdot\|$ denotes a norm on the vector space $\mathbb{C}^n$ and the corresponding induced $n \times n$ -matrix norm, we shall still use the same notation to define

$$\|\mathcal{F}\| = \hat{\rho}_1(\mathcal{F}) = \sup_{i \in \mathcal{I}} \|A^{(i)}\|.$$

The following result can be found, for example, in [RotStr] and Elsner [Els].

PROPOSITION 2.1. *The spectral radius of a bounded family $\mathcal{F}$ of complex $n \times n$ -matrices is characterized by the equality*

$$(2.1) \quad \rho(\mathcal{F}) = \inf_{\|\cdot\| \in \mathcal{N}} \|\mathcal{F}\|,$$

where $\mathcal{N}$ denotes the set of all possible induced $n \times n$ -matrix norms.

Given a family $\mathcal{F}$, an important question to answer is whether or not the inf in (2.1) is actually attained by some induced matrix norm. To this purpose, we give the following definition.

DEFINITION 2.2. *We shall say that a norm $\|\cdot\|_*$ satisfying the condition*

$$(2.2) \quad \|\mathcal{F}\|_* = \rho(\mathcal{F})$$

is extremal for the family $\mathcal{F}$.

It is well known that, for a single family $\{A\}$, the existence of an extremal norm is equivalent to the fact that the matrix A is *nondefective*; i.e., all of the blocks relevant to the eigenvalues of maximum modulus are diagonal in its Jordan canonical form. Whenever $\rho(A) > 0$, we have another equivalent property, that is, with $\hat{A} = \rho(A)^{-1}A$,

the power set $\Sigma(\hat{A}) = \{\hat{A}^k \mid k \geq 1\}$ is bounded. These results generalize to a bounded family $\mathcal{F}$ as follows. Given a bounded family $\mathcal{F} = \{A^{(i)}\}_{i \in \mathcal{I}}$ of complex $n \times n$ -matrices with $\rho(\mathcal{F}) > 0$, let us consider the *normalized* family

$$\hat{\mathcal{F}} = \{\rho(\mathcal{F})^{-1} A^{(i)}\}_{i \in \mathcal{I}},$$

whose spectral radius is $\rho(\hat{\mathcal{F}}) = 1$. Then consider the semigroup of matrices generated by $\hat{\mathcal{F}}$, i.e.,

$$\Sigma(\hat{\mathcal{F}}) = \bigcup_{k \geq 1} \Sigma_k(\hat{\mathcal{F}}).$$

DEFINITION 2.3. *A bounded family $\mathcal{F}$ of complex $n \times n$ -matrices is said to be defective if the corresponding normalized family $\hat{\mathcal{F}}$ is such that the semigroup $\Sigma(\hat{\mathcal{F}})$ is an unbounded set of matrices. Otherwise, if $\Sigma(\hat{\mathcal{F}})$ is bounded, then the family $\mathcal{F}$ is said to be nondefective.*

Note that we gave the definition of defective family without involving directly the spectral properties of its elements. The following result can be found, for example, in [Kozy90, BerWan].

PROPOSITION 2.4. *A bounded family $\mathcal{F}$ of complex $n \times n$ -matrices admits an extremal norm $\|\cdot\|_*$ if and only if it is nondefective. Moreover, if $\mathcal{F}$ is nondefective, any given norm $\|\cdot\|$ on the space of vectors $x \in \mathbb{C}^n$ determines the extremal norm*

$$(2.3) \quad \|x\|_* = \sup_{k \geq 0} \sup_{P \in \Sigma_k(\mathcal{F})} \frac{\|Px\|}{\rho(\mathcal{F})^k}.$$

From Proposition 2.4 it turns out that, for a nondefective family, each vector norm $\|\cdot\|$ canonically determines an extremal norm. However, it is worth remarking that, although (2.3) gives a constructive way of finding an extremal norm, its importance is mainly theoretical since it is often useless from a practical point of view.

In order to state another important result about defective and nondefective families (see Barabanov [Bara88] or Elsner [Els]), we give the following definition according to [RR00].

DEFINITION 2.5. *A bounded family $\mathcal{F} = \{A^{(i)}\}_{i \in \mathcal{I}}$ of complex $n \times n$ -matrices is said to be reducible if there exist a nonsingular $n \times n$ -matrix M and two integers $n_1, n_2 \geq 1$, $n_1 + n_2 = n$, such that, for all $i \in \mathcal{I}$, it holds that*

$$M^{-1} A^{(i)} M = \begin{bmatrix} A_{11}^{(i)} & A_{12}^{(i)} \\ O & A_{22}^{(i)} \end{bmatrix},$$

where the blocks $A_{11}^{(i)}$, $A_{12}^{(i)}$, $A_{22}^{(i)}$ are $n_1 \times n_1$ -, $n_1 \times n_2$ -, and $n_2 \times n_2$ -matrices, respectively. If a family $\mathcal{F}$ is not reducible, then it is said to be irreducible.

THEOREM 2.6. *If a bounded family $\mathcal{F}$ of complex $n \times n$ -matrices is defective, then it is reducible.*

Note that irreducibility is a generic property. Furthermore, in a suitable neighborhood of an irreducible family $\mathcal{F}_0$ the spectral radius is a Lipschitz continuous function of $\mathcal{F}$ (see Wirth [Wirt02]). We remark that, whereas a defective family is always reducible, the opposite implication is not necessarily true. For example, for $n \geq 2$ all single families $\{A\}$ are clearly reducible but not necessarily defective. See again [Wirt02] for conditions that ensure nondefectiveness for reducible sets $\mathcal{F}$. The following corollary to Theorem 2.6 is obvious.

COROLLARY 2.7. *If a bounded family $\mathcal{F}$ of complex $n \times n$ -matrices is irreducible, then it is nondefective.*

We conclude this section by recalling an important conjecture arising from work of Daubechies and Lagarias [DauLag1] and stated by Lagarias and Wang [LagWan], whose validity would be of much help for the actual computation of the spectral radius $\rho(\mathcal{F})$ of finite families.

DEFINITION 2.8 (finiteness property). *A finite family of complex $n \times n$ -matrices $\mathcal{F}$ is said to have the finiteness property if there exist $k^* \geq 1$ and a product $\bar{P} \in \Sigma_{k^*}(\mathcal{F})$ such that*

$$(2.4) \quad \rho(\mathcal{F}) = \bar{\rho}_{k^*}(\mathcal{F})^{1/k^*} = \rho(\bar{P})^{1/k^*}.$$

DEFINITION 2.9. *If $\mathcal{F}$ is a bounded family of complex $n \times n$ -matrices, any matrix $\bar{P} \in \Sigma_{k^*}(\mathcal{F})$ satisfying (2.4) for some $k^* \geq 1$ will be called a spectrum-maximizing product (s.m.p.) for $\mathcal{F}$.*

Observe that the finiteness property yields the existence of at least one s.m.p. $\bar{P}$ for finite families. Lagarias and Wang were able to give sufficient conditions in terms of extremal norms guaranteeing that the finiteness property holds. The general *finiteness conjecture* stating that all finite sets of matrices have the finiteness property was instead recently disproved by Bousch and Mairesse [BouMer] and, later, by Blondel, Theys, and Vladimirov [Blondel1].

Now we recall the following definition from [GugZen1].

DEFINITION 2.10. *Assume that $\mathcal{F}$ is a normalized bounded family of complex $n \times n$ -matrices (i.e., $\rho(\mathcal{F}) = 1$) and that there exists a sequence of products $P_k \in \Sigma_{d_k}(\mathcal{F})$, d_k nondecreasing integers, such that*

$$(2.5) \quad \lim_{k \rightarrow \infty} P_k = \tilde{P},$$

where $\tilde{P} \in \overline{\Sigma(\mathcal{F})}$ and $\rho(\tilde{P}) = 1$. Then $\tilde{P}$ will be called a limit spectrum-maximizing product (l.s.m.p.) for $\mathcal{F}$.

Note that, for a normalized family $\mathcal{F}$, an s.m.p. $\bar{P}$ is an l.s.m.p., too. To see this, just put $P_k = \bar{P}$ for all $k \geq 1$. Moreover, if the family $\mathcal{F}$ is nondefective, another possibility is to consider the power sequence $\{\bar{P}^k\}_{k \geq 1}$ and, since $\Sigma(\mathcal{F})$ is bounded, to extract a subsequence $\{\bar{P}^{k_s}\}_{s \geq 1}$ converging to some $\tilde{P} \in \overline{\Sigma(\mathcal{F})}$, which obviously satisfies $\rho(\tilde{P}) = 1$. For the sake of brevity, we shall say that such a limit point of the sequence $\{\bar{P}^k\}_{k \geq 1}$ is an *infinite power* of the matrix $\bar{P}$. For nondefective families, Guglielmi and Zennaro [GugZen3] proved the following result.

THEOREM 2.11. *Let $\mathcal{F}$ be a (possibly infinite) nondefective bounded family of complex $n \times n$ -matrices. Then there exists an l.s.m.p. $\tilde{P}$ for the normalized family $\hat{\mathcal{F}}$.*

On the contrary, for defective families Guglielmi and Zennaro give some counterexamples to the existence of l.s.m.p.'s whenever the dimension of the matrices is $n \geq 4$.

3. Some properties of extremal norms. In this section we consider nondefective bounded families $\mathcal{F}$ of complex $n \times n$ -matrices and find some straightforward properties of the extremal norms. Here $\|\cdot\|_*$ denotes an extremal norm for $\mathcal{F}$. Note first that as an easy consequence of Definition 2.2 we have

$$(3.1) \quad \|P\|_* \leq \rho(\mathcal{F})^k \quad \text{for all } P \in \Sigma_k(\mathcal{F}).$$

Using (3.1) and the submultiplicative property of the induced matrix norms, we immediately see that if $P \in \Sigma_k(\mathcal{F})$ and $Q \in \Sigma_h(\mathcal{F})$ are such that $\|PQ\|_* = \rho(\mathcal{F})^{k+h}$, then they satisfy the equalities

$$(3.2) \quad \|P\|_* = \rho(\mathcal{F})^k \quad \text{and} \quad \|Q\|_* = \rho(\mathcal{F})^h.$$

The next statement is obtained just by iterating (3.2).

PROPOSITION 3.1. *Let $\mathcal{F} = \{A^{(i)}\}_{i \in \mathcal{I}}$ be a nondefective bounded family of complex $n \times n$ -matrices and let $\|\cdot\|_*$ be an extremal norm for $\mathcal{F}$. If $P = A^{(i_k)} \cdots A^{(i_1)} \in \Sigma_k(\mathcal{F})$ satisfies $\|P\|_* = \rho(\mathcal{F})^k$, then it holds that $\|A^{(i_r)}\|_* = \rho(\mathcal{F})$ for all factors $A^{(i_r)}$, $r = 1, \dots, k$, of P .*

LEMMA 3.2. *Let $\mathcal{F}$ be a nondefective bounded family of complex $n \times n$ -matrices and let $\|\cdot\|_*$ be an extremal norm for $\mathcal{F}$. If $\bar{P} \in \Sigma_{k^*}(\mathcal{F})$ is an s.m.p. for $\mathcal{F}$, then*

$$(3.3) \quad \|\bar{P}\|_* = \rho(\mathcal{F})^{k^*}.$$

Proof. This is an easy consequence of (2.4) and (3.1). $\square$

DEFINITION 3.3. *Let A be a complex $n \times n$ -matrix and let $\|\cdot\|$ be a norm on $\mathbb{C}^n$. Then any vector $x \in \mathbb{C}^n$, $x \neq 0$, such that $\|Ax\| = \|A\| \cdot \|x\|$ will be said to be maximizing for A with respect to the norm $\|\cdot\|$.*

Again, (3.1) and the submultiplicative property of the induced matrix norms yield immediately the following result.

PROPOSITION 3.4. *Let $\mathcal{F}$ be a nondefective bounded family of complex $n \times n$ -matrices and let $\|\cdot\|_*$ be an extremal norm for $\mathcal{F}$. Moreover, let $P \in \Sigma_k(\mathcal{F})$ and $Q \in \Sigma_h(\mathcal{F})$ be such that $\|PQ\|_* = \rho(\mathcal{F})^{k+h}$. If $x \in \mathbb{C}^n$ is maximizing for PQ with respect to the norm $\|\cdot\|_*$, then x is maximizing also for Q , and Qx is maximizing for P .*

We conclude this section with a theorem which will be useful in section 5.

THEOREM 3.5. *Assume that a nondefective bounded family $\mathcal{F}$ of complex $n \times n$ -matrices has an s.m.p. $\bar{P} = A^{(i_{k^*})} \cdots A^{(i_1)}$ and let $\|\cdot\|_*$ be an extremal norm. Then, if $x \neq 0$ is an eigenvector of $\bar{P}$ corresponding to an eigenvalue λ with $|\lambda| = \rho(\bar{P}) = \rho(\mathcal{F})^{k^*}$, then it holds that $\|P^{(r)}x\|_* = \rho(\mathcal{F})^r \|x\|_*$ for all right factors $P^{(r)} = A^{(i_r)} \cdots A^{(i_1)}$ of the s.m.p. $\bar{P}$, $r = 1, \dots, k^*$.*

Proof. Observe that, by (3.3), the eigenvector x satisfies the equalities $\|\bar{P}x\|_* = \|\lambda x\|_* = \rho(\mathcal{F})^{k^*} \|x\|_* = \|\bar{P}\|_* \|x\|_*$; i.e., x is maximizing for $\bar{P}$ with respect to the norm $\|\cdot\|_*$. Then apply Proposition 3.4 iteratively, taking (3.2) into account. $\square$

4. Complex polytopes and related norms. In this section we define *complex polytopes* as generalizations of real polytopes (see, e.g., Ziegler [Zie]) to the complex case and, consequently, we extend the concept of *polytope norm* to the complex case in a straightforward way. Most of the results (here given without proof) are just either more particular or more general instances of other results which can be found in the literature. In any case, a detailed and self-contained presentation of this topic including all the proofs may be found in Guglielmi and Zennaro [GugZen4].

DEFINITION 4.1. *We shall say that a set $\mathcal{X} \subset \mathbb{C}^n$ is absolutely convex if, for all $x', x'' \in \mathcal{X}$, and $\lambda', \lambda'' \in \mathbb{C}$ such that $|\lambda'| + |\lambda''| \leq 1$, it holds that $\lambda'x' + \lambda''x'' \in \mathcal{X}$.*

DEFINITION 4.2. *Let $\mathcal{X} \subset \mathbb{C}^n$. Then the intersection of all absolutely convex sets containing $\mathcal{X}$ will be called the absolutely convex hull of $\mathcal{X}$ and will be denoted by $\text{absco}(\mathcal{X})$.*

It is well known that $\text{absco}(\mathcal{X})$ is the set of all the finite absolutely convex linear combinations of vectors of $\mathcal{X}$, i.e., $x \in \text{absco}(\mathcal{X})$ if and only if there exist

$x^{(1)}, \dots, x^{(k)} \in \mathcal{X}$ with $k \geq 1$ such that

$$x = \sum_{i=1}^k \lambda_i x^{(i)} \quad \text{with} \quad \lambda_i \in \mathbb{C} \quad \text{and} \quad \sum_{i=1}^k |\lambda_i| \leq 1.$$

In particular, if $\mathcal{X} = \{x^{(i)}\}_{1 \leq i \leq m}$ is a finite set of vectors, then

$$(4.1) \quad \text{absco}(\mathcal{X}) = \left\{ x \in \mathbb{C}^n \mid x = \sum_{i=1}^m \lambda_i x^{(i)} \text{ with } \lambda_i \in \mathbb{C} \text{ and } \sum_{i=1}^m |\lambda_i| \leq 1 \right\}$$

and, in this case, it is a closed subset of $\mathbb{C}^n$. In what follows, if $\mathcal{X}'$ and $\mathcal{X}''$ are two subsets of $\mathbb{C}^n$, we shall write $\mathcal{X}' \subset \mathcal{X}''$ ($\mathcal{X}' \supset \mathcal{X}''$) to denote proper inclusions, i.e., $\mathcal{X}' \subseteq \mathcal{X}''$ ($\mathcal{X}' \supseteq \mathcal{X}''$) and $\mathcal{X}' \neq \mathcal{X}''$. The forthcoming definition extends the usual definition of a symmetric polytope in the real space $\mathbb{R}^n$.

DEFINITION 4.3. *We shall say that a bounded set $\mathcal{P} \subset \mathbb{C}^n$ is a balanced complex polytope (b.c.p.) if there exists a finite set $\mathcal{X} = \{x^{(i)}\}_{1 \leq i \leq m}$ of vectors such that $\text{span}(\mathcal{X}) = \mathbb{C}^n$ and*

$$(4.2) \quad \mathcal{P} = \text{absco}(\mathcal{X}).$$

Moreover, if $\text{absco}(\mathcal{X}') \subset \text{absco}(\mathcal{X})$ for all $\mathcal{X}' \subset \mathcal{X}$, then $\mathcal{X}$ will be called an essential system of vertices for $\mathcal{P}$, whereas any vector $ux^{(i)}$ with $u \in \mathbb{C}$, $|u| = 1$, will be called a vertex of $\mathcal{P}$.

Note that, geometrically speaking, a b.c.p. $\mathcal{P}$ is not a classical polytope. In fact, if we identify the complex space $\mathbb{C}^n$ with the real space $\mathbb{R}^{2n}$, we can easily see that $\mathcal{P}$ is not bounded by hyperplanes. In general, even the intersection $\mathcal{P} \cap \mathbb{R}^n$ is not a polytope in $\mathbb{R}^n$.

Example 4.1. Consider the b.c.p.

$$\mathcal{P} = \text{absco}\{[1, i]^T, [1, -i]^T\} \subseteq \mathbb{C}^2.$$

It can be easily verified that $\mathcal{P} \cap \mathbb{R}^2$ is the unit circle

$$(4.3) \quad \mathcal{C} = \left\{ [x_1, x_2]^T \in \mathbb{R}^2 \mid x_1^2 + x_2^2 = 1 \right\},$$

that is, not a polytope in $\mathbb{R}^2$. $\diamond$

On the other hand, it is easy to see that, if the b.c.p. $\mathcal{P}$ admits an essential system of real vertices, then $\mathcal{P} \cap \mathbb{R}^n$ is a polytope in $\mathbb{R}^n$.

In order to recall the concept of *adjoint* set (that, in the literature, is often referred to as *polarity* or *duality* (see again [Zie] or Heuser [Heu]), we consider the usual Euclidean scalar product in $\mathbb{C}^n$ defined by $\langle x, y \rangle = \sum_{j=1}^n x_j \bar{y}_j$.

DEFINITION 4.4. *Let $\mathcal{X} \subset \mathbb{C}^n$. Then the set*

$$(4.4) \quad \text{adj}(\mathcal{X}) = \left\{ y \in \mathbb{C}^n \mid |\langle y, x \rangle| \leq 1 \text{ for all } x \in \mathcal{X} \right\}$$

will be called the adjoint of $\mathcal{X}$.

It is immediately seen that $\text{adj}(\mathcal{X})$ is closed and absolutely convex.

DEFINITION 4.5. *We shall say that a bounded set $\mathcal{P}^* \subset \mathbb{C}^n$ is a b.c.p. of adjoint type (a.b.c.p.) if there exists a finite set $\mathcal{X} = \{x^{(i)}\}_{1 \leq i \leq m}$ of vectors such that $\text{span}(\mathcal{X}) = \mathbb{C}^n$ and*

$$(4.5) \quad \mathcal{P}^* = \text{adj}(\mathcal{X}) = \left\{ y \in \mathbb{C}^n \mid |\langle y, x^{(i)} \rangle| \leq 1, \ i = 1, \dots, m \right\}.$$

Moreover, if $\text{adj}(\mathcal{X}') \supset \text{adj}(\mathcal{X})$ for all $\mathcal{X}' \subset \mathcal{X}$, then $\mathcal{X}$ will be called an essential system of facets for $\mathcal{P}^*$, whereas any vector $ux^{(i)}$ with $u \in \mathbb{C}$, $|u| = 1$, will be called a facet of $\mathcal{P}^*$.

Unlike the case of classical polytopes in $\mathbb{R}^n$, it is not true that the class of b.c.p.'s coincides with the class of a.b.c.p.'s. Indeed, for every b.c.p. $\mathcal{P}$, the equality $\mathcal{P} = \text{adj}(\mathcal{X})$ implies that $\mathcal{X}$ is an infinite set of vectors and, analogously, the same implication holds whenever we express an a.b.c.p. $\mathcal{P}^*$ in the form $\mathcal{P}^* = \text{absco}(\mathcal{X})$. In fact, although the essential system of vertices (facets) $\mathcal{X}$ of a b.c.p. $\mathcal{P}$ (resp., of an a.b.c.p. $\mathcal{P}^*$) is finite, the total number of vertices (facets) is infinite. An interesting geometric property of the boundary of an a.b.c.p. $\mathcal{P}^*$ is that it is *piecewise algebraic* in the following sense.

PROPOSITION 4.6. *Let $\mathcal{P}^*$ be an a.b.c.p. and $\mathcal{X} = \{x^{(i)}\}_{1 \leq i \leq m}$ be an essential system of facets for $\mathcal{P}^*$. Moreover, let $\Phi : \mathbb{C}^n \rightarrow \mathbb{R}^{2n}$ be the standard vector space isomorphism such that $\Phi(x) = \begin{bmatrix} \Re(x) \\ \Im(x) \end{bmatrix}$ for all $x = \Re(x) + i\Im(x) \in \mathbb{C}^n$. Then the boundary of $\Phi(\mathcal{P}^*)$ is contained in the zero set of a polynomial $p(z_1, \dots, z_{2n}) \in \mathbb{R}[z_1, \dots, z_{2n}]$ of degree $2m$ such that $p(0, \dots, 0) \neq 0$.*

One of the most important relationships between b.c.p.'s and a.b.c.p.'s is given by the following result that, in the literature, is often referred to as the *bipolar theorem*.

THEOREM 4.7. *Let $\mathcal{P}$ be a b.c.p. and let $\mathcal{P}^* = \text{adj}(\mathcal{P})$. Then it holds that*

$$(4.6) \quad \mathcal{P} = \text{adj}(\mathcal{P}^*) = \text{adj}(\text{adj}(\mathcal{P})).$$

Conversely, let $\mathcal{P}^*$ be an a.b.c.p. and let $\mathcal{P} = \text{adj}(\mathcal{P}^*)$. Then it holds that

$$(4.7) \quad \mathcal{P}^* = \text{adj}(\mathcal{P}) = \text{adj}(\text{adj}(\mathcal{P}^*)).$$

The next two propositions state that, given a b.c.p. $\mathcal{P}$ (an a.b.c.p. $\mathcal{P}^*$), the essential system of vertices (facets) $\mathcal{X}$ is uniquely determined modulo scalar factors of modulus equal to 1.

PROPOSITION 4.8. *Assume that $\mathcal{X} = \{x^{(i)}\}_{1 \leq i \leq m}$ and $\bar{\mathcal{X}} = \{\bar{x}^{(i)}\}_{1 \leq i \leq k}$ are two essential systems of vertices for a b.c.p. $\mathcal{P}$. Then $k = m$ and, for each $i = 1, \dots, m$, there exist j_i , $1 \leq j_i \leq m$, and $u_i \in \mathbb{C}$, $|u_i| = 1$, such that $\bar{x}^{(i)} = u_i x^{(j_i)}$.*

PROPOSITION 4.9. *Assume that $\mathcal{X} = \{x^{(i)}\}_{1 \leq i \leq m}$ and $\bar{\mathcal{X}} = \{\bar{x}^{(i)}\}_{1 \leq i \leq k}$ are two essential systems of facets for an a.b.c.p. $\mathcal{P}^*$. Then $k = m$ and, for each $i = 1, \dots, m$, there exist j_i , $1 \leq j_i \leq m$, and $u_i \in \mathbb{C}$, $|u_i| = 1$, such that $\bar{x}^{(i)} = u_i x^{(j_i)}$.*

Now we extend the concept of *polytope norm* to the complex case in a straightforward way.

LEMMA 4.10. *Any b.c.p. $\mathcal{P}$ is the unit ball of a norm $\|\cdot\|_{\mathcal{P}}$ on $\mathbb{C}^n$.*

DEFINITION 4.11. *We shall call any norm $\|\cdot\|_{\mathcal{P}}$ whose unit ball is a b.c.p. $\mathcal{P}$ a complex polytope norm.*

LEMMA 4.12. *Any a.b.c.p. $\mathcal{P}^*$ is the unit ball of a norm $\|\cdot\|_{\mathcal{P}^*}$ on $\mathbb{C}^n$.*

DEFINITION 4.13. *We shall call any norm $\|\cdot\|_{\mathcal{P}^*}$ whose unit ball is an a.b.c.p. $\mathcal{P}^*$ an adjoint complex polytope norm.*

An important link between polytope norms and adjoint polytope norms is illustrated by the following theorem.

THEOREM 4.14. *Let $\mathcal{P}$ be a b.c.p. and let $\|\cdot\|_{\mathcal{P}}$ be the corresponding complex polytope norm. Then, for any $z \in \mathbb{C}^n$, it holds that*

$$(4.8) \quad \|z\|_{\mathcal{P}} = \min \left\{ \sum_{i=1}^m |\lambda_i| \mid z = \sum_{i=1}^m \lambda_i x^{(i)} \right\} = \max_{y \in \partial \mathcal{P}^*} |\langle z, y \rangle|,$$

where $\mathcal{P}^* = \text{adj}(\mathcal{P})$ and $\mathcal{X} = \{x^{(i)}\}_{1 \leq i \leq m}$ is an essential system of vertices for $\mathcal{P}$. Analogously, let $\mathcal{P}^*$ be an a.b.c.p. and let $\|\cdot\|_{\mathcal{P}^*}$ be the corresponding adjoint complex polytope norm. Then, for any $z \in \mathbb{C}^n$, it holds that

$$(4.9) \quad \|z\|_{\mathcal{P}^*} = \max_{1 \leq i \leq m} |\langle z, x^{(i)} \rangle| = \max_{x \in \partial \mathcal{P}} |\langle z, x \rangle|,$$

where $\mathcal{P} = \text{adj}(\mathcal{P}^*)$ and $\mathcal{X} = \{x^{(i)}\}_{1 \leq i \leq m}$ is an essential system of facets for $\mathcal{P}^*$.

COROLLARY 4.15. Let $\mathcal{P}$ be a b.c.p. and let $\|\cdot\|_{\mathcal{P}}$ be the corresponding complex polytope norm. Moreover, let $\mathcal{P}^* = \text{adj}(\mathcal{P})$ and let $\|\cdot\|_{\mathcal{P}^*}$ be the corresponding adjoint complex polytope norm. Then, for any complex $n \times n$ -matrix A and its adjoint A^* , it holds that

$$(4.10) \quad \|A^*\|_{\mathcal{P}^*} = \|A\|_{\mathcal{P}}.$$

The next theorem shows that the set of the complex polytope norms is *dense* in the set of all norms defined on $\mathbb{C}^n$ and that, consequently, the corresponding set of induced matrix complex polytope norms is *dense* in the set of all induced $n \times n$ -matrix norms.

THEOREM 4.16. Let $\|\cdot\|$ be a norm on $\mathbb{C}^n$. Then for any $\epsilon > 0$ there exists a b.c.p. $\mathcal{P}_\epsilon$ whose corresponding complex polytope norm $\|\cdot\|_\epsilon$ satisfies the inequalities

$$(4.11) \quad \|x\| \leq \|x\|_\epsilon \leq (1 + \epsilon)\|x\| \quad \text{for all } x \in \mathbb{C}^n.$$

Moreover, denoting by $\|\cdot\|$ and $\|\cdot\|_\epsilon$ also the corresponding induced matrix norms, it holds that

$$(4.12) \quad (1 + \epsilon)^{-1}\|A\| \leq \|A\|_\epsilon \leq (1 + \epsilon)\|A\| \quad \text{for all } A \in \mathbb{C}^{n \times n}.$$

5. The main results. Complex polytope norms play a particular role in our results. In fact, Theorem 4.16 immediately implies the following refinement of Proposition 2.1.

PROPOSITION 5.1. The spectral radius of a bounded family $\mathcal{F}$ of complex $n \times n$ -matrices is characterized by the equality

$$(5.1) \quad \rho(\mathcal{F}) = \inf_{\|\cdot\| \in \mathcal{N}_{pol}} \|\mathcal{F}\|,$$

where $\mathcal{N}_{pol}$ denotes the set of all possible induced $n \times n$ -matrix complex polytope norms.

The natural question arises of whether a nondefective family admits an extremal complex polytope norm. Before trying to answer the above question we shall give an important necessary condition for the existence of an extremal complex polytope norm in Theorem 5.7. It is based on Theorem 5.3, which was proved by Lagarias and Wang [LagWan].

DEFINITION 5.2. A norm $\|\cdot\|$ defined on $\mathbb{R}^d$ is said to be piecewise algebraic if the boundary of its unit ball is contained in the zero set of a polynomial $p(z_1, \dots, z_d) \in \mathbb{R}[z_1, \dots, z_d]$ such that $p(0, \dots, 0) \neq 0$.

THEOREM 5.3. Let $\mathcal{F} = \{A^{(i)}\}_{1 \leq i \leq m}$ be a finite nondefective family of real $d \times d$ -matrices and assume that there exists an extremal piecewise algebraic norm $\|\cdot\|$. Then the family $\mathcal{F}$ has at least an s.m.p. $\bar{P}$.

DEFINITION 5.4. *Given a bounded family $\mathcal{F}$ of complex $n \times n$ -matrices, we say that the family*

$$\mathcal{F}^* = \{A^* \mid A \in \mathcal{F}\},$$

where A^* is the adjoint of the matrix A , is the adjoint family of $\mathcal{F}$.

The proof of the next lemma is obvious.

LEMMA 5.5. *Let $\mathcal{F}$ be a bounded family of complex $n \times n$ -matrices and let $\mathcal{F}^*$ be its adjoint family. Then*

$$\rho(\mathcal{F}) = \rho(\mathcal{F}^*).$$

Moreover, a product $\bar{P} \in \Sigma(\mathcal{F})$ is an s.m.p. of $\mathcal{F}$ if and only if the adjoint product $\bar{P}^* \in \Sigma(\mathcal{F}^*)$ is an s.m.p. of $\mathcal{F}^*$.

The next result is a straightforward consequence of Lemma 5.5 and Corollary 4.15 with $\mathcal{P}^* = \text{adj}(\mathcal{P})$.

LEMMA 5.6. *A bounded nondefective family $\mathcal{F}$ of complex $n \times n$ -matrices has an extremal complex polytope norm $\|\cdot\|_{\mathcal{P}}$ if and only if the adjoint family $\mathcal{F}^*$ has an extremal adjoint complex polytope norm $\|\cdot\|_{\mathcal{P}^*}$.*

THEOREM 5.7. *Let $\mathcal{F} = \{A^{(i)}\}_{1 \leq i \leq m}$ be a finite nondefective family of complex $n \times n$ -matrices and assume that there exists an extremal complex polytope norm $\|\cdot\|_{\mathcal{P}}$. Then the family $\mathcal{F}$ has at least an s.m.p. $\bar{P}$.*

Proof. Lemma 5.6 implies that the adjoint family $\mathcal{F}^*$ has an extremal adjoint complex polytope norm $\|\cdot\|_{\mathcal{P}^*}$. Therefore, if we show that the adjoint family $\mathcal{F}^*$ has an s.m.p. $\bar{P}^*$, then the proof is complete by virtue of Lemma 5.5. To this aim, we consider the standard vector space isomorphism Φ defined in Proposition 4.6. It naturally induces a matrix space homomorphism

$$\Phi : \mathbb{C}^{n \times n} \longrightarrow \mathbb{R}^{2n \times 2n} \quad \text{such that} \quad \Phi(A) = \begin{bmatrix} \Re(A) & -\Im(A) \\ \Im(A) & \Re(A) \end{bmatrix}.$$

It is easy to see that Φ preserves matrix products and matrix-vector products, that is,

$$(5.2) \quad \Phi(AB) = \Phi(A)\Phi(B) \quad \text{for all } A, B \in \mathbb{C}^{n \times n}$$

and

$$(5.3) \quad \Phi(Ay) = \Phi(A)\Phi(y) \quad \text{for all } A \in \mathbb{C}^{n \times n} \text{ and } y \in \mathbb{C}^n.$$

It is also easy to see that Φ preserves convexity and that, in particular, if $\mathcal{B} \subset \mathbb{C}^n$ is the unit ball of a norm on $\mathbb{C}^n$, which is given by the Minkowski functional (see Heuser [Heu])

$$(5.4) \quad \|y\|_{\mathcal{B}} = \inf\{r > 0 \mid y \in r\mathcal{B}\},$$

then $\Phi(\mathcal{B}) \subset \mathbb{R}^{2n}$ is the unit ball of a norm on $\mathbb{R}^{2n}$, which is given by the Minkowski functional

$$(5.5) \quad \|z\|_{\Phi(\mathcal{B})} = \inf\{r > 0 \mid z \in r\Phi(\mathcal{B})\}.$$

In view of (5.4) and (5.5), we immediately have that

$$\|\Phi(y)\|_{\Phi(\mathcal{B})} = \|y\|_{\mathcal{B}} \quad \text{for all } y \in \mathbb{C}^n$$

and hence that, because of (5.3), for the corresponding induced matrix norms,

$$(5.6) \quad \|\Phi(A)\|_{\Phi(\mathcal{B})} = \|A\|_{\mathcal{B}} \quad \text{for all } A \in \mathbb{C}^{n \times n}.$$

Now consider the a.b.c.p. $\mathcal{P}^*$, that is, the unit ball of the adjoint complex polytope norm $\|\cdot\|_{\mathcal{P}^*}$ such that

$$\|\mathcal{F}^*\|_{\mathcal{P}^*} = \rho(\mathcal{F}^*).$$

Since, as it is easy to see, Φ preserves matrix eigenvalues with their algebraic and geometric multiplicities, by virtue of (5.2) and (5.6), we can state that the corresponding family of real $2n \times 2n$ -matrices $\Phi(\mathcal{F}^*)$ has the extremal norm $\|\cdot\|_{\Phi(\mathcal{P}^*)}$. Now, Proposition 4.6 assures that $\|\cdot\|_{\Phi(\mathcal{P}^*)}$ is a piecewise algebraic norm on $\mathbb{R}^{2n}$ (see Definition 5.2). Therefore, Theorem 5.3 implies that the family $\Phi(\mathcal{F}^*)$ has an s.m.p. $\Phi(\bar{P}^*) \in \Sigma(\Phi(\mathcal{F}^*))$. Consequently, $\bar{P}^*$ is an s.m.p. for $\mathcal{F}^*$. $\square$

Our aim is to reverse Theorem 5.7, that is, to prove the *complex polytope extremality (CPE) theorem*. So far we are not able to find any counterexample and not able to give a proof either. So we limit ourselves to the formulation of the following conjecture.

CONJECTURE 5.1 (CPE conjecture). *Assume that a finite family of complex $n \times n$ -matrices $\mathcal{F} = \{A^{(i)}\}_{1 \leq i \leq m}$ is nondefective and has at least an s.m.p. $\bar{P}$. Then there exists an extremal complex polytope norm for $\mathcal{F}$.*

We shall be able to prove a weaker version of the above conjecture with the forthcoming *small CPE theorem* at the cost of some additional hypotheses on the family $\mathcal{F}$. To this aim, for any vector $x \in \mathbb{C}^n$ and for any normalized family $\hat{\mathcal{F}}$, we define the set

$$\mathcal{T}[\hat{\mathcal{F}}, x] = \{x\} \cup \{\hat{P}x \mid \hat{P} \in \Sigma(\hat{\mathcal{F}})\},$$

i.e., the *trajectory* obtained by applying all the normalized products $\hat{P}$ of matrices of $\hat{\mathcal{F}}$ to the vector x . For the convenience of the reader we recall the following characterization.

PROPOSITION 5.8. *Let $\mathcal{F}$ be a bounded family of complex $n \times n$ -matrices and let $x \in \mathbb{C}^n$. Then $\text{span}(\mathcal{T}[\hat{\mathcal{F}}, x])$ is the smallest linear subspace V of $\mathbb{C}^n$ containing x such that $\mathcal{F}(V) \subseteq V$.*

Proof. It is clear that, if V is a linear subspace of $\mathbb{C}^n$ containing x such that $\mathcal{F}(V) \subseteq V$, then it must necessarily contain the whole trajectory $\mathcal{T}[\hat{\mathcal{F}}, x]$. Vice versa, let $y \in \text{span}(\mathcal{T}[\hat{\mathcal{F}}, x])$. Then there exist k vectors $x^{(1)}, \dots, x^{(k)} \in \mathcal{T}[\hat{\mathcal{F}}, x]$ and k complex numbers $\alpha_1, \dots, \alpha_k$ (with $k \leq n$) such that $y = \sum_{i=1}^k \alpha_i x^{(i)}$. Now, if $A \in \mathcal{F}$, then $Ay = \sum_{i=1}^k \alpha_i Ax^{(i)}$. Since $\rho(\mathcal{F})^{-1}Ax^{(i)} \in \mathcal{T}[\hat{\mathcal{F}}, x]$, we have that $Ay \in \text{span}(\mathcal{T}[\hat{\mathcal{F}}, x])$, too. Thus the proof is complete. $\square$

COROLLARY 5.9. *Let $\mathcal{F}$ be an irreducible bounded family of complex $n \times n$ -matrices and let $x \in \mathbb{C}^n$, $x \neq 0$. Then*

$$(5.7) \quad \text{span}(\mathcal{T}[\hat{\mathcal{F}}, x]) = \mathbb{C}^n.$$

Proof. Since $\mathcal{F}$ is irreducible, the matrices of $\mathcal{F}$ cannot have a common nontrivial invariant linear subspace (see Definition 2.5). As a consequence, the smallest linear subspace V of $\mathbb{C}^n$ containing $x \neq 0$ such that $\mathcal{F}(V) \subseteq V$ is $\mathbb{C}^n$. $\square$

For a general family of matrices $\mathcal{F}$, the sets of the type

$$(5.8) \quad \mathcal{S}[\hat{\mathcal{F}}, x] = \overline{\text{absco}(\mathcal{T}[\hat{\mathcal{F}}, x])}$$

play an important role.

PROPOSITION 5.10. *Let $\mathcal{F}$ be a nondefective bounded family of complex $n \times n$ -matrices and, given a vector $x \in \mathbb{C}^n$, let (5.7) hold. Then the set $\mathcal{S}[\hat{\mathcal{F}}, x]$ is the unit ball of an extremal norm for $\mathcal{F}$.*

Proof. Since $\hat{\mathcal{F}}$ is nondefective, the trajectory $\mathcal{T}[\hat{\mathcal{F}}, x]$ is bounded for any vector $x \in \mathbb{C}^n$. Thus, because of (5.7), as in the proof of Theorem 5.7 we have that the Minkowski functional associated with $\mathcal{S} = \mathcal{S}[\hat{\mathcal{F}}, x]$ is a norm on $\mathbb{C}^n$. In order to verify that it is extremal, it is sufficient to observe that $A(\mathcal{S}) \subseteq \rho(\mathcal{F})\mathcal{S}$ for all $A \in \mathcal{F}$. $\square$

By virtue of the foregoing result, it appears evident that it is interesting to find conditions under which $\mathcal{S}[\hat{\mathcal{F}}, x]$ is generated by a finite number of points of the trajectory $\mathcal{T}[\hat{\mathcal{F}}, x]$. So, if (5.7) holds, the set $\mathcal{S}[\hat{\mathcal{F}}, x]$ is a b.c.p. and we can give a positive answer to the question of the existence of extremal complex polytope norms for nondefective families. In particular, in view of Corollaries 2.7 and 5.9, we can give a positive answer for irreducible families.

In order to state the main result of this paper, we focus our attention on families that satisfy some particular properties.

DEFINITION 5.11. *An eigenvector $x \neq 0$ of a matrix P related to an eigenvalue λ with $|\lambda| = \rho(P)$ is said to be a leading eigenvector of P .*

DEFINITION 5.12. *Let $\mathcal{F}$ be a nondefective bounded family of complex $n \times n$ -matrices. A leading eigenvector $x \neq 0$ of either an s.m.p. $\bar{P}$ of $\mathcal{F}$ or of an l.s.m.p. $\tilde{P}$ of the normalized family $\hat{\mathcal{F}}$ is said to be a leading eigenvector of $\mathcal{F}$ (and of $\hat{\mathcal{F}}$, too).*

Remark 5.1. Because of Theorem 2.11, any nondefective bounded family $\mathcal{F}$ has at least one leading eigenvector.

DEFINITION 5.13. *Let $\mathcal{F}$ be a family of complex $n \times n$ -matrices. A set $\mathcal{X} \subset \mathbb{C}^n$ is said to be $\mathcal{F}$ -cyclic if for any pair $(x, y) \in \mathcal{X} \times \mathcal{X}$ there exist $\alpha, \beta \in \mathbb{C}$ with*

$$(5.9) \quad |\alpha| \cdot |\beta| = 1$$

and two (finite) normalized products $\hat{P}, \hat{Q} \in \Sigma(\hat{\mathcal{F}})$ such that

$$y = \alpha \hat{P}x \quad \text{and} \quad x = \beta \hat{Q}y.$$

Remark 5.2. Because of (5.9), the normalized products $\hat{P}\hat{Q}$ and $\hat{Q}\hat{P}$ determined in the above definition are s.m.p. of the normalized family $\hat{\mathcal{F}}$, and the set $\mathcal{X}$ is necessarily included in the set $\mathcal{E}$ of the leading eigenvectors of the family $\mathcal{F}$.

DEFINITION 5.14. *A nondefective bounded family $\mathcal{F}$ of complex $n \times n$ -matrices is said to be asymptotically simple if the set $\mathcal{E}$ of its leading eigenvectors is finite (modulo scalar nonzero factors) and $\mathcal{F}$ -cyclic.*

As in section 2, we shall say that a matrix Q is an *infinite power* of another matrix P if it is a limit point of the sequence $\{P^k\}_{k \geq 1}$. Observe that any eigenvalue λ of an infinite power Q of a matrix P satisfies either $|\lambda| = 1$ or $\lambda = 0$, since these are the only two possible limit values of the numeric power sequence $\{|\mu|^k\}_{k \geq 1}$ whenever $|\mu| \leq 1$. Moreover, given a nondefective matrix P with $\rho(P) = 1$, there exists at least an infinite power Q of P with an eigenvalue $\lambda = 1$, whose multiplicity is equal to the sum of the multiplicities of all the eigenvalues μ of P with $|\mu| = 1$. This easily follows from the fact that the power sequence $\{\mu^k\}_{k \geq 1}$ has the limit point 1 whenever $|\mu| = 1$ (see, for example, Hardy and Wright [HarWri]).

Remark 5.3. It follows from the above observations that, for a (nondefective) asymptotically simple family $\mathcal{F}$, each s.m.p. $\bar{P}$ and each l.s.m.p. $\tilde{P}$ has only one leading eigenvector (modulo scalar nonzero factors). Otherwise there would exist at

least one l.s.m.p. of the normalized family $\hat{\mathcal{F}}$, obtained as an infinite power, with an eigenspace of dimension ≥ 2 related to the eigenvalue $\lambda = 1$. This would contradict the finiteness (modulo scalar nonzero factors) of the set $\mathcal{E}$ of leading eigenvectors.

Observe that all the cyclic permutations of a product $\bar{P}$ have the same eigenvalues with the same multiplicities. Thus, if $\bar{P} = A^{(i_{k^*})} \cdot \dots \cdot A^{(i_1)}$ is an s.m.p. for a family $\mathcal{F}$, then each of its cyclic permutations

$$A^{(i_s)} \cdot \dots \cdot A^{(i_1)} A^{(i_{k^*})} \cdot \dots \cdot A^{(i_{s+1})}, \quad s = 1, \dots, k^* - 1,$$

still is an s.m.p. for $\mathcal{F}$, along with all the powers of $\bar{P}$ and their cyclic permutations.

DEFINITION 5.15. *Let $\mathcal{F}$ be a family of complex $n \times n$ -matrices. An s.m.p. $\bar{P}$ is said to be minimal if it is not a power of another s.m.p. of $\mathcal{F}$.*

It is clear that, for any s.m.p. $\bar{P}$ of a family $\mathcal{F}$, it holds that either $\bar{P}$ is minimal or $\bar{P}$ is a power of another s.m.p., which is minimal.

We have the following characterization of asymptotically simple families.

PROPOSITION 5.16. *A nondefective bounded family $\mathcal{F}$ of complex $n \times n$ -matrices is asymptotically simple if and only if it has a minimal s.m.p. $\bar{P}$ with only one leading eigenvector (modulo scalar nonzero factors) such that the set $\mathcal{E}$ of the leading eigenvectors of $\mathcal{F}$ is equal to the set of the leading eigenvectors of $\bar{P}$ and of its cyclic permutations.*

Proof. Given an s.m.p. $\bar{P}$ with only one leading eigenvector, the set of the leading eigenvectors of $\bar{P}$ and of its cyclic permutations is finite (modulo scalar nonzero factors). Indeed, it consists of k elements, k being not greater than the number of factors of $\bar{P}$. On the other hand, this set of leading eigenvectors is clearly $\mathcal{F}$ -cyclic and, therefore, the sufficiency is proved. In order to prove the necessity, let $x_1, \dots, x_s \in \mathbb{C}^n$ form a set of representatives (modulo scalar nonzero factors) of all the leading eigenvectors of $\mathcal{F}$. Since $\mathcal{F}$ is asymptotically simple, they are finitely many and, for any $i = 1, \dots, s$, there exist $\alpha_i, \beta_i \in \mathbb{C}$ with

$$|\alpha_i| \cdot |\beta_i| = 1$$

and two (finite) normalized products $\hat{P}_i, \hat{Q}_i \in \Sigma(\hat{\mathcal{F}})$ such that

$$x_{i+1} = \alpha_i \hat{P}_i x_i \quad \text{and} \quad x_i = \beta_i \hat{Q}_i x_{i+1},$$

where it is understood that $x_{s+1} = x_1$. Therefore, we have that

$$x_1 = \alpha_1 \cdot \dots \cdot \alpha_s \hat{P}_s \cdot \dots \cdot \hat{P}_1 x_1 = \beta_1 \cdot \dots \cdot \beta_s \hat{Q}_1 \cdot \dots \cdot \hat{Q}_s x_1,$$

where

$$|\alpha_1 \dots \alpha_s| \cdot |\beta_1 \dots \beta_s| = 1.$$

Now, since $\rho(\hat{P}_s \dots \hat{P}_1) \leq 1$ and $\rho(\hat{Q}_1 \dots \hat{Q}_s) \leq 1$, it follows that

$$|\alpha_1 \dots \alpha_s| = 1 \quad \text{and} \quad |\beta_1 \dots \beta_s| = 1,$$

which implies

$$\rho(\hat{P}_s \dots \hat{P}_1) = 1 \quad \text{and} \quad \rho(\hat{Q}_1 \dots \hat{Q}_s) = 1.$$

So we can conclude that the matrix $\hat{P} = \hat{P}_s \dots \hat{P}_1$ is an s.m.p. of $\hat{\mathcal{F}}$ such that the set of the leading eigenvectors of $\hat{P}$ and of its cyclic permutations includes (and thus is equal

to) the set $\mathcal{E}$ of the leading eigenvectors of $\mathcal{F}$. The fact that x_1 is the only leading eigenvector (modulo nonzero scalar factors) of $\hat{P}$ is assured by the considerations in Remark 5.3. Finally, $\hat{P}$ can be clearly assumed to be minimal, so that is the normalization of the desired minimal s.m.p. $\bar{P}$. $\square$

The following definition selects a particular class of asymptotically simple families.

DEFINITION 5.17. *A nondefective bounded family $\mathcal{F}$ of complex $n \times n$ -matrices is said to be absolutely asymptotically simple if it is asymptotically simple and has a unique minimal s.m.p. $\bar{P}$ (modulo cyclic permutations).*

It is clear that, for absolutely asymptotically simple families, the unique minimal s.m.p. $\bar{P}$ coincides with the minimal s.m.p. given by the characterizing proposition, Proposition 5.16. Moreover, the cardinality of the set $\mathcal{E}$ of its leading eigenvectors (modulo scalar nonzero factors) is equal to the number of factors of $\bar{P}$. The result we are going to prove is the main result of this paper. It was inspired by a conjecture formulated by Maesumi [Mae1] (see also [Mae2]). Although not strictly necessary, from now on we shall consider trajectories $\mathcal{T}[\hat{\mathcal{F}}, x]$ such that condition (5.7) is satisfied. On the other hand, this is not restrictive for our purposes, since we are interested in the case that $\mathcal{S}[\hat{\mathcal{F}}, x]$ is the unit ball of a norm.

THEOREM 5.18 (small CPE theorem). *Assume that a finite family $\mathcal{F} = \{A^{(i)}\}_{1 \leq i \leq m}$ of complex $n \times n$ -matrices is nondefective and asymptotically simple. Moreover, let $x \neq 0$ be a leading eigenvector of $\mathcal{F}$ and assume that (5.7) is satisfied. Then the set*

$$(5.10) \quad \partial\mathcal{S}[\hat{\mathcal{F}}, x] \bigcap \mathcal{T}[\hat{\mathcal{F}}, x]$$

is finite modulo scalar factors of unitary modulus. As a consequence, there exist a finite number of normalized products $\hat{P}^{(1)}, \dots, \hat{P}^{(s)} \in \Sigma(\hat{\mathcal{F}})$ such that

$$(5.11) \quad \mathcal{S}[\hat{\mathcal{F}}, x] = \text{absco}\left(\{x, \hat{P}^{(1)}x, \dots, \hat{P}^{(s)}x\}\right),$$

so that $\mathcal{S}[\hat{\mathcal{F}}, x]$ is a b.c.p.

Before giving the proof of this theorem, we state a useful technical lemma.

LEMMA 5.19. *Let $\mathcal{F} = \{A^{(i)}\}_{1 \leq i \leq m}$ be a nondefective finite family of complex $n \times n$ -matrices and, given a vector $x \in \mathbb{C}^n$, assume that (5.7) is satisfied and that the (bounded) set $\partial\mathcal{S}[\hat{\mathcal{F}}, x] \bigcap \mathcal{T}[\hat{\mathcal{F}}, x]$, modulo scalar factors of unitary modulus, is not finite. Then there exists a sequence of distinct vectors*

$$x^{(k)} \in \partial\mathcal{S}[\hat{\mathcal{F}}, x] \bigcap \mathcal{T}[\hat{\mathcal{F}}, x]$$

with $x^{(1)} = x$ such that, for all $k \geq 1$,

$$x^{(k+1)} = \hat{A}^{(\ell_k)} x^{(k)} \quad \text{for some } \ell_k \in \{1, \dots, m\},$$

where $\hat{A}^{(i)} = A^{(i)}/\rho(\mathcal{F}) \in \hat{\mathcal{F}}$, $1 \leq i \leq m$, and such that, whenever $k \neq h$,

$$x^{(k)} \neq ux^{(h)} \quad \text{for all } u \in \mathbb{C} \text{ with } |u| = 1.$$

Proof. We prove the result by induction. Define $x^{(1)} = x$. By hypothesis, there exist infinitely many distinct vectors of the type $\hat{Q}x^{(1)}$, no two of which are multiples of each other by a scalar factor of unitary modulus, with $\hat{Q} \in \Sigma(\hat{\mathcal{F}})$ and $\hat{Q}x^{(1)} \in \partial\mathcal{S}[\hat{\mathcal{F}}, x]$. Since $\hat{\mathcal{F}}$ is finite, there exists a matrix $\hat{A}^{(\ell_1)}$ for which, with $x^{(2)} = \hat{A}^{(\ell_1)}x^{(1)}$, it holds that $x^{(2)} \neq ux^{(1)}$ for all $u \in \mathbb{C}$ with $|u| = 1$ and that infinitely many distinct

vectors of the type $\hat{Q}x^{(2)}$, not proportional to one another by scalar factors of unitary modulus, exist such that $\hat{Q} \in \Sigma(\hat{\mathcal{F}})$ and $\hat{Q}x^{(2)} \in \partial\mathcal{S}[\hat{\mathcal{F}}, x]$. Since (5.7) holds, by Propositions 5.10 and 3.4 we immediately have that $x^{(2)} \in \partial\mathcal{S}[\hat{\mathcal{F}}, x]$.

Now assume that there exist $k \geq 2$ distinct vectors $x^{(1)} = x, x^{(2)}, \dots, x^{(k)} \in \partial\mathcal{S}[\hat{\mathcal{F}}, x]$, such that $x^{(i)} \neq ux^{(j)}$ for all $u \in \mathbb{C}$ with $|u| = 1$ if $i \neq j$, and such that $x^{(j)} = \hat{A}^{(\ell_{j-1})}x^{(j-1)}$ for some $\ell_{j-1} \in \{1, \dots, m\}$ and for which infinitely many distinct vectors of the type $\hat{Q}x^{(j)}$, not proportional to one another by scalar factors of unitary modulus, exist with $\hat{Q} \in \Sigma(\hat{\mathcal{F}})$ and $\hat{Q}x^{(j)} \in \Sigma(\hat{\mathcal{F}})$. Still by the finiteness of $\hat{\mathcal{F}}$, there exists another matrix $\hat{A}^{(\ell_k)}$ for which, with $x^{(k+1)} = \hat{A}^{(\ell_k)}x^{(k)}$, it holds that $x^{(k+1)} \neq ux^{(j)}$ for all $u \in \mathbb{C}$ with $|u| = 1$, $j = 1, \dots, k$, and that infinitely many distinct vectors of the type $\hat{Q}x^{(k+1)}$, not proportional to one another by scalar factors of unitary modulus, exist such that $\hat{Q} \in \Sigma(\hat{\mathcal{F}})$ and $\hat{Q}x^{(k+1)} \in \partial\mathcal{S}[\hat{\mathcal{F}}, x]$. Again, by Propositions 5.10 and 3.4, it turns out that $x^{(k+1)} \in \partial\mathcal{S}[\hat{\mathcal{F}}, x]$. This completes the induction step and the result is proved. $\square$

We remark that the finiteness of $\hat{\mathcal{F}}$ is essential in order to guarantee the result. This is clarified by the following example.

Example 5.1. Consider the infinite family $\mathcal{F} = \{A^{(i)}\}_{i \geq 1}$ of real 2×2 -matrices, where

$$A^{(i)} = \begin{bmatrix} \frac{1}{i} & 0 \\ 1 - \frac{1}{i} & 0 \end{bmatrix}.$$

Since

$$A^{(i)}A^{(j)} = \begin{bmatrix} \frac{1}{ij} & 0 \\ (1 - \frac{1}{i})\frac{1}{j} & 0 \end{bmatrix},$$

it is clear that $\rho(\mathcal{F}) = 1$. Moreover, the norm $\|\cdot\|_1$ defined by $\|[x_1, x_2]^T\|_1 = |x_1| + |x_2|$ is such that $\|\mathcal{F}\|_1 = 1$. Consequently, by Proposition 2.4, the family $\mathcal{F}$ is nondefective.

Now consider $x = [1, 0]^T$ (which is the leading eigenvector of $A^{(1)}$, the unique minimal s.m.p. of $\mathcal{F}$) and the corresponding trajectory $\mathcal{T}[\mathcal{F}, x]$. It is immediately seen that $\mathcal{S}[\mathcal{F}, x]$ is nothing but the unit ball of the norm $\|\cdot\|_1$ and that the infinitely many vectors

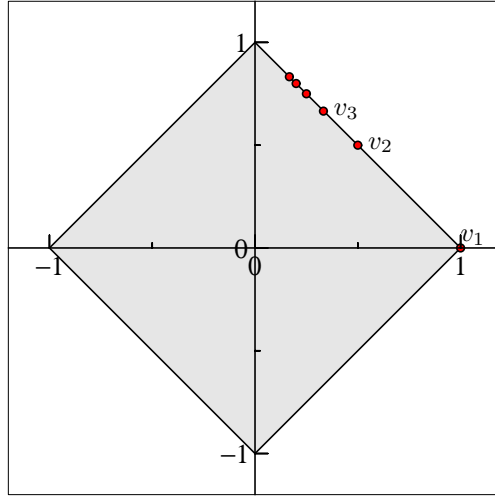
$$v^{(i)} = A^{(i)}x = \left[\frac{1}{i}, 1 - \frac{1}{i} \right]^T, \quad i \geq 1,$$

are not proportional to one another by scalar factors of unitary modulus and lie on $\partial\mathcal{S}[\mathcal{F}, x]$ (see Figure 5.1). Therefore, all the hypotheses of Lemma 5.19 are satisfied except the finiteness of $\mathcal{F}$.

On the other hand, all the remaining vectors of the trajectory are obtained by applying the matrices of the family $\mathcal{F}$ to the following vectors:

$$A^{(i)}v^{(j)} = \left[\frac{1}{ij}, \left(1 - \frac{1}{i}\right)\frac{1}{j} \right]^T, \quad i \geq 1 \text{ and } j \geq 2.$$

Now, all such vectors are inside $\mathcal{S}[\mathcal{F}, x]$, away from its boundary. Therefore, we can conclude that the sequence predicted by Lemma 5.19 does not exist. $\diamond$

FIG. 5.1. Real part of $\mathcal{S}[\hat{\mathcal{F}}, x]$ in Example 5.1.

Proof of Theorem 5.18. Denoting by $\mathcal{E}$ the set of the leading eigenvectors of the family $\mathcal{F}$, let us consider

$$\Xi = \mathcal{E} \cap \partial\mathcal{S}[\hat{\mathcal{F}}, x].$$

Since the family $\mathcal{F}$ is asymptotically simple and $\mathcal{S}[\hat{\mathcal{F}}, x]$ is the unit ball of a norm, the set Ξ is finite modulo scalar factors of unitary modulus. On the contrary, by contradiction, assume that the set

$$\partial\mathcal{S}[\hat{\mathcal{F}}, x] \cap \mathcal{T}[\hat{\mathcal{F}}, x],$$

even if considered modulo scalar factors of unitary modulus, is not finite, so that Lemma 5.19 can be applied to obtain the sequence $\{x^{(k)}\}_{k \geq 1}$ with $x^{(1)} = x$. Therefore, since $x^{(1)} \in \Xi \cap \mathcal{T}[\hat{\mathcal{F}}, x]$, there exists $j \geq 1$ such that

$$(5.12) \quad x^{(j+1)} \notin \Xi \quad \text{and} \quad x^{(i)} \in \Xi \cap \mathcal{T}[\hat{\mathcal{F}}, x] \quad \text{for all } i \leq j.$$

Since $\Sigma(\hat{\mathcal{F}})$ is bounded, the resulting sequence of normalized matrix products $\hat{B}^{(k)} = \hat{A}^{(\ell_{k-1})} \dots \hat{A}^{(\ell_{j+1})}$ such that $x^{(k)} = \hat{B}^{(k)} x^{(j+1)}$ has a subsequence $\{\hat{B}^{(k_s)}\}_{s \geq 1}$ that converges to a limit point $\hat{B}$ in $\overline{\Sigma(\hat{\mathcal{F}})}$. Therefore, also the subsequence of vectors $\{x^{(k_s)}\}_{s \geq 1}$ has a limit point $y = \hat{B}x^{(j+1)}$. Moreover, since $\partial\mathcal{S}[\hat{\mathcal{F}}, x]$ is closed, we have that

$$(5.13) \quad \hat{B}x^{(j+1)} \in \partial\mathcal{S}[\hat{\mathcal{F}}, x].$$

For each $s \geq 1$ there exists a matrix $\hat{R}^{(s)} \in \Sigma(\hat{\mathcal{F}})$ such that

$$\hat{B}^{(k_{s+1})} = \hat{R}^{(s)} \hat{B}^{(k_s)}.$$

Again for the boundedness of $\Sigma(\hat{\mathcal{F}})$, the sequence $\{\hat{R}^{(s)}\}_{s \geq 1}$ has a limit point $\hat{R}$ in $\Sigma(\hat{\mathcal{F}})$. By passing to the limit, this allows us to conclude that

$$\hat{B}x^{(j+1)} = \hat{R}\hat{B}x^{(j+1)}.$$

In other words, $\hat{R}$ is an l.s.m.p. of $\hat{\mathcal{F}}$ and

$$\hat{B}x^{(j+1)} \in \Xi.$$

Since $\mathcal{F}$ is asymptotically simple, the set Ξ is $\mathcal{F}$ -cyclic and, hence, there exist $\alpha, \beta \in \mathbb{C}$ with $|\alpha| \cdot |\beta| = 1$ and two (finite) normalized products $\hat{P}, \hat{Q} \in \Sigma(\hat{\mathcal{F}})$ such that

$$(5.14) \quad \hat{B}x^{(j+1)} = \alpha \hat{P}x^{(j)} \quad \text{and} \quad x^{(j)} = \beta \hat{Q} \hat{B}x^{(j+1)}.$$

This implies

$$\hat{Q} \hat{P} x^{(j)} = (\alpha\beta)^{-1} x^{(j)},$$

and thus, since $|(\alpha\beta)^{-1}| = 1$, by Proposition 5.10 and Theorem 3.5 we have that

$$\hat{P}x^{(j)} \in \partial\mathcal{S}[\hat{\mathcal{F}}, x].$$

Therefore, by (5.13) and (5.14), it turns out that

$$(5.15) \quad |\alpha| = |\beta| = 1.$$

In conclusion, we have

$$\beta^{-1} x^{(j+1)} = \beta^{-1} \hat{A}^{(\ell_j)} x^{(j)} = \hat{A}^{(\ell_j)} \hat{Q} \hat{B} x^{(j+1)}.$$

Now, since $\hat{A}^{(\ell_j)} \hat{Q} \hat{B} \in \overline{\Sigma(\hat{\mathcal{F}})}$ and (5.15) holds, we can state that $x^{(j+1)}$ is a leading eigenvector of $\mathcal{F}$, i.e.,

$$x^{(j+1)} \in \Xi,$$

in contradiction with (5.12). $\square$

Note that, if all the matrices of the family $\mathcal{F}$ are real and also if the starting leading eigenvector x is real, then Theorem 5.18 determines a classical polytope in $\mathbb{R}^n$.

The next results are useful for a deeper understanding of the structure of the b.c.p. $\mathcal{S}[\hat{\mathcal{F}}, x]$ obtained under the hypotheses of Theorem 5.18.

THEOREM 5.20. *Let the hypotheses of Theorem 5.18 hold. Then each leading eigenvector ξ of $\mathcal{F}$ in the set $\Xi = \mathcal{E} \cap \partial\mathcal{S}[\hat{\mathcal{F}}, x]$ satisfies one of the following two statements:*

- (a) ξ is a vertex of the b.c.p. $\mathcal{S}[\hat{\mathcal{F}}, x]$;
- (b) there exist $s \geq 2$ vertices $\xi_1, \dots, \xi_s$ of the b.c.p. $\mathcal{S}[\hat{\mathcal{F}}, x]$ such that

$$(5.16) \quad \xi_1, \dots, \xi_s \in \Xi \quad \text{and} \quad \xi \in \text{absco}(\{\xi_1, \dots, \xi_s\}).$$

Proof. Consider a leading eigenvector $\xi \in \Xi$ and assume that it is not a vertex of the b.c.p. $\mathcal{S}[\hat{\mathcal{F}}, x]$. Then there must exist $s \geq 2$ vertices $\xi_1, \dots, \xi_s$ of $\mathcal{S}[\hat{\mathcal{F}}, x]$ such that

$$\xi = \sum_{i=1}^s \lambda_i \xi_i$$

with

$$(5.17) \quad \lambda_i \neq 0, \quad i = 1, \dots, s, \quad \text{and} \quad \sum_{i=1}^s |\lambda_i| = 1.$$

We are left to prove that $\xi_1, \dots, \xi_s \in \Xi$.

Since ξ is a leading eigenvector of $\mathcal{F}$, there exists an l.s.m.p. (possibly an s.m.p.) $\tilde{P}$ of $\hat{\mathcal{F}}$ such that $\tilde{P}\xi = u\xi$ with $|u| = 1$. Thus, denoting by $\|\cdot\|$ the complex polytope norm determined by $\mathcal{S}[\hat{\mathcal{F}}, x]$, for any $k \geq 1$ we have

$$1 = \|\xi\| = \|\tilde{P}^k \xi\| \leq \sum_{i=1}^s |\lambda_i| \cdot \|\tilde{P}^k \xi_i\|.$$

Since $\|\tilde{P}^k \xi_i\| \leq 1$, in view of (5.17) we can claim that $\|\tilde{P}^k \xi_i\| = 1$, that is,

$$\tilde{P}^k \xi_i \in \partial\mathcal{S}[\hat{\mathcal{F}}, x], \quad i = 1, \dots, s.$$

Denoting by $\tilde{P}^\infty$ an infinite power of $\tilde{P}$, in the same way as we did in the proof of Theorem 5.18, we can prove that

$$\tilde{P}^\infty \xi_i \in \Xi, \quad i = 1, \dots, s.$$

On the other hand, since the set Ξ is $\mathcal{F}$ -cyclic and included in $\partial\mathcal{S}[\hat{\mathcal{F}}, x]$, again in the same way as in the proof of Theorem 5.18, we can prove that there exist (finite) normalized products $\hat{Q}, \hat{R}_i \in \Sigma(\hat{\mathcal{F}})$ and $v, w_i \in \mathbb{C}$ with $|v| = |w_i| = 1$ such that

$$x = v\hat{Q}\xi \quad \text{and} \quad \xi = w_i \hat{R}_i \tilde{P}^\infty \xi_i, \quad i = 1, \dots, s.$$

Moreover, since all the vertices of the b.c.p. $\mathcal{S}[\hat{\mathcal{F}}, x]$ obviously belong to $\mathcal{T}[\hat{\mathcal{F}}, x]$ (modulo scalar factors of unitary modulus), there exist (finite) normalized products $\hat{S}_i \in \Sigma(\hat{\mathcal{F}})$ such that

$$\xi_i = \hat{S}_i x, \quad i = 1, \dots, s.$$

In conclusion, we have

$$\xi_i = vw_i \hat{S}_i \hat{Q} \hat{R}_i \tilde{P}^\infty \xi_i, \quad i = 1, \dots, s,$$

where $|vw_i| = 1$, which implies $\xi_i \in \Xi$, $i = 1, \dots, s$. $\square$

COROLLARY 5.21. *Let the hypotheses of Theorem 5.18 hold and, moreover, let the family $\mathcal{F}$ be absolutely asymptotically simple. Then all the leading eigenvectors of $\mathcal{F}$ (in the set $\Xi = \mathcal{E} \cap \partial\mathcal{S}[\hat{\mathcal{F}}, x]$) are vertices of the b.c.p. $\mathcal{S}[\hat{\mathcal{F}}, x]$.*

Proof. Assume, by contradiction, that there exists a leading eigenvector $\xi \in \Xi$ which is not a vertex of the b.c.p. $\mathcal{S}[\hat{\mathcal{F}}, x]$. Then it necessarily satisfies statement (b) of Theorem 5.20.

On the other hand, there exists a unique normalized minimal s.m.p. $\hat{P}$ such that $\hat{P}\xi = u\xi$ with $|u| = 1$. Therefore, for each ξ_i appearing in statement (b) there exists a proper normalized right factor $\hat{P}_i$ of the s.m.p. $\hat{P}$ such that $\xi_i = u_i \hat{P}_i \xi$ with $|u_i| = 1$. Thus we obtain

$$\xi_i \in \text{absco}\left(\{\hat{P}_i \xi_1, \dots, \hat{P}_i \xi_s\}\right).$$

Now, since the essential system of vertices of a b.c.p. is unique modulo scalar factors of unitary modulus (see Proposition 4.8) and since ξ_i is a vertex of $\mathcal{S}[\hat{\mathcal{F}}, x]$, it necessarily holds that, for all $j = 1, \dots, s$,

$$\xi_i = v_j \hat{P}_i \xi_j \quad \text{with } |v_j| = 1.$$

In particular, we have that

$$\xi_i = v_i \hat{P}_i \xi_i \quad \text{with } |v_i| = 1;$$

that is, the proper normalized right factor $\hat{P}_i$ is an s.m.p., against the uniqueness of $\hat{P}$ (modulo cyclic permutations). $\square$

6. On the necessity of the assumptions in the small CPE theorem.

The assumptions of Theorem 5.18 may seem to be somewhat restrictive. On the other hand, now we present some simple examples that imply their necessity for the finiteness (modulo scalar factors of unitary modulus) of the set $\partial\mathcal{S}[\hat{\mathcal{F}}, x] \cap \mathcal{T}[\hat{\mathcal{F}}, x]$ in the general case. In this case we note that, for the actual construction of the unit ball of an extremal complex polytope norm by means of some suitable algorithm, the finiteness property mentioned above is strongly recommended. Indeed, one of the aims of our future work will be the improvement of the currently available algorithms for the computation and approximation of the spectral radius of a family of matrices by using the theoretical results developed in the present paper. In the following two examples the considered families are not asymptotically simple. More precisely, in Example 6.1 the set of leading eigenvectors is not finite (modulo scalar nonzero factors) and not $\mathcal{F}$ -cyclic either. In Example 6.2 the set is finite but not $\mathcal{F}$ -cyclic.

Example 6.1. Consider the 2×2 -matrix family $\mathcal{F} = \{A, B\}$, where

$$A = \begin{bmatrix} \cos(1) & -\sin(1) \\ \sin(1) & \cos(1) \end{bmatrix} \quad \text{and} \quad B = \begin{bmatrix} \frac{1}{2} & -\frac{i}{2} \\ 0 & 0 \end{bmatrix}.$$

The eigenvalues of A are e^i and e^{-i} with corresponding eigenvectors $[1, -i]^T$ and $[1, i]^T$, respectively. The eigenvalues of B are $\frac{1}{2}$ and 0 . It is easy to see that $\rho(\mathcal{F}) = 1$, that A is the unique minimal s.m.p. of $\mathcal{F}$, and that all the l.s.m.p.'s of $\mathcal{F}$ are infinite powers of A . Since the s.m.p. A has two leading eigenvectors, according to Remark 5.3, the identity matrix I is an l.s.m.p., so that all the vectors of $\mathbb{C}^2$ are leading eigenvectors of $\mathcal{F}$.

It is immediately seen that the set of the leading eigenvectors of the family $\mathcal{F}$ is not even $\mathcal{F}$ -cyclic. As a consequence of the violation of the hypotheses of Theorem 5.18, now we are going to see that, by setting

$$x = [1, i]^T,$$

which is a leading eigenvector of the s.m.p. A , the set $\mathcal{S}[\mathcal{F}, x]$ is not a b.c.p. In fact, it holds that

$$Bx = [1, 0]^T \quad \text{and} \quad A^k Bx = [\cos(k), \sin(k)]^T, \quad k \geq 1.$$

All these vectors lie on the unit circle $\mathcal{C}$ of $\mathbb{R}^2$ (see (4.3)) and form a set which is dense in $\mathcal{C}$. On the other hand,

$$BA^k Bx = \frac{1}{2} e^{-ik} Bx,$$

and, hence, we can conclude that

$$\mathcal{S}[\mathcal{F}, x] = \text{absco}\{\{x\} \cup \mathcal{C}\},$$

which is not a b.c.p. Finally, note that the infinitely many vectors of the trajectory $\mathcal{T}[\mathcal{F}, x]$, namely, $A^k Bx$ for $k \geq 0$, which are not proportional to one another, lie on $\mathcal{C}$, which is included in the boundary of $\mathcal{S}[\mathcal{F}, x]$. $\diamond$

Example 6.2. Consider the 2×2 -matrix family $\mathcal{F} = \{A, B\}$, where

$$A = \begin{bmatrix} 3 - \sqrt{5} & \frac{3 - \sqrt{5}}{2} \\ \frac{3 - \sqrt{5}}{2} & 3 - \sqrt{5} \end{bmatrix} \quad \text{and} \quad B = \begin{bmatrix} \frac{3 - \sqrt{5}}{2} & \frac{3 - \sqrt{5}}{2} \\ \frac{3 - \sqrt{5}}{2} & 3 - \sqrt{5} \end{bmatrix}.$$

The matrices A and B have the same eigenvalues 1 and $\frac{7-3\sqrt{5}}{2}$. It can be seen that $\rho(\mathcal{F}) = 1$, that A and B are the only minimal s.m.p.'s of $\mathcal{F}$, and that the unique l.s.m.p.'s of $\mathcal{F}$ are just $A^\infty = \lim_{k \rightarrow \infty} A^k$ and $B^\infty = \lim_{k \rightarrow \infty} B^k$. Therefore, the set of the leading eigenvectors of $\mathcal{F}$ is given by the leading eigenvectors of A and B , namely, $\alpha[\frac{1+\sqrt{5}}{2}, 1]^T$, $\alpha \neq 0$, and $\beta[1, \frac{1+\sqrt{5}}{2}]^T$, $\beta \neq 0$, respectively. This set is finite (modulo scalar nonzero factors) but, as is easy to see, it is not $\mathcal{F}$ -cyclic. As in the previous example, the hypotheses of Theorem 5.18 are violated, even if to a smaller extent. Now set

$$x = \left[\frac{1+\sqrt{5}}{2}, 1 \right]^T,$$

which is the leading eigenvector of the s.m.p. A . It can be checked that all the vectors $B^k x$, $k \geq 1$, are distinct and lie on the segment of $\mathbb{R}^2$ that joins x and

$$B^\infty x = \frac{14+6\sqrt{5}}{15+7\sqrt{5}} \left[1, \frac{1+\sqrt{5}}{2} \right]^T,$$

which is the leading eigenvector of the s.m.p. B , and that all the vectors $AB^k x$, $k \geq 1$, lie inside the b.c.p. $\text{absco}\{x, B^\infty x\}$. We can conclude that

$$\mathcal{S}[\mathcal{F}, x] = \text{absco}\{x, B^\infty x\}.$$

Although $\mathcal{S}[\mathcal{F}, x]$ is a b.c.p., we have seen that the finiteness property of the set $\partial\mathcal{S}[\mathcal{F}, x] \cap \mathcal{T}[\mathcal{F}, x]$ assured by Theorem 5.18 does not hold. The real part of $\mathcal{S}[\mathcal{F}, x]$ is shown in Figure 6.1. $\diamond$

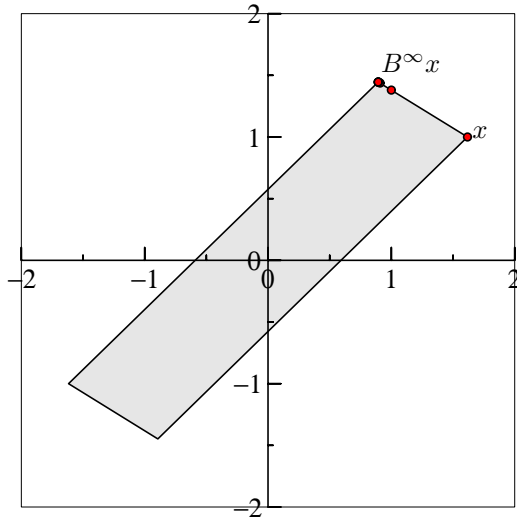


FIG. 6.1. Real part of $\mathcal{S}[\mathcal{F}, x]$ in Example 6.2.

In the next example the considered family is asymptotically simple (even absolutely). Nevertheless, we shall see the substantial potential difference between a trajectory $\mathcal{T}[\mathcal{F}, x]$ exiting from a leading eigenvector and from a vector x which is not such.

Example 6.3. Consider the 3×3 -matrix family $\mathcal{F} = \{A, B\}$, where

$$A = \begin{bmatrix} 1 & 0 & 0 \\ 0 & \frac{1}{2} & 0 \\ 0 & 0 & \frac{1}{4} \end{bmatrix} \quad \text{and} \quad B = \begin{bmatrix} \frac{1}{2} & 0 & 0 \\ \frac{1}{2} & 0 & 0 \\ \frac{1}{2} & 0 & 0 \end{bmatrix}.$$

Since both A and B are lower triangular, it is immediately seen that $\rho(\mathcal{F}) = 1$ and that A is the unique minimal s.m.p., which has only one leading eigenvector (modulo scalar nonzero factors), related to the eigenvalue $\lambda = 1$, given by $\alpha[1, 0, 0]^T$, $\alpha \neq 0$. Therefore, $\mathcal{F}$ is absolutely asymptotically simple. In view of Theorem 5.18, if we choose

$$x = [1, 0, 0]^T,$$

then the corresponding set $\mathcal{S}[\mathcal{F}, x]$ is a b.c.p. generated by a finite number of points of the trajectory $\mathcal{T}[\mathcal{F}, x]$. In order to verify this fact, we observe that

$$Bx = \frac{1}{2}[1, 1, 1]^T, \quad B[0, 1, 0]^T = B[0, 0, 1]^T = [0, 0, 0]^T,$$

$$A^k[1, 1, 1]^T = \left[1, \frac{1}{2^k}, \frac{1}{4^k}\right]^T, \quad k \geq 1.$$

Consequently, it easily turns out that

$$\mathcal{S}[\mathcal{F}, x] = \text{absco}\{x, Bx, ABx, A^2Bx\}$$

and that all the other vectors of the trajectory $\mathcal{T}[\mathcal{F}, x]$ lie inside the above b.c.p.

Now choose instead

$$x = [1, 1, 1]^T,$$

which is not a leading eigenvector. This time $\mathcal{S}[\mathcal{F}, x]$ is not a b.c.p. In fact, the infinitely many vectors $A^k x$, $k \geq 1$, which all belong to $\partial\mathcal{S}[\mathcal{F}, x]$ and accumulate at $[1, 0, 0]^T$, are such that their projections into the plane (x_2, x_3) of the last two variables lie on the parabola whose equation is $x_3 = x_2^2$. $\diamond$

We remark that, although necessary for the general validity of the finiteness property of the set $\partial\mathcal{S}[\hat{\mathcal{F}}, x] \cap \mathcal{T}[\hat{\mathcal{F}}, x]$, the special hypotheses of Theorem 5.18 seem not to be necessary for the simple existence of an extremal complex polytope (see the CPE conjecture). Indeed, even for the family of Example 6.1, it can be easily verified that a b.c.p. unit ball is given by

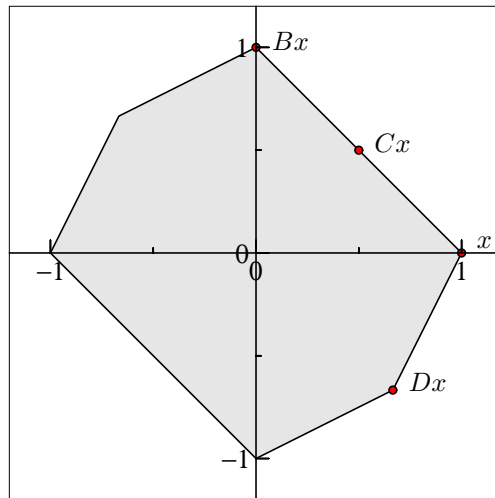
$$\mathcal{P} = \text{absco}\{[1, i]^T, [1, -i]^T\}.$$

We conclude the paper with the following example which shows that, for a family which is asymptotically simple but not absolutely, not all the leading eigenvectors are necessarily vertices of the b.c.p. $\mathcal{S}[\hat{\mathcal{F}}, x]$.

Example 6.4. Consider the 2×2 -matrix family $\mathcal{F} = \{A, B, C, D\}$, where

$$A = \begin{bmatrix} 1 & 1 \\ 0 & 0 \end{bmatrix}, \quad B = \begin{bmatrix} 0 & 0 \\ 1 & 1 \end{bmatrix}, \quad C = \begin{bmatrix} \frac{1}{2} & \frac{1}{2} \\ \frac{1}{2} & \frac{1}{2} \end{bmatrix}, \quad D = \begin{bmatrix} \frac{2}{3} & 0 \\ -\frac{2}{3} & 0 \end{bmatrix}.$$

It turns out that $XY = X$ and $XD = O$, the zero matrix, for all $X, Y \in \{A, B, C\}$. Therefore, since $\rho(A) = \rho(B) = \rho(C) = 1$ and $\rho(D) = \frac{2}{3}$, we have that $\rho(\mathcal{F}) = 1$ and

FIG. 6.2. Real part of $\mathcal{S}[\mathcal{F}, x]$ in Example 6.4.

that A, B, C are all minimal s.m.p.'s of $\mathcal{F}$. Moreover, the leading eigenvectors of $\mathcal{F}$ are just the three leading eigenvectors of A, B , and C , that is, $\alpha[1, 0]^T$, $\beta[0, 1]^T$, and $\gamma[\frac{1}{2}, \frac{1}{2}]^T$, $\alpha, \beta, \gamma \neq 0$, respectively, which are related to the common eigenvalue $\lambda = 1$. This set of leading eigenvectors is clearly finite and, as is immediately seen, also $\mathcal{F}$ -cyclic. Therefore, $\mathcal{F}$ is asymptotically simple, but not absolutely. If we choose

$$x = [1, 0]^T,$$

which is the leading eigenvector of A , it easily turns out that

$$\mathcal{S}[\mathcal{F}, x] = \text{absco}\{x, Bx, Dx\},$$

where $Bx = [0, 1]^T$ is the leading eigenvector of B and $Dx = [\frac{2}{3}, -\frac{2}{3}]^T$ is not a leading eigenvector of $\mathcal{F}$. Note that the leading eigenvector of C belonging to $\partial\mathcal{S}[\hat{\mathcal{F}}, x]$, namely, $Cx = [\frac{1}{2}, \frac{1}{2}]^T$, satisfies the convexity relationship $Cx = \frac{1}{2}x + \frac{1}{2}Bx$ and, hence, is not a vertex of the b.c.p. $\mathcal{S}[\mathcal{F}, x]$. The intersection of $\mathcal{S}[\mathcal{F}, x]$ with $\mathbb{R}^2$ is displayed in Figure 6.2. $\diamond$

REFERENCES

- [Bara88] N. E. BARABANOV, *Lyapunov indicator of discrete inclusions. I–III*, Automat. Remote Control, 49 (1988), pp. 152–157; 283–287; 558–565.
- [BerWan] M. A. BERGER AND Y. WANG, *Bounded semigroups of matrices*, Linear Algebra Appl., 166 (1992), pp. 21–27.
- [Blondel1] V. D. BLONDEL, J. THEYS, AND A. A. VLADIMIROV, *An elementary counterexample to the finiteness conjecture*, SIAM J. Matrix Anal. Appl., 24 (2003), pp. 963–970.
- [BlonNest05] V. D. BLONDEL AND Y. NESTEROV, *Computationally efficient approximations of the joint spectral radius*, SIAM J. Matrix Anal. Appl., 27 (2005), pp. 256–272.
- [BouMer] T. BOUSCH AND J. MAIRESSE, *Asymptotic height optimization for topical IFS, Tetris heaps and the finiteness conjecture*, J. Amer. Math. Soc., 15 (2002), pp. 77–111.
- [DauLag1] I. DAUBECHIES AND J. C. LAGARIAS, *Sets of matrices all infinite products of which converge*, Linear Algebra Appl., 161 (1992), pp. 227–263.
- [DauLag2] I. DAUBECHIES AND J. C. LAGARIAS, *Corrigendum/addendum to: Sets of matrices all infinite products of which converge*, Linear Algebra Appl., 327 (2001), pp. 69–83.

- [Els] L. ELSNER, *The generalized spectral-radius theorem: An analytic-geometric proof*, Linear Algebra Appl., 220 (1995), pp. 151–159.
- [GugZen1] N. GUGLIELMI AND M. ZENNARO, *On the asymptotic properties of a family of matrices*, Linear Algebra Appl., 322 (2001), pp. 169–192.
- [GugZen2] N. GUGLIELMI AND M. ZENNARO, *On the zero-stability of variable stepsize multistep methods: The spectral radius approach*, Numer. Math., 88 (2001), pp. 445–458.
- [GugZen3] N. GUGLIELMI AND M. ZENNARO, *On the limit products of a family of matrices*, Linear Algebra Appl., 362 (2003), pp. 11–27.
- [GugZen4] N. GUGLIELMI AND M. ZENNARO, *Balanced Complex Polytopes and Related Vector and Matrix Norms*, Preprint, 2005.
- [HarWri] G. H. HARDY AND E. M. WRIGHT, *An Introduction to the Theory of Numbers*, Clarendon Press, Oxford, UK, 1979.
- [Heu] H. G. HEUSER, *Functional Analysis*, John Wiley & Sons, New York, 1982.
- [Kozy90] V. S. KOZYAKIN, *Algebraic unsolvability of a problem on the absolute stability of desynchronized systems*, Automat. Remote Control, 51 (1990), pp. 754–759.
- [LagWan] J. C. LAGARIAS AND Y. WANG, *The finiteness conjecture for the generalized spectral radius of a set of matrices*, Linear Algebra Appl., 214 (1995), pp. 17–42.
- [Mae1] M. MAESUMI, *Optimum unit ball for joint spectral radius: An example from four-coefficient MRA*, in Approximation Theory VIII: Wavelets and Multilevel Approximation, Vol. 2, C. K. Chui and L. L. Schumaker, eds., World Scientific, River Edge, NJ, 1995, pp. 267–274.
- [Mae2] M. MAESUMI, *Calculating joint spectral radius of matrices and Hölder exponent of wavelets*, in Approximation Theory IX, Vol. 2, C. K. Chui and L. L. Schumaker, eds., Vanderbilt University Press, Nashville, TN, 1998, pp. 205–212.
- [MoisOrli01] B. E. MOISION, A. ORLITSKY, AND P. H. SIEGEL, *On codes that avoid specified differences*, IEEE Trans. Inform. Theory, 47 (2001), pp. 433–442.
- [RR00] H. RADJAVI AND P. ROSENTHAL, *Simultaneous Triangularization*, Universitext, Springer-Verlag, New York, 2000.
- [RotStr] G. C. ROTA AND G. STRANG, *A note on the joint spectral radius*, Indag. Math., 22 (1960), pp. 379–381.
- [Shi] M. H. SHIH, *Simultaneous Schur stability*, Linear Algebra Appl., 287 (1999), pp. 323–336.
- [ShWuPa] M. H. SHIH, J. W. WU, AND C. T. PANG, *Asymptotic stability and generalized Gelfand spectral radius formula*, Linear Algebra Appl., 252 (1997), pp. 61–70.
- [TsitBlon97] J. N. TSITSIKLIS AND V. BLONDEL, *The Lyapunov exponent and joint spectral radius of pairs of matrices are hard—when not impossible—to compute and to approximate*, Math. Control Signals Systems, 10 (1997), pp. 31–40.
- [Wirt98] F. WIRTH, *On the calculation of time-varying stability radii*, Internat. J. Robust Nonlinear Control, 8 (1998), pp. 1043–1058.
- [Wirt02] F. WIRTH, *The generalized spectral radius and extremal norms*, Linear Algebra Appl., 342 (2002), pp. 17–40.
- [Zie] G. M. ZIEGLER, *Lectures on Polytopes*, Springer-Verlag, New York, 1995.

AN IMPROVED CONVERGENCE BOUND FOR AGGREGATION-BASED DOMAIN DECOMPOSITION PRECONDITIONERS*

MARZIO SALA[†], JOHN N. SHADID[‡], AND RAY S. TUMINARO[§]

Abstract. In this paper we present a two-level overlapping domain decomposition preconditioner for the finite-element discretization of elliptic problems in two and three dimensions. The computational domain is partitioned into overlapping subdomains, and a coarse space correction, based on aggregation techniques, is added. Our definition of the coarse space does not require the introduction of a coarse grid. We consider a set of assumptions on the coarse basis functions to bound the condition number of the resulting preconditioned system. These assumptions involve only geometrical quantities associated with the aggregates and the subdomains. We prove that the condition number using the two-level additive Schwarz preconditioner is $\mathcal{O}(H/\delta + H_0/\delta)$, where H and H_0 are the diameters of the subdomains and the aggregates, respectively, and δ is the overlap among the subdomains and the aggregates. This extends the bounds presented in [C. Lasser and A. Toselli, *Convergence of some two-level overlapping domain decomposition preconditioners with smoothed aggregation coarse spaces*, in Recent Developments in Domain Decomposition Methods, Lecture Notes in Comput. Sci. Engrg. 23, L. Pavarino and A. Toselli, eds., Springer-Verlag, Berlin, 2002, pp. 95–117; M. Sala, *Domain Decomposition Preconditioners: Theoretical Properties, Application to the Compressible Euler Equations, Parallel Aspects*, Ph.D. thesis, Ecole Polytechnique Fédérale de Lausanne, Lausanne, Switzerland, 2003; M. Sala, *Math. Model. Numer. Anal.*, 38 (2004), pp. 765–780]. Numerical experiments on a model problem are reported to illustrate the performance of the proposed preconditioner.

Key words. domain decomposition methods, aggregation preconditioners, iterative methods

AMS subject classifications. 65F10, 65N55, 65Y05

DOI. 10.1137/040617455

1. Introduction. In this paper we consider the iterative solution of the linear system arising from the discretization of the scalar Poisson problem

$$(1.1) \quad \begin{cases} -\Delta u &= f & \text{in } \Omega, \\ u &= 0 & \text{on } \partial\Omega, \end{cases}$$

where Ω is a bounded polyhedral domain in $\mathbb{R}^d$, $d \in \{2, 3\}$. The finite-element discretization leads to a large sparse linear system, which is typically solved using the conjugate gradient method [8]. As the condition number of the associated matrix is $\mathcal{O}(h^{-2})$, where h is the mesh size of the triangulation, a preconditioner needs to be employed [15]. In this paper we explore preconditioners based on domain decomposition techniques. These methods share many features and often compete with multigrid

*Received by the editors October 21, 2004; accepted for publication (in revised form) by A. J. Wathen June 21, 2005; published electronically December 30, 2005. This work was partially supported by the DOE NNSA ASC program and the DOE Office of Science MICS program at Sandia National Laboratory. Sandia is a multiprogram laboratory operated by Sandia Corporation, a Lockheed Martin Company, for the United States Department of Energy's National Nuclear Security Administration under contract DE-AC04-94AL85000.

<http://www.siam.org/journals/simax/27-3/61745.html>

[†]Sandia National Laboratories, P. O. Box 5800, MS 1110, Albuquerque, NM 87185 (msala@sandia.gov).

[‡]Sandia National Laboratories, P. O. Box 5800, MS 0316, Albuquerque, NM 87185 (jnshadi@sandia.gov).

[§]Sandia National Laboratories, P. O. Box 969, MS 9217, Livermore, CA 94551 (tuminaro@ca.sandia.gov).

methods [2, 20]. A domain decomposition preconditioner is naturally suited for parallel computations, as the general principle is to divide the problem into subproblems or subdomains and assign each to a different processor. The key idea associated with obtaining scalable convergence is the projection of the operator onto an additional coarse space.

Here, we consider Schwarz procedures. In its simplest form, the Schwarz preconditioner is defined by subdividing the computational domain Ω into M overlapping subdomains Ω_i , then solving local Dirichlet-type problems on each Ω_i . It is well known that this procedure, referred to as the one-level Schwarz method, is nonscalable. In fact, the information exchange among the subdomains is only through the overlapping regions, while for elliptic problems the domain of dependence is global. A scalable method may be recovered by the addition of a suitable coarse operator; i.e., a projection of the original problem is solved on an additional coarse space. This auxiliary problem furnishes a way to efficiently transfer information to all subdomains in each step. As in a multigrid setting, the coarse problem is used to correct the “smooth” (low-frequency) part of the error, whereas the local preconditioner is used to relax the “high-frequency” ones.

A typical approach to defining the coarse operator is to resort to finite-element approximation on a coarse triangulation. For structured meshes of complex geometric domain, it is a relatively easy task, while for unstructured meshes this may be problematic or computationally expensive, especially for 3D (three-dimensional) computations. A possible way to overcome the difficulties induced by the definition of a coarse triangulation, without losing the power of two-level methods, is to resort to aggregation procedures. This is done as follows. First, the PDE problem is discretized on the fine grid. Then, the fine grid nodes are grouped in contiguous sets, called *aggregates*. The basis functions of the coarse space are formed by summing up the finite-element basis functions of the nodes in each aggregate.

Aggregation procedures have been presented in the literature by various authors. An aggregation technique was first introduced in 1951 by Leontief [12, Chapter 9] and then extensively used in economics; see [13] and the references therein. For multigrid applications, smoothed aggregation techniques are used, for example, in [1, 22], where investigations of the smoothed aggregation properties have been reported. In the framework of domain decomposition methods, the focus is mainly on two-level methods. Results are presented in [14] for the shallow water equations and 2D potential flows, in [6] for 3D potential flow computations, in [9] for groundwater flows, in [10] for multiphase flows, in [5] for discontinuous Galerkin approximation of advection-diffusion problems, and in [18, 19] for the 3D compressible Euler equations on unstructured tetrahedral grids.

For symmetric coercive problems, it is possible to estimate the condition number of the preconditioned operator using the abstract Schwarz theory of Dryja and Widlund [4]. The contribution of our paper is the presentation of a convergence bound that depends linearly on $(H + H_0)/\delta$, where H is the size of the subdomains, H_0 is the size of the aggregates, and δ is the amount of overlap among the subdomains and the aggregates. This extends the bound presented in [11], where the authors assume $H = H_0$ (or, more precisely, that aggregates and subdomains coincide).

This paper is organized as follows. In section 2 we outline the model problem and its discretization. Section 3 presents the one-level and two-level Schwarz preconditioners and introduces the aggregation technique. Section 4 presents the improved convergence bound for aggregation. Section 5 reports numerical results for 2D and 3D domains. In section 6 conclusions are presented.

2. Model problem. Let us consider the weak form of the Poisson problem (1.1), whose finite-element approximation reads

$$(2.1) \quad \begin{aligned} &\text{Find } u_h \in V_h \text{ such that} \\ &a(u_h, v_h) = f(v_h) \quad \forall v_h \in V_h, \end{aligned}$$

with $a(u_h, v_h) = \int_{\Omega} \nabla u_h \cdot \nabla v_h \, d\Omega$, $f(v_h) = \int_{\Omega} f v_h \, d\Omega$. By assumption, $f \in L_2(\Omega)$, and $V_h \subset H_0^1(\Omega)$ is the space of linear finite-element functions defined on a conforming quasi-uniform triangulation $\mathcal{T}_h$ of Ω . Existence of a unique solution of (2.1) is shown, for instance, in [15].

The algebraic formulation of problem (2.1) reads

$$(2.2) \quad A\mathbf{u} = \mathbf{f},$$

where $A \in \mathbb{R}^{n \times n}$ is a symmetric positive definite matrix, $\mathbf{u} \in \mathbb{R}^n$ the solution vector, and $\mathbf{f} \in \mathbb{R}^n$ the discretization of the force term f .

3. Schwarz preconditioners. The one-level Schwarz preconditioner for the iterative solution of (2.2) can be defined as follows. First, the computational domain Ω is partitioned into M overlapping subdomains Ω_i . On each Ω_i we have, by construction, $\text{diam}(\Omega_i) = \mathcal{O}(H)$. Then, the preconditioner reads

$$(3.1) \quad B' = \sum_{i=1}^M B_i,$$

where $B_i = R_i^T A_i^{-1} R_i$, $A_i = R_i A R_i^T$, and R_i^T is the discrete representation of the interpolation from the subspace V_i (of functions of V_h whose support is contained in Ω_i) to the space V_h .

Preconditioner (3.1) is not scalable and the resulting condition number can be bounded by

$$(3.2) \quad \kappa(B'A) \leq C \frac{1}{H\delta}$$

(see, for instance, [3]). Here and in the following, C (with or without subscripts) denotes a constant independent of H , h , the overlap among the subdomains δ , and H_0 .

To avoid the term $1/H$ in (3.2), a further correction term B_0 can be added. The resulting preconditioner reads

$$(3.3) \quad B'' = B_0 + B' = \sum_{i=0}^M B_i,$$

where $B_0 = R_0^T A_0^{-1} R_0$ is a projection on a space V_0 , which is “coarse” in the sense that it contains a limited number of degrees of freedom, to make the “exact” inversion of A_0 computationally acceptable—that is, $n_0 = \dim(V_0) \ll n$. Preconditioner (3.3) is usually referred to as a two-level additive Schwarz preconditioner.

The properties (and the parallel performance) of two-level Schwarz preconditioners will strongly depend on the definition of the coarse space V_0 . When possible, the coarse space V_0 may be itself embedded into V_h . Often, the coarse operator represents the discretization of the continuous problem on a very coarse mesh. In this case, the following result holds.

LEMMA 3.1. *Consider the additive two-level overlapping Schwarz method, where the overlap is uniform of width $\mathcal{O}(\delta)$, the coarse grid space V_0 corresponds to the finite-element functions on elements of $\mathcal{O}(H)$, and $V_0 \subset V_h$. Then*

$$(3.4) \quad \kappa(B''A) \leq C \left(1 + \frac{H}{\delta}\right),$$

where C is a constant independent of h , H , and δ .

Proof. For the proof, see [20]. $\square$

Alternatively, V_0 can be defined by resorting to aggregation procedures. This is done as follows. First, the differential problem is discretized on the fine grid to form the matrix A . Then, the fine grid nodes are grouped into nonoverlapping sets of contiguous nodes, called *aggregates*. This decomposition is usually obtained using a graph partitioning algorithm. The number of aggregates n_0 represents the dimension of the coarse space, since each aggregate will be given a unique coarse grid function. Using these aggregates, a simple procedure can be followed to define a restriction operator R_0^* (in section 5 we will report one possible strategy to define R_0^*). This operator is usually referred to as a tentative restriction. Then, a smoother is applied to R_0^* in order to reduce the energy of the basis functions and increase their overlap. This defines the final restriction operator R_0 . This second step consists of applying a smoother S to produce the final operator R_0 via $R_0 = SR_0^*$.

Possible choices for S are, for example, a simple Richardson smoother [5, 11] or a recursive Richardson smoother [1, 23]. This construction of R_0 is sometimes referred to as *smoothed aggregation* (S). If $S = I$ (i.e., $R_0 = R_0^*$), the procedure is usually referred to as *nonsmoothed aggregation*. Both R_0^* and S are selected so that $R_0 \in \mathbb{R}^{n_0 \times n}$.

The linear operator R_0 defines the basis function of the coarse space as follows. Let φ_j be a generic basis function of V_h ; then, a generic coarse space basis function Φ_i can be written as

$$\Phi_i(x) = \sum_j R_0(i, j) \varphi_j(x),$$

and the final aggregates are defined as $\Theta_i = \text{supp}(\Phi_i)$ (where $\text{supp}(\Phi_i)$ refers to the support of Φ_i). We define $H_0 = \max_{j=1, \dots, n_0} \text{diam}(\Theta_j)$, and δ_0 is the amount of overlap among the aggregates.

Remark 1. $V_0 \subset V_h$, since V_0 is defined such that each element of V_0 is actually composed of a linear combination of elements of V_h .

Once R_0 is computed, the coarse matrix is defined as $A_0 = R_0 A R_0^T$. More details on two-level methods based on aggregation can be found in [16, 17] and the references therein. We note that, although this paper focuses on two-level preconditioners, similar ideas can be used to define multilevel preconditioners; see, for instance, [22, 21] and the references therein.

4. Convergence estimate for aggregation. Let $\{\Theta_i, i = 1, \dots, n_0\}$ be a set of aggregates, defined as outlined in section 3, and $\{\Phi_i, i = 1, \dots, n_0\}$ a set of basis functions that satisfy $\text{supp}(\Phi_i) = \Theta_i$.

In order to prove our theoretical estimates, we need to introduce aggregates corresponding to the boundary nodes. Those aggregates and their corresponding basis functions are needed in the proof to define a partition of the unity over the entire computation domain. These additional aggregates do not need to be implemented in

practice, since Dirichlet boundary nodes can be eliminated from linear system (2.2) before its solution.

Let ω be the set of aggregates corresponding to internal nodes, and γ be the set of aggregates constructed by aggregating boundary nodes. For each $i \in \gamma$, we introduce an additional function $\Phi_i \in V_h$ so that $\sum_{i \in \omega \cup \gamma} \Phi_i(x) = 1 \ \forall x \in \bar{\Omega}$.

The following properties will be used to characterize the aggregates and the basis functions of the coarse space.

PROPERTY 1 (partition). *There are three constants C_1 , C_2 , and C_3 such that, for each aggregate Θ_i , $i \in \omega \cup \gamma$, we have that*

- a. $\text{diam}(\Theta_i) \leq C_1 H_0$;
- b. *the Lebesgue measure $|\Theta_i|$ of Θ_i satisfies $|\Theta_i| \leq C_2 H_0^d$ if $i \in \omega$, and $|\Theta_i| \leq C_3 H_0^{d-1} \delta_0$ if $i \in \gamma$;*
- c. *the overlap among the aggregates is of order δ_0 .*

Property 1 states that the aggregates have diameter of comparable size H_0 and are shape regular. Property 2 below requires a certain regularity on the coarse space basis function Φ_i .

PROPERTY 2 (coarse space). *There exists a constant $C > 0$ such that the basis functions $\{\Phi_i\}$, $i \in \omega \cup \gamma$, of the coarse space satisfy*

- a. $\|\Phi_i\|_{L^\infty(\Omega)}^2 \leq C$;
- b. $\|\nabla \Phi_i\|_{L^\infty(\Omega)}^2 \leq C/\delta_0^2$;
- c. $\sum_{i \in \omega \cup \gamma} \Phi_i(x) = 1 \ \forall x \in \bar{\Omega}$.

Properties 2.a and 2.b require a certain pointwise regularity, while Property 2.c states that the $\{\Phi_i\}$ forms a partition of the unity on the entire domain (verified by construction).

From Property 2 it easily follows that

$$(4.1) \quad |\Phi_i|_{H^1(\Omega)}^2 \leq C \frac{H_0^{d-1}}{\delta_0},$$

$$(4.2) \quad \|\Phi_i\|_{L^2(\Omega)}^2 \leq C H_0^d.$$

Remark 2. A nonnegative function Φ_i , which is constant in the interior of Θ_i and decreases to zero with $\|\nabla \Phi_i\|_{L^\infty(\Omega)} \leq 1/\delta_0$ in a layer of width δ_0 , satisfies Property 2. In particular, the nonsmoothed case defined by (5.1) defines a set of functions $\{\Phi_i\}$ which satisfies Property 2 with $\delta_0 = h$.

Also, let us define the operator $Q_0 : V_h \rightarrow V_0$ by

$$(4.3) \quad \begin{aligned} Q_0 u &= \sum_{i \in \omega \cup \gamma} \alpha_i \Phi_i, \\ \alpha_i &= \begin{cases} \frac{1}{|\Theta_i|} \int_{\Theta_i} u(\mathbf{x}) d\Omega, & i \in \omega, \\ 0, & i \in \gamma. \end{cases} \end{aligned}$$

LEMMA 4.1. *If Properties 1 and 2 are satisfied, then there exists a constant $C > 0$ independent of H_0, δ_0 , and h such that $\forall u_h \in V_h$,*

- a. $\|u - Q_0 u\|_{L^2(\Omega)}^2 \leq C H_0^2 |u|_{H^1(\Omega)}^2$;
- b. $|Q_0 u|_{H^1(\Omega)}^2 \leq C \frac{H_0}{\delta_0} |u|_{H^1(\Omega)}^2$.

Proof. For the proof, see [16, 17]. $\square$

In the following we will use Lemma 4.2.

LEMMA 4.2. *Let $\Omega_i \subset \mathbb{R}^d$, $d = 2, 3$, be a rectangle of diameter H , and let Γ_{δ_i} be*

a strip along its boundary of width $\delta > 0$. Then, for any function $u \in H^1(\Omega_i)$,

$$(4.4) \quad \|u\|_{L^2(\Gamma_{\delta_i})}^2 \leq C\delta^2 \left[\left(1 + \frac{H}{\delta}\right) |u|_{H^1(\Omega_i)}^2 + \frac{1}{H\delta} \|u\|_{L^2(\Omega_i)}^2 \right].$$

Proof. For the proof, see, for instance, [4]. $\square$

For completeness, we also report the convergence bounds presented in [11] and [17].

THEOREM 4.3. *Let Properties 1 (with $H_0 = H$ and $\delta_0 = \delta$) and 2 hold. Then, there exists a constant $C > 0$ such that*

$$\kappa(B''A) \leq C \left(1 + \frac{H}{\delta}\right).$$

Proof. For the proof, see [11, Lemma 7]. $\square$

THEOREM 4.4. *Let Properties 1 and 2 hold. Then there exists $C > 0$ such that*

$$(4.5) \quad \kappa(B''A) \leq C \left(1 + \frac{H_0}{\delta_0}\right) \left(1 + \frac{H}{\delta}\right).$$

Proof. For the proof, see [17]. $\square$

Theorem 4.4 states that, for a given decomposition into subdomains, the condition number can be reduced by increasing the number of the aggregates. However, Theorem 4.4 is not optimal, as in the case $H = H_0$, $\delta = \delta_0$; the bound in the condition number reads

$$(4.6) \quad \kappa(B''A) \leq C \left(1 + \frac{H}{\delta}\right)^n,$$

with $n = 2$, while Theorem 4.3 states that (4.6) holds with $n = 1$. The following theorem, which is the main result of this paper, addresses this problem.

THEOREM 4.5 (aggregation coarse space). *Let Properties 1 and 2 hold. Then, for the additive two-level overlapping Schwarz method, when the overlap is uniform of width $\mathcal{O}(\delta)$, $\delta_0 = \delta$, each aggregate is completely contained in only one subdomain, $H \geq H_0$, and $V_0 = \text{span}\{\Phi_i\}$, there exist $C > 0, C_0 > 0$ independent of h, δ, H, H_0 such that*

$$(4.7) \quad \kappa(B''A) \leq C \frac{H}{\delta} \left(1 + C_0 \frac{H_0}{H}\right).$$

Proof. Define $\mathcal{I}_h : V \rightarrow V_h$ such that

$$\mathcal{I}_h(u) = \sum_{i=1}^n u(x_i) \varphi_i,$$

where φ_i is a generic basis function of the mesh, and the x_i 's are the mesh nodal points.

Let $\omega_i = \{j \in \omega \cup \gamma \mid \text{supp}(\Phi_j) \subseteq \Omega_i\}$. Let $V_i = H_0^1(\Omega_i) \cap V_h$, $i = 1, \dots, M$. A function $u \in V_h$ can be decomposed as $u = u_0 + \sum_{i=1}^M u_i$, with

$$u_0 = Q_0 u, \quad u_i = \sum_{j \in \omega_i} \mathcal{I}_h(\Phi_j(u - \alpha_j)), \quad i = 1, \dots, M,$$

with Q_0 defined as in (4.3), where

$$u_0 = \sum_{i \in \omega \cup \gamma} \Phi_i \alpha_i = \sum_{i=1}^M \sum_{j \in \omega_i} \Phi_j \alpha_j + \sum_{i \in \gamma} \Phi_i \alpha_i,$$

corresponding to the n_0 aggregates plus the additional n_γ functions on the boundary. This is a proper decomposition, because

$$\begin{aligned} \sum_{i=0}^M u_i &= \sum_{j \in \omega \cup \gamma} \Phi_j \alpha_j + \sum_{i=1}^M \sum_{j \in \omega_i} \mathcal{I}_h (\Phi_j u - \Phi_j \alpha_j) \\ &= \sum_{i=1}^M \sum_{j \in \omega_i} \{\Phi_j \alpha_j + \Phi_j u - \Phi_j \alpha_j\} \\ &= \sum_j \Phi_j u = u, \end{aligned}$$

since $\sum_i \Phi_i(x) = 1 \ \forall x \in \bar{\Omega}$, owing to Property 2 and $u_i \in V_i$. Define $\Psi_i = \sum_{j \in \omega_i} \Phi_j$. (Note that Ψ_i defines a partition of unity.) Using the inverse inequality, we get, over a generic element $K \subset \Omega_i$,

$$\begin{aligned} |u_i|_{H^1(K)}^2 &= \left| \sum_{j \in \omega_i} \mathcal{I}_h (\Phi_j u - \Phi_j \alpha_j) \right|_{H^1(K)}^2 \\ &= \left| \sum_{j \in \omega_i} \mathcal{I}_h \Phi_j u - \sum_{j \in \omega_i} \mathcal{I}_h \Phi_j \alpha_j \right|_{H^1(K)}^2 \\ (4.8) \quad &\leq |\Psi_i u|_{H^1(K)}^2 + \left| \sum_{j \in \omega_i} \Phi_j \alpha_j \right|_{H^1(K)}^2. \end{aligned}$$

Let $\bar{\Psi}_{K,i}$ indicate the average of Ψ_i over the element K . With this notation, and using Property 2 and the fact that $\delta = \mathcal{O}(h)$, the first term on the right-hand side of (4.8) can be bounded as follows:

$$\begin{aligned} (4.9) \quad |\Psi_i u|_{H^1(K)}^2 &= |\bar{\Psi}_{K,i} u + (\Psi_i - \bar{\Psi}_{K,i}) u|_{H^1(K)}^2 \\ &\leq 2 |\bar{\Psi}_{K,i} u|_{H^1(K)}^2 + 2 |(\Psi_i - \bar{\Psi}_{K,i}) u|_{H^1(K)}^2 \\ &\leq 2 |u|_{H^1(K)}^2 + C \delta^{-2} \|(\Psi_i - \bar{\Psi}_{K,i}) u\|_{L^2(K)}^2 \\ (4.10) \quad &\leq 2 |u|_{H^1(K)}^2 + C \delta^{-2} \|\Psi_i - \bar{\Psi}_{K,i}\|_{L^\infty(K)}^2 \|u\|_{L^2(K)}^2. \end{aligned}$$

The second term of (4.10) has a nonzero contribution only near the boundaries of Ω_i , given by summing over all the elements of Ω_i and using Lemma 4.2,

$$\begin{aligned} |\Psi_i u|_{H^1(\Omega_i)}^2 &\leq C |u|_{H^1(\Omega_i)}^2 + C \delta^{-2} |u|_{L^2(\Gamma_{\delta_i})}^2 \\ &= C |u|_{H^1(\Omega_i)}^2 + C \left(1 + \frac{H}{\delta}\right) |u|_{H^1(\Omega_i)}^2 \\ &\leq C \left(1 + \frac{H}{\delta}\right) |u|_{H^1(\Omega_i)}^2. \end{aligned}$$

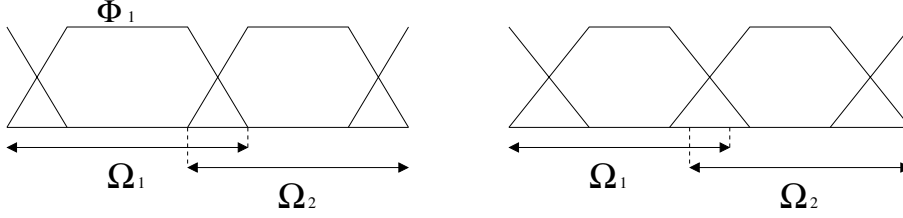


FIG. 1. Example of coarse basis functions that satisfy (left) and do not satisfy (right) the hypothesis of Theorem 4.5.

By manipulating the second term on the right-hand side of (4.8), we obtain

$$\sum_{i=1}^M \sum_{K \in \Omega_i} \left| \sum_{j \in \omega_i} \Phi_j \alpha_j \right|_{H^1(K)}^2 \leq C |Q_0 u|_{H^1(\Omega)}^2.$$

Finally, using Lemma 4.1.b and $\delta_0 = \delta$, we obtain

$$\begin{aligned} \sum_{i=0}^M |u_i|_{H^1(\Omega)}^2 &= |u_0|_{H^1(\Omega)}^2 + \sum_{i=1}^M |u_i|_{H^1(\Omega)}^2 \\ &\leq C_0 \frac{H_0}{\delta} |u|_{H^1(\Omega)}^2 + C \left(1 + \frac{H}{\delta}\right) |u|_{H^1(\Omega)}^2 \\ &\leq \left(C_0 \frac{H_0}{\delta} + C \frac{H}{\delta}\right) |u|_{H^1(\Omega)}^2. \end{aligned}$$

Inequality (4.7) follows from the standard theory for abstract Schwarz methods (see, for instance, [20] and references therein). $\square$

Figure 1 shows an example of coarse basis functions that satisfy (on the left) and do not satisfy (on the right) the hypothesis of Theorem 4.5.

Remark 3. Theorem 4.5 states that, for a given decomposition into subdomains, the condition number can be reduced by decreasing the diameter of the aggregates. This is of particular advantage in situations where the decomposition into subdomains (and henceforth H) is fixed. This may occur, for instance, if the Schwarz preconditioner is used in a parallel environment, and each subdomain is assigned to a different processor.

5. Numerical results. We report in this section some numerical results for our model problem (2.1), with $\Omega = (0, 1) \times (0, 1)$ and $\Omega = (0, 1) \times (0, 1) \times (0, 1)$. The mesh is constructed by dividing Ω into n^2 equal squares, subdivided into triangles, or n^3 cubes, subdivided into tetrahedra. Thus, we obtain a triangulation with $h = \frac{1}{n}$. As regards the decomposition into subdomains, we consider overlapping squares (or cubes) Ω_i of area H^d , $d = 2, 3$. We use linear finite elements and solve the linear system (2.2) by the preconditioned conjugate gradient method.

As described in section 3, the first step of aggregation requires a simple procedure to define the restriction operator R_0^* in such a way that each aggregate will eventually correspond to a “degree of freedom” of the coarse operator. To define the aggregates, we decompose the set of nodes into nonoverlapping sets, ϑ_j^* , $j = 1, \dots, n_0$. The $n_0 \times n$ Boolean matrix R_0^* is constructed in such a way that each row corresponds to an

aggregate and each column to a node, as follows:

$$(5.1) \quad R_0^*(i, j) = \begin{cases} 1 & \text{if } j \in \vartheta_i^*, \\ 0 & \text{if } j \notin \vartheta_i^*. \end{cases}$$

R_0^* can be viewed as a simple grid transfer operator corresponding to piecewise constant interpolation; see Remark 2.

Remark 4. To define (5.1), we construct aggregates with minimal overlap $\delta_0 = h$. Alternatively, nonoverlapping aggregates can be defined as nonoverlapping partitions of Ω ; see [11].

For nonsmoothed aggregation, the aggregates (whose shape is approximately a square for $d = 2$ and a cube for $d = 3$) are defined as described by (5.1). The corresponding preconditioner is indicated by B_{nsa}'' . For smoothed aggregation, the following Richardson smoother is employed:

$$(5.2) \quad S = \left(I - \frac{\varpi}{\lambda_M} A \right),$$

where λ_M is the maximum eigenvalue of A (estimated using 10 steps of a conjugate gradient method), and $\varpi = 4/3$ is the damping parameter. The corresponding preconditioner is indicated by B_{sa}'' .

Tables 1–4 report the estimated condition number for the preconditioned system; see, for instance, [7]. The stopping criteria is 10^{-12} on the relative residual.

A comparison of the condition numbers obtained using a geometric coarse grid and aggregation is reported in [17], where it is shown that, for the considered 2D model problem, aggregation procedures result in condition numbers about four times larger than numbers resulting from using a geometric coarse grid. In this paper, we present results concerning aggregation schemes only. Let $\chi = H_0/H$ indicate the ratio between the diameter of the aggregates and the diameter of the subdomains. $\chi = 1$ means that all the nodes in each (minimal overlap) subdomain have been assigned to the same aggregate, while larger values of χ indicate that the coarse space has been enriched by using more aggregates on each subdomain.

Tables 1 and 3 report the estimated condition number for nonsmoothed aggregation, using $\delta = \delta_0 = h$ (minimal overlap), for different values of χ . From Theorem 4.5, the condition number in this particular case has a linear dependency on h^{-1} . This is confirmed by the tables. For example, in two dimensions with $\chi = 1$ and $H = 1/10$, the condition number increases from 110.35 to 220.01 when h decreases from $1/240$ to $1/480$.

Tables 2 and 4 use $\delta = h$ and $\delta_0 = 3h$. Although the proof of Theorem 4.5 requires $\delta_0 = \delta$, these results suggest that, at least for this problem, this condition may be relaxed. Especially for small values of χ , $\kappa(B_{sa}'')$ is significantly smaller than $\kappa(B_{nsa}'')$. (Should $\delta = \delta_0 = 3h$, then $\kappa(B_{sa}'') = \kappa(B_{nsa}'')/3$.) Therefore, it may be convenient, in a parallel environment, to use smoothed aggregation even if the subdomains share minimal overlap.

Theorem 4.5 also states that the quantity

$$(5.3) \quad \frac{\kappa(B''A) \delta}{H}$$

depends linearly on χ . Figure 2 displays graphically the quantity (5.3) for data in Tables 1 (figure on the left) and 2 (figure on the right) with respect to χ . The figure

TABLE 1
 2D Laplace problem. Estimated condition number using B''_{nsa} ($\delta = \delta_0 = h$). $\chi = H_0/H$.

		$\chi = 1$	$\chi = 1/2$	$\chi = 1/4$	$\chi = 1/8$
$h^{-1} = 120$	$H^{-1} = 4$	103.94	70.08	38.64	—
	$H^{-1} = 6$	81.88	50.31	33.28	—
	$H^{-1} = 8$	66.41	40.32	27.62	—
	$H^{-1} = 10$	—	—	—	—
$h^{-1} = 180$	$H^{-1} = 4$	155.67	101.57	61.87	34.63
	$H^{-1} = 6$	122.41	78.46	43.05	29.37
	$H^{-1} = 8$	99.69	59.39	37.66	24.95
	$H^{-1} = 10$	82.95	45.92	25.35	—
$h^{-1} = 240$	$H^{-1} = 4$	207.63	144.71	84.06	44.26
	$H^{-1} = 6$	163.01	99.71	58.09	39.90
	$H^{-1} = 8$	132.15	82.65	45.76	30.24
	$H^{-1} = 10$	110.35	62.71	35.57	—
$h^{-1} = 360$	$H^{-1} = 4$	311.11	195.71	135.98	70.37
	$H^{-1} = 6$	244.24	162.47	90.01	46.74
	$H^{-1} = 8$	197.94	118.94	69.23	36.60
	$H^{-1} = 10$	164.33	105.37	58.47	—
$h^{-1} = 480$	$H^{-1} = 4$	—	—	—	—
	$H^{-1} = 6$	325.49	206.15	124.88	62.37
	$H^{-1} = 8$	263.50	171.13	92.97	47.96
	$H^{-1} = 10$	220.01	141.09	74.44	39.91

TABLE 2
 2D Laplace problem. Estimated condition number using B''_{sa} . $\chi = H_0/H$.

		$\chi = 1$	$\chi = 1/2$	$\chi = 1/4$	$\chi = 1/8$
$h^{-1} = 120$	$H^{-1} = 4$	73.76	40.03	20.14	—
	$H^{-1} = 6$	51.22	26.02	12.55	—
	$H^{-1} = 8$	38.63	19.52	11.70	—
	$H^{-1} = 10$	—	—	—	—
$h^{-1} = 180$	$H^{-1} = 4$	111.24	58.62	32.03	14.55
	$H^{-1} = 6$	77.47	41.29	20.33	9.36
	$H^{-1} = 8$	59.68	30.52	14.65	10.09
	$H^{-1} = 10$	47.74	23.67	15.80	9.22
$h^{-1} = 240$	$H^{-1} = 4$	150.60	85.59	44.78	21.10
	$H^{-1} = 6$	105.53	53.70	27.34	13.06
	$H^{-1} = 8$	80.18	43.26	20.79	9.84
	$H^{-1} = 10$	64.31	31.96	—	8.80
$h^{-1} = 360$	$H^{-1} = 4$	226.59	119.54	64.48	34.28
	$H^{-1} = 6$	158.85	89.74	49.59	21.07
	$H^{-1} = 8$	120.15	61.05	22.86	14.81
	$H^{-1} = 10$	95.37	54.32	27.51	13.61
$h^{-1} = 480$	$H^{-1} = 4$	—	—	—	—
	$H^{-1} = 6$	213.22	115.59	65.38	29.60
	$H^{-1} = 8$	158.63	91.45	46.92	21.11
	$H^{-1} = 10$	129.60	76.55	34.69	16.60

also reports the least-square approximation of the considered data. The scatter in the data is primarily due to the presence of the Dirichlet nodes, which make the linear system easier to solve for large values of h and small values of H . While there is some scatter, the overall trend of the data closely matches the asymptotic behavior predicted by the theory.

TABLE 3
3D Laplace problem. Estimated condition number using B''_{nsa} . $\chi = H_0/H$.

		$\chi = 1$	$\chi = 1/2$	$\chi = 1/4$
$h^{-1} = 36$	$H^{-1} = 2$	42.02	32.92	23.15
	$H^{-1} = 3$	39.51	26.82	19.89
	$H^{-1} = 4$	35.34	22.68	16.63
	$H^{-1} = 5$	—	—	—
$h^{-1} = 48$	$H^{-1} = 2$	55.79	43.44	30.54
	$H^{-1} = 3$	52.43	36.21	23.18
	$H^{-1} = 4$	46.85	30.04	21.60
	$H^{-1} = 5$	41.25	27.36	17.61
$h^{-1} = 72$	$H^{-1} = 2$	—	—	—
	$H^{-1} = 3$	78.38	54.61	35.36
	$H^{-1} = 4$	69.90	46.40	28.71
	$H^{-1} = 5$	—	—	—
$h^{-1} = 96$	$H^{-1} = 2$	—	—	—
	$H^{-1} = 3$	104.37	75.31	47.37
	$H^{-1} = 4$	92.99	61.20	37.61
	$H^{-1} = 5$	—	—	—

TABLE 4
3D Laplace problem. Estimated condition number using B''_{sa} . $\chi = H_0/H$.

		$\chi = 1$	$\chi = 1/2$	$\chi = 1/4$
$h^{-1} = 36$	$H^{-1} = 2$	—	—	14.76
	$H^{-1} = 3$	34.11	18.54	9.53
	$H^{-1} = 4$	28.22	13.92	7.70
	$H^{-1} = 5$	—	—	—
$h^{-1} = 48$	$H^{-1} = 2$	52.72	34.92	20.12
	$H^{-1} = 3$	45.56	24.67	13.53
	$H^{-1} = 4$	37.90	18.82	9.76
	$H^{-1} = 5$	30.56	17.06	8.33
$h^{-1} = 72$	$H^{-1} = 2$	—	—	—
	$H^{-1} = 3$	68.86	39.46	21.67
	$H^{-1} = 4$	57.32	30.91	15.66
	$H^{-1} = 5$	—	—	—
$h^{-1} = 96$	$H^{-1} = 2$	—	—	—
	$H^{-1} = 3$	91.85	—	—
	$H^{-1} = 4$	76.66	40.49	22.34
	$H^{-1} = 5$	—	—	—

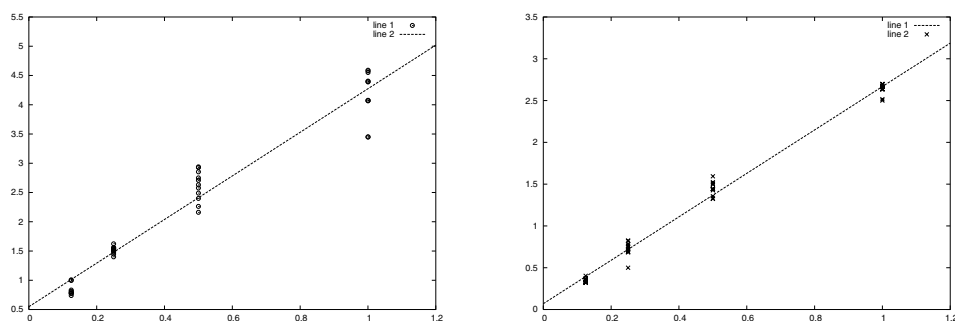


FIG. 2. Graphical representation of data contained in Table 1 (left) and Table 2 (right). The x -axis reports the values of $\chi = H_0/H$; the y -axis reports $\kappa\delta/H$. The dots represent the data in the table; the line is the least-square approximation.

6. Conclusions. We have presented an improved convergence bound for two-level additive domain decomposition preconditioners based on aggregation. This bound generalizes other results presented in the literature. Numerical results are reported to confirm the proposed estimate.

REFERENCES

- [1] M. BREZINA AND P. VANĚK, *A black-box iterative solver based on a two-level Schwarz method*, Computing, 63 (1999), pp. 233–263.
- [2] W. L. BRIGGS, V. E. HENSON, AND S. F. MCCORMICK, *A Multigrid Tutorial*, 2nd ed., SIAM, Philadelphia, 2000.
- [3] M. DRYJA AND O. B. WIDLUND, *Towards a unified theory of domain decomposition algorithms for elliptic problems*, in Proceedings of the Third International Symposium on Domain Decomposition Methods for Partial Differential Equations, T. F. Chan, R. Glowinski, J. Péreaux, and O. Widlund, eds., SIAM, Philadelphia, PA, 1990, pp. 3–21.
- [4] M. DRYJA AND O. B. WIDLUND, *Domain decomposition algorithms with small overlap*, SIAM J. Sci. Comput., 15 (1994), pp. 604–620.
- [5] M. DRYJA AND O. B. WIDLUND, *An overlapping domain decomposition preconditioner for a class of discontinuous Galerkin approximations of advection-diffusion problems*, Math. Comput., 72 (2003), pp. 1215–1238.
- [6] L. FORMAGGIA, A. SCHEININE, AND A. QUARTERONI, *A numerical investigation of Schwarz domain decomposition techniques for elliptic problems on unstructured grids*, Math. Comput. Simulations, 44 (1994), pp. 313–330.
- [7] G. H. GOLUB AND C. F. VAN LOAN, *Matrix Computations*, 3rd ed., The Johns Hopkins University Press, Baltimore, MD, 1996.
- [8] M. HESTENES AND E. STEIFEL, *Method of conjugate gradients for solving linear systems*, J. Res., 20 (1952), pp. 409–436.
- [9] L. JENKINS, T. KELLEY, C. T. MILLER, AND C. E. KEES, *An Aggregation-Based Domain Decomposition Preconditioner for Groundwater Flow*, Tech. report TR00–13, Department of Mathematics, North Carolina State University, Raleigh, NC, 2000.
- [10] C. E. KEES, C. T. MILLER, E. W. JENKINS, AND C. T. KELLEY, *Versatile Multilevel Schwarz Preconditioners for Multiphase Flow*, Tech. report CRSC-TR01-32, Center for Research in Scientific Computation, North Carolina State University, Raleigh, NC, 2001.
- [11] C. LASSER AND A. TOSELLI, *Convergence of some two-level overlapping domain decomposition preconditioners with smoothed aggregation coarse spaces*, in Recent Developments in Domain Decomposition Methods, Lecture Notes in Comput. Sci. Engrg. 23, L. Pavarino and A. Toselli, eds., Springer-Verlag, Berlin, 2002, pp. 95–117.
- [12] W. LEONTIEF, *The Structure of the American Economy*, Oxford University Press, New York, 1951.
- [13] J. MANDEL AND B. SEKERKA, *A local convergence proof for the iterative aggregation method*, Linear Algebra Appl., 51 (1983), pp. 163–172.
- [14] L. PAGLIERI, A. SCHEININE, L. FORMAGGIA, AND A. QUARTERONI, *Parallel conjugate gradient with Schwarz preconditioner applied to fluid dynamics problems*, in Proceedings of the Conference on Parallel Computational Fluid Dynamics, Algorithms and Results Using Advanced Computers (Capri, Italy, May 20–23, 1996), P. Schiano et al., eds., North-Holland, Amsterdam, 1997, pp. 21–30.
- [15] A. QUARTERONI AND A. VALLI, *Numerical Approximation of Partial Differential Equations*, Springer-Verlag, Berlin, 1994.
- [16] M. SALA, *Domain Decomposition Preconditioners: Theoretical Properties, Application to the Compressible Euler Equations, Parallel Aspects*, Ph.D. thesis, Ecole Polytechnique Fédérale de Lausanne, Switzerland, 2003. Available online at <http://library.epfl.ch/en/theses/?nr=2733>
- [17] M. SALA, *Analysis of two-level domain decomposition preconditioners based on aggregation*, Math. Model. Numer. Anal., 38 (2004), pp. 765–780.
- [18] L. FORMAGGIA, M. SALA, AND F. SALERI, *Domain decomposition techniques*, in Numerical Solutions of Partial Differential Equations on Parallel Computers, A. M. Brauset et al., eds., Springer-Verlag, Berlin, 2006, to appear.
- [19] M. SALA AND L. FORMAGGIA, *Algebraic coarse grid operators for domain decomposition based preconditioners*, in Parallel Computational Fluid Dynamics—Practice and Theory, P. Wilders et al., eds., Elsevier Science, Amsterdam, The Netherlands, 2002, pp. 119–126.

- [20] B. F. SMITH, P. BJØRSTAD, AND W. D. GROPP, *Parallel Multilevel Methods for Elliptic Partial Differential Equations*, Cambridge University Press, Cambridge, UK, 1996.
- [21] R. S. TUMINARO AND C. TONG, *Parallel smoothed aggregation multigrid: Aggregation strategies on massively parallel machines*, in Proceedings of the 2000 ACM/IEEE Conference on Supercomputing (Dallas, TX), IEEE, Washington, DC, 2000, p. 5 (CD-ROM).
- [22] P. VANĚK, M. BREZINA, AND J. MANDEL, *Convergence of algebraic multigrid based on smoothed aggregation*, Numer. Math., 88 (2001), pp. 559–579.
- [23] P. VANĚK, M. BREZINA, AND R. TEZAUR, *Two-grid method for linear elasticity on unstructured meshes*, SIAM J. Sci. Comput., 21 (1999), pp. 900–923.

REPRESENTATIONS FOR THE DRAZIN INVERSE OF A 2×2 BLOCK MATRIX*

ROBERT HARTWIG[†], XIEZHANG LI[‡], AND YIMIN WEI[§]

Abstract. Two representations for the Drazin inverse of a 2×2 block matrix $M = \begin{bmatrix} A & B \\ C & D \end{bmatrix}$, where A and D are square matrices, in terms of the Drazin inverses of A and D have been recently developed under the assumptions that $C(I - AA^D) = 0$ and $(I - AA^D)B = 0$, and that the generalized Schur complement $D - CA^DB$ is either nonsingular or zero. These two representations of M^D are extended to the case where $C(I - AA^D) = 0$ and $(I - AA^D)B = 0$ are substituted with $C(I - AA^D)B = 0$ and $A(I - AA^D)B = 0$. Moreover, upper bounds for the index of M are studied. Numerical examples are given to illustrate the new results.

Key words. index, Drazin inverse, generalized Schur complement, weighted Drazin inverse

AMS subject classifications. 15A09, 65F20

DOI. 10.1137/040606685

1. Introduction. Let A be an $n \times n$ complex matrix. The smallest nonnegative integer k such that $\text{rank}(A^{k+1}) = \text{rank}(A^k)$, denoted by $\text{Ind}(A)$, is called the *index* of A . Let $\text{Ind}(A) = k$. There is a unique matrix, called the *Drazin inverse* of A and denoted by A^D [4], satisfying the following equations:

$$(1.1) \quad A^k A^D A = A^k, \quad A^D A A^D = A^D, \quad A A^D = A^D A.$$

In the case where $\text{Ind}(A) \leq 1$, A^D is called the *group inverse* of A and denoted by $A^\#$. In particular, A is nonsingular if and only if $\text{Ind}(A) = 0$, and in this case $A^D = A^{-1}$.

The Drazin inverse of a square matrix has various applications in singular differential equations and singular difference equations, Markov chains, and iterative methods (cf. [3, 4, 7, 10]). The perturbation theory of the Drazin inverse has been recently studied (cf. [11, 16, 17]). We introduce briefly an application of the Drazin inverse of a 2×2 block matrix. Suppose that E and F are $n \times n$ matrices, where E is singular. Assume that there exists a scalar μ such that $\mu E + F$ is nonsingular. Then the general solution of the singular system of differential equations

$$Ex'(t) + Fx(t) = 0, \quad t \geq t_0,$$

is given by (cf. [2, 3])

$$(1.2) \quad x(t) = e^{-\hat{E}^D \hat{F}(t-t_0)} \hat{E}^D \hat{E} q,$$

*Received by the editors April 13, 2004; accepted for publication (in revised form) by I. S. Dhillon June 27, 2005; published electronically January 6, 2006.

<http://www.siam.org/journals/simax/27-3/60668.html>

[†]Department of Mathematics, North Carolina State University, Harrelson Hall, P.O. Box 8205, Raleigh, NC 27695 (hartwig@math.ncsu.edu).

[‡]Corresponding author. Department of Mathematical Sciences, Georgia Southern University, Statesboro, GA 30460 (xli@georgiasouthern.edu). Part of this work was done during the author's visit to Fudan University, China.

[§]School of Mathematical Sciences, Fudan University, Shanghai 200433, P. R. China, and Key Laboratory of Mathematics for Nonlinear Sciences (Fudan University), Ministry of Education, Shanghai 200433, P. R. China (ymwei@fudan.edu.cn). The work of this author was supported by National Natural Science Foundation of China grant 10471027 and by the Shanghai Education Committee.

where

$$\hat{E} = (\mu E + F)^{-1}E, \quad \hat{F} = (\mu E + F)^{-1}F,$$

and q is an arbitrary vector of dimension n .

Now we consider a second-order system

$$(1.3) \quad Ex''(t) + Fx'(t) + Gx(t) = 0,$$

where again E is singular. It is reasonable to assume that there is a λ such that $\lambda^2 E + \lambda F + G$ is invertible. Otherwise, the solution will not be uniquely determined by a consistent initial condition. Assume that G is invertible. With $x(t) = e^{\lambda t}y(t)$, (1.3) is equivalent to

$$(\lambda^2 E + \lambda F + G)^{-1}Ey''(t) + (\lambda^2 E + \lambda F + G)^{-1}(F + 2\lambda E)y'(t) + y(t) = 0.$$

Furthermore, let $w(t) = y'(t)$. The above system is equivalent to the first-order system

$$(1.4) \quad \begin{bmatrix} 0 & -I \\ \tilde{E} & \tilde{F} \end{bmatrix} \begin{bmatrix} w \\ y \end{bmatrix}' + \begin{bmatrix} I & 0 \\ 0 & I \end{bmatrix} \begin{bmatrix} w \\ y \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \end{bmatrix},$$

where

$$(1.5) \quad \tilde{E} = (\lambda^2 E + \lambda F + G)^{-1}E \quad \text{and} \quad \tilde{F} = (\lambda^2 E + \lambda F + G)^{-1}(F + 2\lambda E).$$

It is pointed out that the inverse

$$\left(\mu \begin{bmatrix} 0 & -I \\ \tilde{E} & \tilde{F} \end{bmatrix} + \begin{bmatrix} I & 0 \\ 0 & I \end{bmatrix} \right)^{-1}$$

does exist whenever $|\mu|$ is sufficiently small. In order to express the solutions of (1.4) explicitly in terms of $\tilde{E}$ and $\tilde{F}$, we need to find an explicit representation for the Drazin inverse of a 2×2 block matrix,

$$(1.6) \quad \hat{E} = \begin{bmatrix} I & -\mu I \\ \mu \tilde{E} & \mu \tilde{F} + I \end{bmatrix}^{-1} \begin{bmatrix} 0 & -\mu I \\ \mu \tilde{E} & \mu \tilde{F} \end{bmatrix}.$$

Another differential equation example with the Drazin inverse of a 2×2 block matrix is given in [18]. Finding an explicit representation for the Drazin inverse of a general 2×2 block matrix, posed by Campbell in [3], appears to be difficult.

Next we will recall some properties of the Drazin inverse of a square matrix A . Suppose that A has a decomposition

$$(1.7) \quad A = X \begin{bmatrix} \Sigma & 0 \\ 0 & N \end{bmatrix} X^{-1},$$

where X and Σ are nonsingular matrices, and N is a nilpotent matrix. Then by definition we have

$$(1.8) \quad A^D = X \begin{bmatrix} \Sigma^{-1} & 0 \\ 0 & 0 \end{bmatrix} X^{-1} \quad \text{and} \quad \text{Ind}(N) = \text{Ind}(A) = k.$$

The following properties (cf. [1, 4]) of the Drazin inverse of a square matrix A with index k are direct results from (1.8):

- (i) $R(A^D) = R(A^k)$, where $R(A^D)$ is the range of A^D ;

- (ii) $N(A^D) = N(A^k)$, where $N(A^D)$ is the null space of A^D ;
- (iii) $AA^D = P_{R(A^k), N(A^k)}$ is the projection on $R(A^k)$ along $N(A^k)$ (cf. [1, p. 59]);
- (iv) $(I - AA^D) = P_{N(A^k), R(A^k)}$;
- (v) for a nonnegative integer p , $A^{p+1}A^D = A^p$ if and only if $p \geq k$;
- (vi) $(A^*)^D = (A^D)^*$.

Let X and X^{-1} in (1.7) be partitioned respectively as

$$(1.9) \quad X = \begin{bmatrix} U_1 & U_2 \end{bmatrix} \quad \text{and} \quad X^{-1} = \begin{bmatrix} V_1 \\ V_2 \end{bmatrix},$$

where the column dimensions of U_1 and V_1^* are the same as those of Σ . Then we may write the following useful identities:

$$(1.10) \quad \begin{aligned} A &= U_1 \Sigma V_1 + U_2 N V_2, & A^D &= U_1 \Sigma^{-1} V_1, \\ AA^D &= U_1 V_1, & I - AA^D &= U_2 V_2. \end{aligned}$$

Let $M = \begin{bmatrix} A & B \\ C & D \end{bmatrix}$ be a 2×2 block matrix; throughout the paper A and D are always assumed to be square but their sizes need not be the same. There is no explicit formula for the Drazin inverse of M in terms of A^D and D^D with arbitrary A, B, C , and D (cf. [4, p. 122]). However, a general expression for the Drazin inverse of a 2×2 block triangular matrix (either $B = 0$ or $C = 0$) is known (cf. [8, 13]). For example (cf. [4, Thm. 7.7.1]), if $k = \text{Ind}(A)$ and $j = \text{Ind}(D)$, then

$$(1.11) \quad \begin{bmatrix} A & B \\ 0 & D \end{bmatrix}^D = \begin{bmatrix} A^D & Y \\ 0 & D^D \end{bmatrix},$$

where

$$\begin{aligned} Y &= (A^D)^2 \left[\sum_{i=0}^j (A^D)^i B D^i \right] (I - D D^D) \\ &\quad + (I - A A^D) \left[\sum_{i=0}^{k-1} A^i B (D^D)^i \right] (D^D)^2 - A^D B D^D. \end{aligned}$$

(We define $0^0 = I$.)

Other representations for the group inverse of a block matrix were developed in [5, 9, 14]. Recently, the Drazin inverse of a 2×2 block matrix for two special cases was studied by Wei [15] and Miao [12].

THEOREM 1.1 (see [15]). *Let $M = \begin{bmatrix} A & B \\ C & D \end{bmatrix}$, where A and D are square matrices. If*

$$(1.12) \quad C(I - A A^D) = 0 \quad \text{and} \quad (I - A A^D)B = 0$$

and the generalized Schur complement $Z = D - C A^D B$ is nonsingular, then

$$(1.13) \quad M^D = \begin{bmatrix} A^D + A^D B Z^{-1} C A^D & -A^D B Z^{-1} \\ -Z^{-1} C A^D & Z^{-1} \end{bmatrix}.$$

THEOREM 1.2 (see [12]). *Let $M = \begin{bmatrix} A & B \\ C & D \end{bmatrix}$, where A and D are square matrices. If*

$$C(I - A A^D) = 0, \quad (I - A A^D)B = 0,$$

and the generalized Schur complement $Z = D - CA^DB$ is equal to 0, then

$$(1.14) \quad M^D = \begin{bmatrix} I \\ CA^D \end{bmatrix} [(AW)^D]^2 A \begin{bmatrix} I & A^D B \end{bmatrix},$$

where $W = AA^D + A^D B C A^D$.

The objective of this paper is to derive formulae for the Drazin inverse of a 2×2 block matrix under conditions weaker than those in the two theorems above. The paper is organized as follows. A key lemma about the Drazin inverse of a 2×2 block matrix is proved in section 2. In section 3 the key lemma will be applied to derive new formulae for M^D which are extensions of (1.13) in Theorem 1.1. Other formulae for M^D with a pattern similar to those derived in section 3 are developed in section 4. Two examples are given to illustrate the new representations.

2. Key lemma. First we present an additive result for the Drazin inverse of $P + Q$, which is closely connected to the Drazin inverse of a 2×2 block matrix.

LEMMA 2.1 (see [10]). *Let P and Q be square matrices of the same order. If $PQ = 0$ and $Q^2 = 0$, then*

$$(P + Q)^D = P^D + Q(P^D)^2.$$

A representation for $\begin{bmatrix} A & B \\ C & D \end{bmatrix}^D$ has been given in [7] for the case where $BC = 0$, $DC = 0$, and $BD = 0$. An alternative representation for $\begin{bmatrix} A & B \\ C & D \end{bmatrix}^D$ is derived for the case where the condition $BD = 0$ is replaced by the condition that D is nilpotent, as shown in the following key lemma.

LEMMA 2.2. *Let $M = \begin{bmatrix} A & B \\ C & D \end{bmatrix}$, where A and D are square matrices. If $BC = 0$, $DC = 0$, and D is nilpotent, then*

$$(2.1) \quad \text{Ind}(M) \leq \text{Ind}(A) + \text{Ind}(D) + 1$$

and

$$(2.2) \quad M^D = \begin{bmatrix} I \\ CA^D \end{bmatrix} A^D \begin{bmatrix} I & \sum_{i=0}^{j-1} (A^D)^{i+1} BD^i \end{bmatrix}, \quad \text{where } j := \text{Ind}(D).$$

Proof. We rewrite $M = P + Q$, where $P = \begin{bmatrix} A & B \\ 0 & D \end{bmatrix}$ and $Q = \begin{bmatrix} 0 & 0 \\ C & 0 \end{bmatrix}$. With $BC = 0$ and $DC = 0$, it is easy to verify that $PQ = 0$ and $Q^2 = 0$. It follows from (1.11) and $D^D = 0$ that

$$P^D = \begin{bmatrix} A^D & \sum_{i=0}^{j-1} (A^D)^{i+2} BD^i \\ 0 & 0 \end{bmatrix} \quad \text{for } j = \text{Ind}(D).$$

Then we have

$$Q(P^D)^2 = \begin{bmatrix} 0 & 0 \\ C(A^D)^2 & CA^D \sum_{i=0}^{j-1} (A^D)^{i+2} BD^i \end{bmatrix}.$$

The explicit formula (2.2) for M^D immediately follows from Lemma 2.1.

Now we derive an upper bound for $\text{Ind}(M)$. It follows from $BC = 0$ and $DC = 0$ by induction that

$$M^p = \begin{bmatrix} A^p & \sum_{i=0}^{p-1} A^{p-i-1} BD^i \\ CA^{p-1} & C \sum_{i=0}^{p-2} A^{p-i-2} BD^i + D^p \end{bmatrix} \quad \text{for } p \geq 1,$$

where by convention $G^0 = I$ for any square matrix G . It follows from $D^p = 0$ for $p \geq j = \text{Ind}(D)$ that

$$M^p = \begin{bmatrix} A^p & \sum_{i=0}^{p-1} A^{p-i-1} BD^i \\ CA^{p-1} & C \sum_{i=0}^{p-2} A^{p-i-2} BD^i \end{bmatrix} \quad \text{for } p \geq j = \text{Ind}(D).$$

Thus

$$\begin{aligned} M^{p+1}M^D &= \begin{bmatrix} A^{p+1} & \sum_{i=0}^p A^{p-i} BD^i \\ CA^p & C \sum_{i=0}^{p-1} A^{p-i-1} BD^i \end{bmatrix} \begin{bmatrix} A^D & \sum_{i=0}^{j-1} (A^D)^{i+2} BD^i \\ C(A^D)^2 & CA^D \sum_{i=0}^{j-1} (A^D)^{i+2} BD^i \end{bmatrix} \\ &= \begin{bmatrix} A^{p+1}A^D & A^{p+1} \sum_{i=0}^{j-1} (A^D)^{i+2} BD^i \\ CA^pA^D & CA^p \sum_{i=0}^{j-1} (A^D)^{i+2} BD^i \end{bmatrix} \\ &= \begin{bmatrix} A^{p+1}A^D & \sum_{i=0}^{j-1} A^{p-i-1} BD^i \\ CA^pA^D & C \sum_{i=0}^{j-1} A^{p-i-2} BD^i \end{bmatrix}. \end{aligned}$$

The last equality holds because $A^p(A^D)^{i+2} = A^{p-i-2}$ for $i = 0, 1, \dots, j-1$, by property (v) in section 1, provided $p-j-1 \geq \text{Ind}(A)$. Therefore,

$$M^{p+1}M^D = M^p \quad \text{for } p \geq \text{Ind}(A) + j + 1.$$

Equation (2.1) follows by applying property (v) once again. This completes the proof. $\square$

An immediate corollary follows from property (vi) in section 1.

COROLLARY 2.3. *Let $M = \begin{bmatrix} A & B \\ C & D \end{bmatrix}$. If $BC = 0$, $BD = 0$, and D is nilpotent, then*

$$\text{Ind}(M) \leq \text{Ind}(A) + \text{Ind}(D) + 1$$

and

$$M^D = \begin{bmatrix} I & \\ \sum_{i=0}^{j-1} D^i C (A^D)^{i+1} & \end{bmatrix} A^D \begin{bmatrix} I & A^D B \end{bmatrix}, \quad \text{where } j = \text{Ind}(D).$$

Before the end of this section, we introduce another simple lemma which will be used in the following sections.

LEMMA 2.4. *Let M be an $n \times n$ matrix. If G and H are $m \times n$ and $n \times m$ matrices, respectively, where $m \geq n$ and $HG = I_n$, then*

$$(GMH)^D = GM^D H$$

and

$$\begin{aligned} \text{Ind}(GMH) &= 0 \text{ if } M \text{ is nonsingular and } m = n, \\ \text{Ind}(GMH) &= 1 \text{ if } M \text{ is nonsingular and } m > n, \\ \text{Ind}(GMH) &= \text{Ind}(M) \text{ if } M \text{ is singular.} \end{aligned}$$

Proof. The case where M is nonsingular and $m = n$ is trivial. With the assumptions on G and H , we have

$$\text{rank}((GMH)^p) = \text{rank}(GM^p H) = \text{rank}(M^p) \quad \text{for any positive integer } p.$$

Let M be nonsingular and $m > n$. It is easy to verify that

$$\text{rank}(GMH) = \text{rank}(M) = n < m,$$

but

$$\text{rank}((GMH)^2) = \text{rank}(M^2) = \text{rank}(M) = \text{rank}(GMH).$$

Thus, $\text{Ind}(GMH) = 1$. One can check that $GM^{-1}H$ is the group inverse of GMH .

Now we consider the last case, where M is singular (or $\text{Ind}(M) = k > 0$). We have

$$\text{rank}(GMH) = \text{rank}(M) < n < m,$$

which implies $\text{Ind}(GMH) > 0$. Thus k is the smallest positive integer such that

$$\text{rank}((GMH)^{k+1}) = \text{rank}(M^{k+1}) = \text{rank}(M^k) = \text{rank}((GMH)^k).$$

In other words, $\text{Ind}(GMH) = \text{Ind}(M) = k$. It can be verified that

$$\begin{aligned} (GMH)^k (GM^D H) (GMH) &= (GMH)^k, \\ (GM^D H) (GMH) (GM^D H) &= GM^D H, \\ (GMH) (GM^D H) &= (GM^D H) (GMH). \end{aligned}$$

Therefore, $(GMH)^D = GM^D H$. $\square$

3. Nonsingular generalized Schur complement. In this section, we derive three representations for $\begin{bmatrix} A & B \\ C & D \end{bmatrix}^D$ which are extensions of Theorem 1.1. Let $M = \begin{bmatrix} A & B \\ C & D \end{bmatrix}$. Assume that A has a decomposition (1.7). Using the notation in (1.9), we can define

$$\begin{bmatrix} P_1 \\ P_2 \end{bmatrix} := \begin{bmatrix} V_1 B \\ V_2 B \end{bmatrix} = X^{-1} B \quad \text{and} \quad \begin{bmatrix} Q_1 & Q_2 \end{bmatrix} := \begin{bmatrix} CU_1 & CU_2 \end{bmatrix} = CX.$$

Then

$$\begin{bmatrix} X^{-1} & 0 \\ 0 & I \end{bmatrix} \begin{bmatrix} A & B \\ C & D \end{bmatrix} \begin{bmatrix} X & 0 \\ 0 & I \end{bmatrix} = \begin{bmatrix} \Sigma & 0 & P_1 \\ 0 & N & P_2 \\ Q_1 & Q_2 & D \end{bmatrix},$$

or equivalently,

$$(3.1) \quad M = \begin{bmatrix} A & B \\ C & D \end{bmatrix} = \begin{bmatrix} U_1 & U_2 & 0 \\ 0 & 0 & I \end{bmatrix} \begin{bmatrix} \Sigma & 0 & P_1 \\ 0 & N & P_2 \\ Q_1 & Q_2 & D \end{bmatrix} \begin{bmatrix} V_1 & 0 \\ V_2 & 0 \\ 0 & I \end{bmatrix}.$$

We have

$$(3.2) \quad \begin{bmatrix} I & 0 & 0 \\ 0 & 0 & I \\ 0 & I & 0 \end{bmatrix} \begin{bmatrix} \Sigma & 0 & P_1 \\ 0 & N & P_2 \\ Q_1 & Q_2 & D \end{bmatrix} \begin{bmatrix} I & 0 & 0 \\ 0 & 0 & I \\ 0 & I & 0 \end{bmatrix} = \begin{bmatrix} \Sigma & P_1 & 0 \\ Q_1 & D & Q_2 \\ 0 & P_2 & N \end{bmatrix}.$$

Letting

$$M_1 := \begin{bmatrix} \Sigma & P_1 & 0 \\ Q_1 & D & Q_2 \\ 0 & P_2 & N \end{bmatrix},$$

it follows from (3.1) and (3.2) that

$$M = \begin{bmatrix} U_1 & 0 & U_2 \\ 0 & I & 0 \end{bmatrix} M_1 \begin{bmatrix} V_1 & 0 \\ 0 & I \\ V_2 & 0 \end{bmatrix}.$$

Notice that both M and M_1 are square matrices of the same order and that the product of two matrices,

$$\begin{bmatrix} V_1 & 0 \\ 0 & I \\ V_2 & 0 \end{bmatrix} \times \begin{bmatrix} U_1 & 0 & U_2 \\ 0 & I & 0 \end{bmatrix},$$

is the identity matrix. Consequently, it follows from Lemma 2.4 that

$$(3.3) \quad \text{Ind}(M) = \text{Ind}(M_1) \quad \text{and} \quad M^D = \begin{bmatrix} U_1 & 0 & U_2 \\ 0 & I & 0 \end{bmatrix} M_1^D \begin{bmatrix} V_1 & 0 \\ 0 & I \\ V_2 & 0 \end{bmatrix}.$$

In order to get an explicit formula for M_1^D with an application of Lemma 2.2, we repartition M_1 as a 2×2 block matrix, i.e.,

$$(3.4) \quad M_1 = \begin{bmatrix} \Sigma & P_1 & 0 \\ Q_1 & D & Q_2 \\ 0 & P_2 & N \end{bmatrix} = \begin{bmatrix} F & \tilde{Q}_2 \\ \tilde{P}_2 & N \end{bmatrix},$$

where

$$(3.5) \quad F = \begin{bmatrix} \Sigma & P_1 \\ Q_1 & D \end{bmatrix}, \quad \tilde{P}_2 = [0 \quad P_2], \quad \text{and} \quad \tilde{Q}_2 = \begin{bmatrix} 0 \\ Q_2 \end{bmatrix}.$$

Now, we can prove one of the main results of this paper.

THEOREM 3.1. *Let $M = \begin{bmatrix} A & B \\ C & D \end{bmatrix}$, where A and D are square matrices. If*

$$(3.6) \quad C(I - AA^D)B = 0, \quad A(I - AA^D)B = 0,$$

and $Z = D - CA^D B$ is nonsingular, then $\text{Ind}(M) \leq \text{Ind}(A) + 1$ and

$$(3.7) \quad M^D = \left(I + \begin{bmatrix} 0 & (I - AA^D)B \\ 0 & 0 \end{bmatrix} R \right) R \left(I + \sum_{i=0}^{k-1} R^{i+1} \begin{bmatrix} 0 & 0 \\ C(I - AA^D)A^i & 0 \end{bmatrix} \right),$$

where $k = \text{Ind}(A)$ and

$$(3.8) \quad R = \begin{bmatrix} A^D + A^D B Z^{-1} C A^D & -A^D B Z^{-1} \\ -Z^{-1} C A^D & Z^{-1} \end{bmatrix}.$$

Proof. We first convert (3.6) into an equivalent form in terms of matrices $\tilde{P}_2$, $\tilde{Q}_2$, and N . Using the identities of (1.10), we obtain

$$\begin{aligned} C(I - AA^D)B &= C U_2 V_2 B = Q_2 P_2, \\ A(I - AA^D)B &= (U_1 \Sigma V_1 + U_2 N V_2) U_2 V_2 B = U_2 N P_2. \end{aligned}$$

Then (3.6) is equivalent to

$$\tilde{Q}_2 \tilde{P}_2 = 0 \quad \text{and} \quad N \tilde{P}_2 = 0.$$

Recall that N is nilpotent. Applying Lemma 2.2 to M_1 in (3.4) yields

$$\text{Ind}(M_1) \leq \text{Ind}(F) + \text{Ind}(N) + 1$$

and

$$(3.9) \quad M_1^D = \begin{bmatrix} I \\ S \end{bmatrix} F^D \begin{bmatrix} I & \sum_{i=0}^{k-1} (F^D)^i T N^i \end{bmatrix}, \quad k = \text{Ind}(A),$$

where F is given by (3.5) and

$$S = \tilde{P}_2 F^D = \begin{bmatrix} 0 & P_2 \end{bmatrix} F^D, \quad T = F^D \tilde{Q}_2 = F^D \begin{bmatrix} 0 \\ Q_2 \end{bmatrix}.$$

Substituting (3.9) into (3.3), we have

$$(3.10) \quad \begin{aligned} M^D &= \left(\begin{bmatrix} U_1 & 0 \\ 0 & I \end{bmatrix} + \begin{bmatrix} U_2 \\ 0 \end{bmatrix} S \right) F^D \left(\begin{bmatrix} V_1 & 0 \\ 0 & I \end{bmatrix} + \sum_{i=0}^{k-1} (F^D)^i T N^i \begin{bmatrix} V_2 & 0 \end{bmatrix} \right) \\ &= \left(I + \begin{bmatrix} U_2 \\ 0 \end{bmatrix} S \begin{bmatrix} V_1 & 0 \\ 0 & I \end{bmatrix} \right) R \left(I + \begin{bmatrix} U_1 & 0 \\ 0 & I \end{bmatrix} \sum_{i=0}^{k-1} (F^D)^i T N^i \begin{bmatrix} V_2 & 0 \end{bmatrix} \right), \end{aligned}$$

where

$$(3.11) \quad R = \begin{bmatrix} U_1 & 0 \\ 0 & I \end{bmatrix} F^D \begin{bmatrix} V_1 & 0 \\ 0 & I \end{bmatrix}.$$

It is easy to verify that

$$(3.12) \quad \begin{aligned} \begin{bmatrix} U_2 \\ 0 \end{bmatrix} S \begin{bmatrix} V_1 & 0 \\ 0 & I \end{bmatrix} &= \begin{bmatrix} 0 & (I - AA^D)B \\ 0 & 0 \end{bmatrix} R, \\ \begin{bmatrix} U_1 & 0 \\ 0 & I \end{bmatrix} T \begin{bmatrix} V_2 & 0 \end{bmatrix} &= R \begin{bmatrix} 0 & 0 \\ C(I - AA^D) & 0 \end{bmatrix}, \\ \begin{bmatrix} U_2 \\ 0 \end{bmatrix} N^i \begin{bmatrix} V_2 & 0 \end{bmatrix} &= \begin{bmatrix} (I - AA^D)A^i & 0 \\ 0 & 0 \end{bmatrix}. \end{aligned}$$

Equation (3.7) immediately follows from (3.11) and (3.12), where R is given by (3.11).

We remark that what we have derived so far will be used in the next section.

It suffices to show that $\text{Ind}(F) = 0$ and that (3.8) follows from (3.11). As a matter of fact, the Schur complement of F in (3.5), $D - Q_1 \Sigma^{-1} P_1 = D - CA^D B = Z$, is nonsingular by hypothesis. Then F is nonsingular and its index is zero. Applying Theorem 1.1 to F , we have

$$(3.13) \quad F^{-1} = \begin{bmatrix} \Sigma^{-1} + \Sigma^{-1} P_1 Z^{-1} Q_1 \Sigma^{-1} & -\Sigma^{-1} P_1 Z^{-1} \\ -Z^{-1} Q_1 \Sigma^{-1} & Z^{-1} \end{bmatrix}.$$

Substituting (3.13) into (3.11), we obtain

$$R = \begin{bmatrix} U_1(\Sigma^{-1} + \Sigma^{-1} P_1 Z^{-1} Q_1 \Sigma^{-1}) V_1 & -U_1 \Sigma^{-1} P_1 Z^{-1} \\ -Z^{-1} Q_1 \Sigma^{-1} V_1 & Z^{-1} \end{bmatrix},$$

which implies (3.8) from identities (1.10). This completes the proof. $\square$

Remark. We remark that (3.6) is weaker than (1.12). Furthermore, we give a geometric interpretation in the case of $k = \text{Ind}(A) \geq 1$. Equation (1.12) in Theorem 1.1 means that $R(B) \subset R(A^k)$ and $N(A^k) \subset N(C)$, while (3.6) in Theorem 3.1 means that $R(B) \subset R(A^{k-1})$ and that the projection of $R(B)$ on $N(A^k)$ along $R(A^k)$ is in $N(C)$.

Two corollaries are easily derived from Theorem 3.1 by taking conjugate transposes. If $A(I - AA^D)B = 0$ of (3.6) is replaced with $C(I - AA^D)A = 0$ (which means that $N(A^{k-1}) \subset N(C)$), we have the following corollary.

COROLLARY 3.2. *Let $M = \begin{bmatrix} A & B \\ C & D \end{bmatrix}$, where A and D are square matrices. If*

$$C(I - AA^D)B = 0, \quad C(I - AA^D)A = 0,$$

and $Z = D - CA^D B$ is nonsingular, then $\text{Ind}(M) \leq \text{Ind}(A) + 1$ and

$$M^D = \left(I + \sum_{i=0}^{k-1} \begin{bmatrix} 0 & A^i(I - AA^D)B \\ 0 & 0 \end{bmatrix} R^{i+1} \right) R \left(I + R \begin{bmatrix} 0 & 0 \\ C(I - AA^D) & 0 \end{bmatrix} \right),$$

where $k = \text{Ind}(A)$ and R is given by (3.8).

If an extra condition $C(I - AA^D)A = 0$ is fulfilled in Theorem 3.1, then we have a simpler formula for M^D .

COROLLARY 3.3. *Let $M = \begin{bmatrix} A & B \\ C & D \end{bmatrix}$, where A and D are square matrices. If*

$$(3.14) \quad C(I - AA^D)B = 0, \quad A(I - AA^D)B = 0, \quad C(I - AA^D)A = 0,$$

and $Z = D - CA^D B$ is nonsingular, then $\text{Ind}(M) \leq \text{Ind}(A) + 1$ and

$$(3.15) \quad M^D = \begin{bmatrix} I - (I - AA^D)BZ^{-1}CA^D & (I - AA^D)BZ^{-1} \\ 0 & I \end{bmatrix} \times \begin{bmatrix} A^D + A^D BZ^{-1}CA^D & -A^D BZ^{-1} \\ -Z^{-1}CA^D & Z^{-1} \end{bmatrix} \begin{bmatrix} I - A^D BZ^{-1}C(I - AA^D) & 0 \\ Z^{-1}C(I - AA^D) & I \end{bmatrix}.$$

It is trivial to verify that Theorem 3.1 and Corollaries 3.2 and 3.3 are extensions of Theorem 1.1. They imply many results related to the Drazin inverse of a partitioned matrix in [15].

4. Zero generalized Schur complement. In this section more representations for M^D , which are extensions of Theorem 1.2, are developed in a compact form similar to (3.7) in Theorem 3.1.

We first introduce the weighted Drazin inverse due to Cline and Greville in [6]. For two rectangular matrices $G \in C^{m \times n}$ and $W \in C^{n \times m}$, the *weighted* Drazin inverse of G with a weight matrix W , denoted by $G_{d,W}$, is defined as

$$(4.1) \quad G_{d,W} = [(GW)^D]^2 G.$$

$G_{d,W}$ is the unique matrix satisfying the following three matrix equations:

$$(GW)^k = (GW)^{k+1} G_{d,W} W, \quad G_{d,W} = G_{d,W} W G W G_{d,W}, \quad G W G_{d,W} = G_{d,W} W G,$$

where $k = \text{Ind}(GW)$. If G is a square matrix and $W = I$ is the identity matrix with the same dimensions as G , then the *weighted* Drazin inverse of G with the weight matrix I is the Drazin inverse of G , i.e.,

$$G_{d,I} = G^D.$$

Thus, the *weighted* Drazin inverse of G with a weight matrix W is an extension of the Drazin inverse of a square matrix defined in section 1.

THEOREM 4.1. Let $M = \begin{bmatrix} A & B \\ C & D \end{bmatrix}$, where A and D are square matrices, and let

$$W = AA^D + A^D B C A^D.$$

If the generalized Schur complement $D - CA^D B$ is equal to 0, and

$$(4.2) \quad C(I - AA^D)B = 0, \quad A(I - AA^D)B = 0,$$

then

$$(4.3) \quad \text{Ind}(M) \leq \text{Ind}(AW) + \text{Ind}(A) + 2$$

and

$$(4.4) \quad M^D = \left(I + \begin{bmatrix} 0 & (I - AA^D)B \\ 0 & 0 \end{bmatrix} R_1 \right) R_1 \left(I + \sum_{i=0}^{k-1} R_1^{i+1} \begin{bmatrix} 0 & 0 \\ C(I - AA^D)A^i & 0 \end{bmatrix} \right),$$

where $k = \text{Ind}(A)$ and

$$(4.5) \quad R_1 = \begin{bmatrix} I \\ CA^D \end{bmatrix} A_{d,W} \begin{bmatrix} I & A^D B \end{bmatrix},$$

where $A_{d,w} = [(AW)^D]^2 A$ is the weighted Drazin inverse of A with weight matrix W .

Proof. Let us recall (3.4) and (3.5) in section 3,

$$M_1 = \begin{bmatrix} F & \tilde{Q}_2 \\ \tilde{P}_2 & N \end{bmatrix},$$

where

$$F = \begin{bmatrix} \Sigma & P_1 \\ Q_1 & D \end{bmatrix}, \quad \tilde{P}_2 = \begin{bmatrix} 0 & P_2 \end{bmatrix}, \quad \text{and} \quad \tilde{Q}_2 = \begin{bmatrix} 0 \\ Q_2 \end{bmatrix}.$$

With the remark following (3.12), we know that if (4.2) holds, then

$$(4.6) \quad \text{Ind}(M) \leq \text{Ind}(F) + \text{Ind}(A) + 1$$

and (4.4) is valid for R_1 given by

$$(4.7) \quad R_1 = \begin{bmatrix} U_1 & 0 \\ 0 & I \end{bmatrix} F^D \begin{bmatrix} V_1 & 0 \\ 0 & I \end{bmatrix}.$$

It suffices to derive (4.3) and (4.5) under the assumption of $Z = D - CA^D B = 0$.

We first show that (4.5) follows from (4.7). It is known that Σ is nonsingular and $\text{rank}(F) = \text{rank}(\Sigma)$ since $D - Q_1 \Sigma^{-1} P_1 = D - CA^D B = 0$ by our hypothesis. It follows from [4, Thms. 7.7.5 and 7.7.6] that

$$(4.8) \quad \text{Ind}(F) = \text{Ind}[\Sigma(I + \Sigma^{-1} P_1 Q_1 \Sigma^{-1})] + 1$$

and

$$(4.9) \quad F^D = \begin{bmatrix} I \\ Q_1 \Sigma^{-1} \end{bmatrix} [(\Sigma H)^2]^D \Sigma \begin{bmatrix} I & \Sigma^{-1} P_1 \end{bmatrix},$$

where

$$H = I + \Sigma^{-1} P_1 Q_1 \Sigma^{-1}.$$

Notice that

$$\begin{aligned} \begin{bmatrix} U_1 & 0 \\ 0 & I \end{bmatrix} \begin{bmatrix} I \\ Q_1 \Sigma^{-1} \end{bmatrix} &= \begin{bmatrix} I \\ CA^D \end{bmatrix} U_1, \\ \begin{bmatrix} I & \Sigma^{-1} P_1 \end{bmatrix} \begin{bmatrix} V_1 & 0 \\ 0 & I \end{bmatrix} &= V_1 \begin{bmatrix} I & A^D B \end{bmatrix}. \end{aligned}$$

Applying (1.10) yields

$$\begin{aligned} U_1 \Sigma H V_1 &= U_1 \Sigma V_1 + U_1 V_1 B C U_1 \Sigma^{-1} V_1 \\ &= (U_1 \Sigma V_1 + U_2 N V_2) U_1 V_1 + U_1 V_1 B C U_1 V_1 A^D U_1 V_1 \\ &= A(AA^D + A^D B C A^D) = AW. \end{aligned}$$

With an application of Lemma 2.4, we obtain

$$(4.10) \quad U_1 [(\Sigma H)^2]^D V_1 = [U_1(\Sigma H)^D V_1]^2 = [(U_1 \Sigma H V_1)^D]^2 = [(AW)^D]^2.$$

Then it follows from (4.7) and (4.9) that

$$R_1 = \begin{bmatrix} I \\ CA^D \end{bmatrix} [(AW)^D]^2 A^2 A^D \begin{bmatrix} I & A^D B \end{bmatrix}.$$

We have from (4.10) and Lemma 2.4 that

$$[(AW)^D]^2 AA^D = U_1 [(\Sigma H)^2]^D V_1 U_1 V_1 = U_1 [(\Sigma H)^2]^D V_1 = [(AW)^D]^2.$$

Thus, (4.5) follows immediately.

Finally, we show (4.3) holds for the index of M . By (4.6) and (4.8), we have

$$\text{Ind}(M) \leq \text{Ind}[\Sigma(I + \Sigma^{-1} P_1 Q_1 \Sigma^{-1})] + \text{Ind}(A) + 2.$$

With an application of Lemma 2.4,

$$\begin{aligned} \text{Ind}[\Sigma(I + \Sigma^{-1} P_1 Q_1 \Sigma^{-1})] &\leq \text{Ind}[U_1 \Sigma V_1 U_1 (I + \Sigma^{-1} P_1 Q_1 \Sigma^{-1}) V_1] \\ &= \text{Ind}[A(AA^D + A^D B C A^D)] = \text{Ind}(AW), \end{aligned}$$

which immediately gives (4.3). This completes the proof. $\square$

Remark. The constraint on $Z = D - CA^D B$ in either Theorem 3.1 or Theorem 4.1 (i.e., Z is either nonsingular or zero) is essential in this paper. It is an open problem to find an explicit formula for M^D in terms of A , B , C , D , A^{-1} , and D^D in the case where A is nonsingular and $Z = D - CA^{-1}B$ is a nonzero singular matrix.

Two corollaries of Theorem 4.1 are stated without proof. If we replace $A(I - AA^D)B = 0$ (4.2) in Theorem 4.1 with $C(I - AA^D)A = 0$, we obtain the following corollary by taking conjugate transposes.

COROLLARY 4.2. *Let $M = \begin{bmatrix} A & B \\ C & D \end{bmatrix}$, where A and D are square matrices, and let $W = AA^D + A^D B C A^D$. If the generalized Schur complement $Z = D - CA^D B$ is equal to zero and*

$$C(I - AA^D)B = 0, \quad C(I - AA^D)A = 0,$$

then $\text{Ind}(M) \leq \text{Ind}(AW) + \text{Ind}(A) + 2$ and

$$M^D = \left(I + \sum_{i=0}^{k-1} \begin{bmatrix} 0 & A^i(I - AA^D)B \\ 0 & 0 \end{bmatrix} R_1^{i+1} \right) R_1 \left(I + R_1 \begin{bmatrix} 0 & 0 \\ C(I - AA^D) & 0 \end{bmatrix} \right),$$

where $k = \text{Ind}(A)$ and R_1 is given by (4.5) in Theorem 4.1.

If the condition $C(I - AA^D)A = 0$ is added to Theorem 4.1, then we have a simpler formula for M^D .

COROLLARY 4.3. *Let $M = \begin{bmatrix} A & B \\ C & D \end{bmatrix}$, where A and D are square matrices, and let $W = AA^D + A^D B C A^D$. If $Z = D - CA^D B = 0$ and*

$$C(I - AA^D)B = 0, \quad A(I - AA^D)B = 0, \quad C(I - AA^D)A = 0,$$

then $\text{Ind}(M) \leq \text{Ind}(AW) + \text{Ind}(A) + 2$ and

$$\begin{aligned} M^D &= \begin{bmatrix} I + (I - AA^D)BCA^D A_{d,W} & (I - AA^D)BCA^D A_{d,W} A^D B \\ 0 & I \end{bmatrix} \\ &\quad \times R_1 \begin{bmatrix} I + A_{d,W} A^D BC(I - AA^D) & 0 \\ CA^D A_{d,W} A^D BC(I - AA^D) & I \end{bmatrix}, \end{aligned}$$

where R_1 is given by (4.5) in Theorem 4.1.

It is trivial to verify that Theorem 4.1 and Corollaries 4.2 and 4.3 are extensions of Theorem 1.2. They imply many results related to the Drazin inverse of a partitioned matrix as given in [15].

5. Examples. Three examples are given to illustrate the new representations for the Drazin inverse of a 2×2 block matrix in Theorems 3.1 and 4.1.

Example 5.1. We first give an example where it would be helpful to find the Drazin inverse of a certain 2×2 block matrix. Consider a second-order system $Ex''(t) + Fx'(t) + Gx(t) = 0$ in (1.3), where

$$E = \begin{bmatrix} 1 & 3 \\ 2 & 6 \end{bmatrix}, \quad F = \begin{bmatrix} 1 & 2 \\ 3 & 4 \end{bmatrix}, \quad \text{and} \quad G = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}.$$

Choose $\lambda = 1$. Then $\lambda^2 E + \lambda F + G = \begin{bmatrix} 3 & 5 \\ 5 & 11 \end{bmatrix}$ is nonsingular. It follows from (1.5) and $\mu = 1$ in (1.6) that in order to have explicit solutions of (1.4) we need to find an explicit expression for the Drazin inverse of the 2×2 block matrix

$$\hat{E} = \begin{bmatrix} I & -I \\ \tilde{E} & \tilde{F} \end{bmatrix}^{-1} \begin{bmatrix} 0 & -I \\ \tilde{E} & \tilde{F} \end{bmatrix} = \begin{bmatrix} A & B \\ C & D \end{bmatrix},$$

where

$$A = \begin{bmatrix} \frac{1}{7} & \frac{3}{7} \\ 0 & 0 \end{bmatrix}, \quad B = \begin{bmatrix} -\frac{19}{7} & \frac{11}{7} \\ 1 & -1 \end{bmatrix}, \quad C = \begin{bmatrix} \frac{1}{7} & \frac{3}{7} \\ 0 & 0 \end{bmatrix}, \quad D = \begin{bmatrix} -\frac{12}{7} & \frac{11}{7} \\ 1 & 0 \end{bmatrix}.$$

It can be verified that $Z = D - CA^D B = \begin{bmatrix} -2 & 3 \\ 1 & 0 \end{bmatrix}$ is nonsingular. Theorem 1.1 fails to apply since $(I - AA^D)B \neq 0$. However, (3.14) is satisfied. With Corollary 3.3, (3.15) gives the exact value of the explicit expression of $(\hat{E})^D$,

$$(\hat{E})^D = \begin{bmatrix} \frac{1}{3} & 1 & -\frac{1}{3} & \frac{1}{3} \\ \frac{10}{9} & \frac{10}{3} & \frac{11}{9} & \frac{13}{9} \\ 0 & 0 & 0 & 1 \\ -\frac{1}{3} & -1 & \frac{1}{3} & \frac{2}{3} \end{bmatrix}.$$

Therefore, the solutions of (1.4) can be obtained by applying (1.2) and, consequently, the solution of the given second-order system (1.3) is available.

Example 5.2. Consider a 2×2 block matrix $M = \begin{bmatrix} A & B \\ C & D \end{bmatrix}$, where

$$A = \begin{bmatrix} 0 & 1 & -1 & 1 \\ 0 & 1 & -1 & 1 \\ 0 & 1 & -1 & 1 \\ 0 & 0 & 0 & 1 \end{bmatrix}, \quad B = \begin{bmatrix} 4 & 5 \\ 3 & 4 \\ 3 & 4 \\ 2 & 3 \end{bmatrix},$$

$$C = \begin{bmatrix} -1 & 4 & -2 & 0 \\ -2 & 3 & 1 & 0 \end{bmatrix}, \quad D = \begin{bmatrix} 3 & 3 \\ 5 & 7 \end{bmatrix}.$$

It is calculated that

$$A^D = \begin{bmatrix} 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 1 \end{bmatrix}, \quad I - AA^D = \begin{bmatrix} 1 & 0 & 0 & -1 \\ 0 & 1 & 0 & -1 \\ 0 & 0 & 1 & -1 \\ 0 & 0 & 0 & 0 \end{bmatrix}.$$

Then $D - CA^D B = \begin{bmatrix} 1 & 0 \\ 1 & 1 \end{bmatrix}$ is nonsingular and (3.6) is satisfied. Corollaries 3.2 and 3.3 fail to apply because $C(I - AA^D)A \neq 0$. It is verified that $\text{Ind}(A) = 2$ and $\text{Ind}(M) = 3$ satisfying Theorem 3.1. Equation (3.7) gives the exact value of M^D ,

$$M^D = \begin{bmatrix} -106 & 734 & -522 & -126 & -5 & 11 \\ -37 & 255 & -181 & -44 & -2 & 4 \\ -37 & 255 & -181 & -44 & -2 & 4 \\ 32 & -224 & 160 & 38 & 1 & -3 \\ -6 & 47 & -35 & -7 & 1 & 0 \\ -5 & 31 & -21 & -6 & -1 & 1 \end{bmatrix}.$$

Example 5.3. Consider the same matrix M as in Example 5.2 but with $D = CA^D B = \begin{bmatrix} 2 & 3 \\ 4 & 6 \end{bmatrix}$. In this example, $\text{Ind}(M) = 4$ and $\text{Ind}(AW) + \text{Ind}(A) + 2 = 1 + 2 + 2 = 5$, so the inequality (4.3) is strict here. R_1 of (4.5) is given by

$$R_1 = \frac{1}{81} \begin{bmatrix} 0 & 0 & 0 & 1 & 2 & 3 \\ 0 & 0 & 0 & 1 & 2 & 3 \\ 0 & 0 & 0 & 1 & 2 & 3 \\ 0 & 0 & 0 & 1 & 2 & 3 \\ 0 & 0 & 0 & 1 & 2 & 3 \\ 0 & 0 & 0 & 2 & 4 & 6 \end{bmatrix}.$$

Equation (4.4) gives the exact value of M^D ,

$$M^D = \frac{1}{19683} \begin{bmatrix} -360 & 805 & -85 & 45 & 810 & 1215 \\ -288 & 644 & -68 & 36 & 648 & 972 \\ -288 & 644 & -68 & 36 & 648 & 972 \\ -216 & 483 & -51 & 27 & 486 & 729 \\ -216 & 483 & -51 & 27 & 486 & 729 \\ -432 & 966 & -102 & 54 & 972 & 1458 \end{bmatrix}.$$

Acknowledgments. The authors would like to thank Professor Inderjit Dhillon and two anonymous referees for their helpful and constructive comments.

REFERENCES

- [1] A. BEN-ISRAEL AND T. N. E. GREVILLE, *Generalized Inverse: Theory and Applications*, 2nd ed., Springer, New York, 2003.
- [2] S. L. CAMPBELL, *Singular Systems of Differential Equations*, Pitman, London, 1980.
- [3] S. L. CAMPBELL, *The Drazin inverse and systems of second order linear differential equations*, Linear and Multilinear Algebra, 14 (1983), pp. 195–198.
- [4] S. L. CAMPBELL AND C. D. MEYER, *Generalized Inverse of Linear Transformations*, Pitman, London, 1979.
- [5] X. CHEN AND R. E. HARTWIG, *The group inverse of a triangular matrix*, Linear Algebra Appl., 237/238 (1996), pp. 97–108.
- [6] R. E. CLINE AND T. N. E. GREVILLE, *A Drazin inverse for rectangular matrices*, Linear Algebra Appl., 29 (1980), pp. 53–62.
- [7] D. S. DJORDJEVIĆ AND P. S. STANIMIROVIĆ, *On the generalized Drazin inverse and generalized resolvent*, Czechoslovak Math. J., 51(126) (2001), pp. 617–634.
- [8] R. E. HARTWIG AND J. M. SHOAF, *Group inverses and Drazin inverses of bidiagonal and triangular Toeplitz matrices*, J. Austral. Math. Soc. Ser. A, 24 (1977), pp. 10–34.
- [9] R. E. HARTWIG, *The group inverse of a block triangular matrix*, in Current Trends in Matrix Theory, F. Uhlig and R. Grone, eds., North-Holland, New York, 1987, pp. 137–146.

- [10] R. E. HARTWIG, G. WANG, AND Y. WEI, *Some additive results on Drazin inverse*, Linear Algebra Appl., 322 (2001), pp. 207–217.
- [11] X. LI AND Y. WEI, *An improvement on the perturbation of the group inverse and oblique projection*, Linear Algebra Appl., 338 (2001), pp. 53–66.
- [12] J. MIAO, *Results of the Drazin inverse of block matrices*, J. Shanghai Normal University, 18 (1989), pp. 25–31 (in Chinese).
- [13] C. D. MEYER, JR., AND N. J. ROSE, *The index and the Drazin inverse of block triangular matrices*, SIAM J. Appl. Math., 33 (1977), pp. 1–7.
- [14] M. NEUMANN AND H. J. WERNER, *Nonnegative group inverse*, Linear Algebra Appl., 151 (1991), pp. 85–96.
- [15] Y. WEI, *Expressions for the Drazin inverse of a 2×2 block matrix*, Linear and Multilinear Algebra, 45 (1998), pp. 131–146.
- [16] Y. WEI AND X. LI, *An improvement on perturbation bounds for the Drazin inverse*, Numer. Linear Algebra Appl., 10 (2003), pp. 563–575.
- [17] Y. WEI, X. LI, AND F. BU, *A perturbation bound of the Drazin inverse of a matrix by separation of simple invariant subspaces*, SIAM J. Matrix Anal. Appl., 27 (2005), pp. 72–81.
- [18] N. ZHANG AND Y. WEI, *Solving EP singular linear systems*, Internat. J. Comput. Math., 81 (2004), pp. 1395–1405.

SOME REMARKS ON THE ELMAN ESTIMATE FOR GMRES*

B. BECKERMANN[†], S. A. GOREINOV[‡], AND E. E. TYRTYSHNIKOV[‡]

Abstract. Starting from a GMRES error estimate proposed by Elman in terms of the ratio of the smallest eigenvalue of the hermitian part and the norm of some nonsymmetric matrix, we propose some asymptotically tighter bound in terms of the same ratio. Here we make use of a recent deep result of Crouzeix and others on the norm of functions of matrices.

Key words. GMRES, nonsymmetric systems, error estimates, field of values

AMS subject classifications. 65F10, 65N22

DOI. 10.1137/040618849

1. Introduction. A popular method for solving nonhermitian systems $Ax = b$ is given by GMRES [13]. Provided that A has a positive definite hermitian part $(A + A^*)/2$, Elman [9] (see also [8]) pointed out the following upper bound for the norm of the k th residual r_k of GMRES for every $k \geq 0$:

$$(1.1) \quad \frac{\|r_k\|}{\|r_0\|} \leq \sin^k(\beta), \quad \text{where } \cos(\beta) := \frac{\lambda_{\min}((A + A^*)/2)}{\|A\|}$$

with $\beta \in [0, \pi/2)$. Here and hereafter we use $\|\cdot\|$ to denote the Euclidean vector norm or spectral matrix norm. Recall that the field of values (or numerical range)

$$W(A) := \{(Ay, y) : y \in \mathbb{C}^n, \|y\| = 1\}$$

of a matrix is bounded by the rectangle defined by the extremal eigenvalues of the hermitian and the skew-hermitian parts of A . Hence $\lambda_{\min}((A + A^*)/2)$ bounds from below the distance between the origin and $W(A)$. More generally, since $W(A)$ is convex by the Hausdorff theorem, one may show in the case $0 \notin W(A)$ that there is a $t \in \mathbb{R}$ with $\text{dist}(0, W(A)) = \lambda_{\min}((e^{it}A + (e^{it}A)^*)/2)$, and $t \in \{0, \pi\}$ for real matrices A . Since $\|r_k\|$ does not change after multiplying $Ax = b$ by some number of modulus 1, we may rewrite in the case $0 \notin W(A)$ the Elman bound (1.1) as

$$(1.2) \quad k \geq 0 : \quad \frac{\|r_k\|}{\|r_0\|} \leq \sin^k(\beta), \quad \cos(\beta) = \frac{\text{dist}(0, W(A))}{\|A\|},$$

where we recall that (1.2) can be sharper than (1.1).

Estimates (1.1) or (1.2) are obtained by iterating the inequality for $k = 1$ corresponding to a one-dimensional minimization problem, which should allow for some

*Received by the editors November 13, 2004; accepted for publication (in revised form) by D. B. Szyld July 5, 2005; published electronically January 6, 2006.

<http://www.siam.org/journals/simax/27-3/61884.html>

[†]Laboratoire Paul Painlevé UMR 8524 (ANO-EDP), UFR Mathématiques-M3, UST Lille, F-59655 Villeneuve d'Ascq CEDEX, France (bbecker@math.univ-lille1.fr). The research of this author was supported in part by INTAS network NeCCA 03-51-6637, and in part by the Ministry of Science and Technology (MCYT) of Spain and the European Regional Development Fund (ERDF) through grant BFM2001-3878-C02-02.

[‡]Institute of Numerical Mathematics, Russian Academy of Sciences, Gubkina Street, 8, Moscow, 119991, Russia (serge@inm.ras.ru, tee@inm.ras.ru). The research of these authors was supported in part by RFBR grant 02-01-00590a.

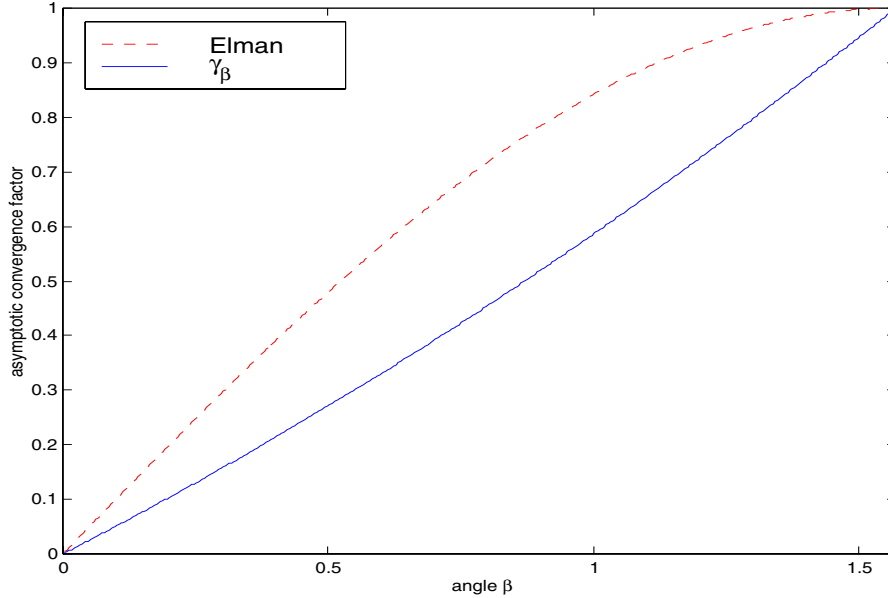


FIG. 1. Elman's asymptotic convergence factor $\sin(\beta)$ (dashed) versus the new asymptotic convergence factor γ_β of Theorem 2.1 (solid).

improvements (compare, for instance, the approach in [7, Theorem 6.1] in terms of angles between subspaces). Here we propose some asymptotically sharper bounds only in terms of the above angle $\beta \in (0, \pi/2)$, which to our knowledge seems to be new.

2. Main result.

THEOREM 2.1. *Let A be a matrix with $0 \notin W(A)$, and let $\beta \in (0, \pi/2)$ be as in (1.2). Then for the k th relative residual, $k \geq 1$, of GMRES we have*

$$(2.1) \quad \frac{\|r_k\|}{\|r_0\|} \leq (2 + 2/\sqrt{3}) (2 + \gamma_\beta) \gamma_\beta^k,$$

where

$$\gamma_\beta := 2 \sin\left(\frac{\beta}{4 - 2\beta/\pi}\right) < \sin(\beta).$$

All convergence bounds (1.1), (1.2), (2.1) are of the form $C\gamma^k$, where we will call C the constant factor and γ the asymptotic convergence factor. We have drawn in Figure 1 the two different asymptotic convergence factors of (1.1) and of Theorem 2.1. Especially for β close to $\pi/2$ (i.e., the critical case where the origin is close to the field of values), the asymptotic convergence factor of Theorem 2.1 is clearly more interesting. However, the constant factor of (2.1) (which can be shown to be not optimum) slightly deteriorates the sharpness of the new convergence bound; see Figure 2. Notice that bounds of type (2.1) do not capture the range of superlinear convergence of GMRES.

Before entering in the proof of Theorem 2.1, let us briefly recall some well-known facts on the convergence of GMRES. Starting from the observation

$$\frac{\|r_k\|}{\|r_0\|} \leq \min\{\|p(A)\| : p \text{ a polynomial of degree } \leq k, p(0) = 1\},$$

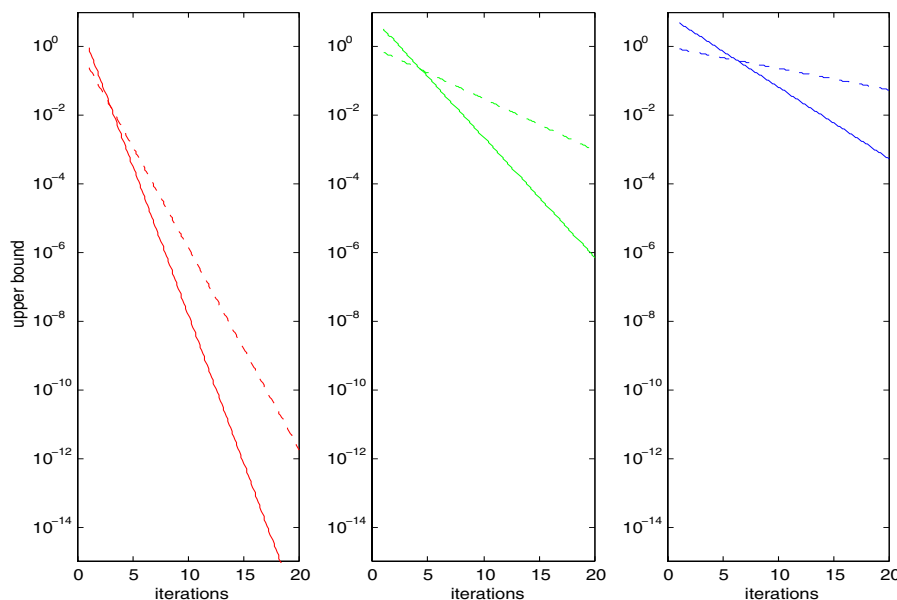


FIG. 2. Upper bounds for the relative residuals of the k th iterate ($k = 1, \dots, 20$) of GMRES for $\beta = \pi/12$, $\beta = \pi/4$, $\beta = \pi/3$ (from left to right); the curve corresponding to Elman's bound is dashed, and the one of Theorem 2.1 is solid.

there are many classical GMRES estimates not involving special properties of the right-hand side b of the system; see, for instance, [14, Chapter 6.11] or [10, Chapter 3]. Here the quantity on the right is usually estimated in terms of a polynomial extremal problem: For a compact set K , consider the constrained Chebyshev approximation problem

$$E_k(K) := \min\{\|p\|_K : p \text{ a polynomial of degree } \leq k, p(0) = 1\},$$

where by $\|\cdot\|_K$ we denote the maximum norm on K . For instance, for diagonalizable A one immediately obtains the classical bound

$$(2.2) \quad k \geq 0 : \quad \frac{\|r_k\|}{\|r_0\|} \leq \|V\| \|V^{-1}\| E_k(\sigma(A))$$

in terms of the spectrum $\sigma(A)$ and the matrix of eigenvectors V of A . Other more sophisticated estimates are based, for instance, on the pseudospectrum, but in general the shape of these sets is difficult to predict. A third group of estimates is obtained via the field of values, starting perhaps with a paper of Eiermann [6]; see also [10]. These estimates are based on the observation that, given a convex set $K \neq \mathbb{C}$, there exists a constant $C(K) < \infty$ such that for all matrices A and for all rational functions f having no pole in K there holds

$$(2.3) \quad W(A) \subset K \quad \implies \quad \|f(A)\| \leq C(K) \|f\|_K.$$

Recently, Crouzeix and his coworkers [1, 3, 4] gave quite deep results concerning (2.3). The existence of such a finite constant in (2.3) depending only on the set K was established by B. Delyon and F. Delyon in [5]. In [1, Corollary 2.3], it was shown

that for the optimal constant in (2.3) (also denoted by $C(K)$) one has

$$(2.4) \quad C(K) \leq 2 + \pi + \int_0^{2\pi} \frac{|\rho'(t)|}{\rho(t)} dt,$$

where $[0, 2\pi] \ni t \mapsto z_0 + \rho(t)e^{it} \in \partial K$, $\rho(t) > 0$, is any polar parametrization of the boundary of K for some z_0 in the interior of K (the case of a segment K is trivial; here $aI + bA$ is hermitian for some $a, b \in \mathbb{C}$ and thus $C(K) = 1$). The authors also give improved estimates for particular K , e.g., for a sector [4]

$$(2.5) \quad C(S_\alpha) \leq 2 + 2/\sqrt{3}, \quad S_\alpha = \{z \in \mathbb{C} : 0 \leq \arg(z) \leq \alpha\}, \quad 0 \leq \alpha \leq \pi.$$

In a recent manuscript [3], Crouzeix showed that

$$(2.6) \quad C(K) \leq 33.75$$

for any convex compact set K , independently of its shape. It is a conjecture of Crouzeix that $C(K)$ in (2.3) can be replaced by the number 2.

Let us return to the proof of Theorem 2.1. By possibly multiplying A with some complex number of modulus 1, we may suppose without loss of generality that the element of $W(A)$ closest to 0 is real positive, and thus

$$W(A) \subset \{z : \operatorname{Re}(z) \geq \|A\| \cos(\beta)\},$$

with β as in (1.2). Define K_β to be the (convex) intersection between the closed unit disk and the half-plane $\{\operatorname{Re}(z) \geq \cos(\beta)\}$. Since $W(A) \subset \{|z| \leq \|A\|\}$, we see that $W(A) \subset \|A\| K_\beta$. Moreover, by constructing block-diagonal matrices with blocks

$$B(\phi) = \begin{bmatrix} \cos(\phi) & \sin(\phi) \\ -\sin(\phi) & \cos(\phi) \end{bmatrix}, \quad 0 \leq \phi \leq \beta,$$

we find unitary matrices $A_n \in \mathbb{R}^{2^n \times 2^n}$ with $W(A_n) \subset W(A_{n+1}) \subset K_\beta$, and the closure of $\bigcup_n W(A_n)$ coinciding with K_β . Hence in general the relation $W(A) \subset \|A\| K_\beta$ cannot be improved without further knowledge on A .

We have the following result for our constrained Chebyshev approximation problem.

LEMMA 2.2. *There holds for any $k \geq 1$ for any $\beta \in (0, \pi/2)$*

$$\gamma_\beta^k < E_k(K_\beta) \leq \min \left\{ 2 + \gamma_\beta, \frac{2}{1 - \gamma_\beta^{k+1}} \right\} \gamma_\beta^k,$$

with $\gamma_\beta = 2 \sin(\frac{\beta}{4 - 2\beta/\pi})$ as in Theorem 2.1.

Proof. Let K be some convex compact set containing at least two elements, $0 \notin K$, and denote by ϕ the Riemann conformal map mapping from $\overline{\mathbb{C}} \setminus K$ onto the exterior of the closed unit disk, with $\phi(\infty) = \infty$. In the first part of the proof we claim that

$$(2.7) \quad k \geq 1 : \quad \gamma^k \leq E_k(K) \leq \min \left\{ 2 + \gamma, \frac{2}{1 - \gamma^{k+1}} \right\} \gamma^k, \quad \gamma := 1/|\phi(0)|.$$

The inequality $\gamma^k \leq E_k(K)$ follows by applying the maximum principle to the function p/ϕ^k for an arbitrary polynomial p of degree $\leq k$; see also the classical Bernstein–Walsh inequality [12]. Notice also that, again by the maximum principle, we have

$\gamma^k = E_k(K)$ for some $k > 0$ if and only if ϕ^k is a polynomial of degree k or, in other words, if K is a lemniscate, which is certainly not true for our set K_β .

For the other inequality of (2.7) we need a good candidate p of degree k . Consider the Faber polynomial F_k being the polynomial part of the Laurent expansion of ϕ^k at ∞ ; see, e.g., [15, 16]. It is shown in [11, Theorem 2] for general convex sets K that

$$(2.8) \quad \delta_k := \|F_k - \phi^k\|_{\partial K} \leq 1.$$

From the maximum principle applied to $\phi F_k - \phi^{k+1}$ we know that $|\phi(0)| |F_k(0) - \phi(0)^k| < \|\phi F_k - \phi^{k+1}\|_{\partial K} \leq \delta_k$, and hence for the polynomial depending on some parameter $v \in [0, 1]$,

$$p_v(z) = F_k(z) + v(\phi(0)^k - F_k(0)),$$

we obtain $|p_v(0)| \geq |\phi(0)|^k - (1-v)\delta_k/|\phi(0)|$ and

$$\|p_v\|_K = \|p_v\|_{\partial K} \leq \|\phi^k\|_{\partial K} + \|\phi^k - F_k\|_{\partial K} + v|\phi(0)^k - F_k(0)| \leq 1 + \delta_k + v\delta_k/|\phi(0)|,$$

and thus

$$E_k(K_\beta) \leq \min_{v \in [0,1]} \frac{\|p_v\|_K}{|p_v(0)|} \leq \min_{v \in [0,1]} \gamma^k \frac{1 + \delta_k(1+v\gamma)}{1 - (1-v)\delta_k\gamma^{k+1}},$$

leading to the other inequality of (2.7).

Now the assertion of the lemma follows from (2.7) once we have shown that the Riemann conformal map ϕ for $K = K_\beta$ satisfies $1/|\phi(0)| = \gamma_\beta$. Indeed, this map can be explicitly constructed as a composition $\phi = T_3 \circ T_2 \circ T_1$, with

$$T_1(z) = \frac{z - e^{i\beta}}{z - e^{-i\beta}}, \quad T_2(z) = (e^{i(\pi-\beta)}z)^{\pi/(2\pi-\beta)}, \quad T_3(z) = \frac{z - \overline{T_2(1)}}{z - T_2(1)},$$

where T_1 maps the complement of K_β conformally onto $\{z \in \mathbb{C} : -\pi + \beta < \arg(z) < \pi\}$, T_2 maps $\{z \in \mathbb{C} : -\pi + \beta < \arg(z) < \pi\}$ conformally onto the upper half-plane $\{z \in \mathbb{C} : 0 < \arg(z) < \pi\}$, and finally T_3 maps the upper half-plane conformally onto the complement of the exterior of the closed unit disk. Finally, notice that

$$\frac{1}{|\phi(0)|} = \left| \frac{\exp\left(i\pi \frac{\pi+\beta}{2\pi-\beta}\right) - \exp\left(i\pi \frac{\pi-\beta}{2\pi-\beta}\right)}{\exp\left(i\pi \frac{\pi+\beta}{2\pi-\beta}\right) - \exp\left(-i\pi \frac{\pi-\beta}{2\pi-\beta}\right)} \right| = \frac{\sin\left(\frac{2\beta}{4-2\beta/\pi}\right)}{\sin\left(\frac{\pi}{2} + \frac{\beta}{4-2\beta/\pi}\right)} = \gamma_\beta,$$

as required for the assertion of Lemma 2.2. $\square$

By means of elementary computations one checks that the asymptotic convergence factor $\sin(\beta)$ found by Elman coincides with $E_1(K_\beta)$, which by Lemma 2.2 is larger than our constant γ_β . Thus (2.1) is asymptotically sharper than (1.1), (1.2); compare with Figures 1 and 2.

The second ingredient in our proof of Theorem 2.1 is the following observation.

LEMMA 2.3. *With $\beta \in (0, \pi/2)$ as in (1.2) there holds for any polynomial $p \neq 0$*

$$\|p(A)\| \leq (2 + 2/\sqrt{3}) \|p\|_K, \quad K := \|A\| K_\beta.$$

Proof. Choose $\alpha \in (\beta, \pi/2)$, and consider the linear fractional transformation

$$r(z) = \frac{\|A\| e^{i\alpha} - z}{z - \|A\| e^{-i\alpha}}.$$

Then $f := p \circ r^{-1}$ is a rational function with all poles at $-1 \notin S_\alpha$. Also, observe that $r(K) \subset r(\|A\| K_\alpha) = S_\alpha$. According to the Crouzeix results (2.3) and (2.5), for the claim of Lemma 2.3 it is sufficient to show the relation $W(r(A)) \subset S_\alpha$. For a vector $y \neq 0$, we define $\tilde{y} := (A - \|A\| e^{-i\alpha} I)^{-1} y \neq 0$, and consider

$$\begin{aligned} d &:= \frac{(r(A)y, y)}{(\tilde{y}, \tilde{y})} = \frac{((A - \|A\| e^{-i\alpha} I)^*(\|A\| e^{i\alpha} I - A)\tilde{y}, \tilde{y})}{(\tilde{y}, \tilde{y})} \\ &= -\|A\|^2 e^{2i\alpha} - \frac{\|A\tilde{y}\|^2}{\|\tilde{y}\|^2} + 2\|A\| e^{i\alpha} \operatorname{Re} \left(\frac{(A\tilde{y}, \tilde{y})}{(\tilde{y}, \tilde{y})} \right). \end{aligned}$$

Thus $\operatorname{Im}(d) \geq 2\|A\|^2 \sin(\alpha)[- \cos(\alpha) + \cos(\beta)] > 0$, and $\operatorname{Im}(e^{-i\alpha} d) = \sin(\alpha)[- \|A\|^2 + \|A\tilde{y}\|^2 / \|\tilde{y}\|^2] \leq 0$, implying that $W(r(A)) \subset S_\alpha$. $\square$

Since the quantity $E_k(K)$ is invariant under a scaling of the set K , we obtain from Lemma 2.3

$$\begin{aligned} &\min\{\|p(A)\| : p \text{ a polynomial of degree } \leq k, p(0) = 1\} \\ &\leq (2 + 2/\sqrt{3}) E_k(\|A\| K_\beta) = (2 + 2/\sqrt{3}) E_k(K_\beta), \end{aligned}$$

and Lemma 2.2 allows us to conclude the proof of Theorem 2.1.

Since the method of proof of (2.7) does not depend on the particular choice of the shape of the domain including the field of values, we have shown implicitly the following result complementary to Theorem 2.1 (compare with [3, section 9]).

COROLLARY 2.4. *Let K be some compact convex set not including the origin, and let A be some matrix with $W(A) \subset \|A\| K$. Then for the k th relative residual, $k \geq 1$, of GMRES we have*

$$\frac{\|r_k\|}{\|r_0\|} \leq [2 + \gamma] C(K) \gamma^k < [2 + \gamma] C(K) E_k(K),$$

where $C(K)$ can be chosen as in (2.4) or in (2.6), and $\gamma = 1/|\phi(0)| < 1$, ϕ denoting the Riemann conformal map mapping from $\overline{\mathbb{C}} \setminus K$ onto the exterior of the unit disk, with $\phi(\infty) = \infty$.

In particular, we get from Corollary 2.4 that the norm of the k th relative residual of GMRES is bounded above by $101.25 \gamma^k$, with $\gamma < 1$ as before.

Acknowledgments. This work was started during a visit of B. Beckermann at the Institute of Numerical Mathematics in Moscow. He gratefully acknowledges the hospitality of the Institute of Numerical Mathematics, as well as useful discussions with M. Eiermann which helped to improve the manuscript.

Note added in proof. After finishing this paper, one of the authors [2] was able to show an improvement of Corollary 2.4: For a convex compact set E and a matrix A with $W(A) \subset E$ it is shown in [2, Theorem 1] that $\|F_n(A)\| \leq 2$, where F_n denotes the n th Faber polynomial corresponding to the set E . As a consequence [2, Theorem 2 and Corollary 3], Corollary 2.4 remains valid after replacing the constant $C(K)$ by 1.

REFERENCES

- [1] C. BADEA, M. CROUZEIX, AND B. DELYON, *Convex domains and K -spectral sets*, Math. Z., 252 (2006), pp. 345–365.
- [2] B. BECKERMANN, *Image numérique, GMRES et polynômes de Faber*, C. R. Acad. Sci. Paris, Sér. I Math., 340 (2005), pp. 855–860.
- [3] M. CROUZEIX, *Numerical Range and Functional Calculus in Hilbert Space*, manuscript, Université de Rennes 1, Rennes, France, 2005; available online from <http://perso.univ-rennes1.fr/michel.crouzeix>.
- [4] M. CROUZEIX AND B. DELYON, *Some estimates for analytic functions of strip or sectorial operators*, Arch. Math. (Basel), 81 (2003), pp. 559–566.
- [5] B. DELYON AND F. DELYON, *Generalization of Von Neumann's spectral sets and integral representation of operators*, Bull. Soc. Math. France, 1 (1999), pp. 25–42.
- [6] M. EIERMANN, *Fields of values and iterative methods*, Linear Algebra Appl., 180 (1993), pp. 167–197.
- [7] M. EIERMANN AND O. G. ERNST, *Geometric aspects in the theory of Krylov subspace methods*, Acta Numer., 10 (2001), pp. 251–312.
- [8] S. C. EISENSTAT, H. C. ELMAN, AND M. H. SCHULTZ, *Variational iterative methods for non-symmetric systems of linear equations*, SIAM J. Numer. Anal., 20 (1983), pp. 345–357.
- [9] H. C. ELMAN, *Iterative Methods for Sparse Nonsymmetric Systems of Linear Equations*, Ph.D. thesis, Yale University, Department of Computer Science, New Haven, CT, 1982.
- [10] A. GREENBAUM, *Iterative Methods for Solving Linear Systems*, Frontiers Appl. Math. 17, SIAM, Philadelphia, 1997.
- [11] T. KÖVARI AND CH. POMMERENKE, *On Faber polynomials and Faber expansions*, Math. Z., 99 (1967), pp. 193–206.
- [12] E. M. NIKISHIN AND V. N. SOROKIN, *Rational Approximations and Orthogonality*, Nauka, Moscow, 1988 (in Russian); AMS Transl. Math. Monogr. 92, AMS, Providence, RI, 1991 (in English).
- [13] Y. SAAD AND M. H. SCHULTZ, *GMRES: A generalized minimal residual algorithm for solving nonsymmetric linear systems*, SIAM J. Sci. Statist. Comput., 7 (1986), pp. 856–869.
- [14] Y. SAAD, *Iterative Methods for Sparse Linear Systems*, PWS Publishing, Boston, 1996.
- [15] V. I. SMIRNOV AND N. A. LEBEDEV, *Functions of a Complex Variable*, MIT Press, Cambridge, MA, 1968.
- [16] P. K. SUETIN, *Polynomials orthogonal over a region and Bieberbach polynomials*, Proc. Steklov Inst. Math., 100 (1971), pp. 1–91 (in English); Tr. Mat. Inst. Steklov, 100 (1971), pp. 1–92 (in Russian).

AN ALGEBRAIC ANALYSIS OF A BLOCK DIAGONAL PRECONDITIONER FOR SADDLE POINT SYSTEMS*

GENE H. GOLUB[†], CHEN GREIF[‡], AND JAMES M. VARAH[‡]

Dedicated to Michael Saunders on the occasion of his 60th birthday

Abstract. We consider a positive definite block preconditioner for solving saddle point linear systems. An approach based on augmenting the (1,1) block while keeping its condition number small is described, and algebraic analysis is performed. Ways of selecting the parameters involved are discussed, and analytical and numerical observations are given.

Key words. saddle point linear systems, augmented Lagrangian, block preconditioning, positive semidefinite matrix, null space

AMS subject classifications. 65F10, 15A12

DOI. 10.1137/04060679X

1. Introduction. Consider the large and sparse saddle point linear system

$$(1.1) \quad \mathcal{H}u = b \equiv \begin{pmatrix} A & B \\ B^T & 0 \end{pmatrix} \begin{pmatrix} x \\ y \end{pmatrix} = \begin{pmatrix} c \\ d \end{pmatrix},$$

where A is $n \times n$ and B is $n \times m$, with $m \leq n$ (possibly $m \ll n$). Systems of the form (1.1) arise in many areas; for a recent comprehensive survey of numerical solution methods see [3]. The numerical treatment of these systems strongly depends on the properties of the (1,1) block, A , and we assume throughout (unless otherwise stated) that it is positive semidefinite. We also assume that $\mathcal{H}$ is nonsingular, which implies that B has full column rank.

It is possible to replace (1.1) by

$$(1.2) \quad \begin{pmatrix} A + \gamma BB^T & B \\ B^T & 0 \end{pmatrix} \begin{pmatrix} x \\ y \end{pmatrix} = \begin{pmatrix} c + \gamma Bd \\ d \end{pmatrix},$$

where γ is a positive scalar. Such an approach has been considered in various applications (see, e.g., [6]). In the context of optimization, it is related to forming an augmented Lagrangian function [14]; for example, the method of multipliers [2] amounts to applying the Uzawa algorithm [1] to (1.2).

Equation (1.1) can also be reformulated as a linear system that is more general than (1.2), as follows (see, for example, [5]):

$$(1.3) \quad \begin{pmatrix} A + BWB^T & B \\ B^T & 0 \end{pmatrix} \begin{pmatrix} x \\ y \end{pmatrix} = \begin{pmatrix} c + BWd \\ d \end{pmatrix},$$

*Received by the editors April 15, 2004; accepted for publication (in revised form) by A. Wathen March 24, 2005; published electronically January 27, 2006.

<http://www.siam.org/journals/simax/27-3/60679.html>

[†]SCCM Program, Gates 2B, Stanford University, Stanford, CA 94305-9025 (golub@sccm.stanford.edu). The work of this author was supported in part by the US Department of Energy.

[‡]Department of Computer Science, The University of British Columbia, Vancouver BC V6T-1Z4, Canada (greif@cs.ubc.ca, varah@cs.ubc.ca). The work of these authors was supported in part by the Natural Sciences and Engineering Research Council of Canada.

where W is an $m \times m$ matrix. We will assume that W is symmetric positive semidefinite yet $A + BWB^T$ is positive definite. Let us define

$$(1.4) \quad M(W) = A + BWB^T; \quad \mathcal{H}(W) = \begin{pmatrix} M(W) & B \\ B^T & 0 \end{pmatrix}.$$

Note that $\mathcal{H}$ defined in (1.1) is equivalent to $\mathcal{H}(0)$ in (1.4).

An advantage of forming (1.2) or the more general form (1.3) is that the Schur complement is well defined whereas in (1.1) there is no Schur complement associated with a singular A . Thus, one could solve the system by using the decomposition

$$(1.5) \quad \begin{pmatrix} M & B \\ B^T & 0 \end{pmatrix} = \begin{pmatrix} I & 0 \\ B^T M^{-1} & I \end{pmatrix} \begin{pmatrix} M & 0 \\ 0 & -B^T M^{-1} B \end{pmatrix} \begin{pmatrix} I & M^{-1} B \\ 0 & I \end{pmatrix},$$

which holds for any symmetric nonsingular matrix M , including $M = M(W)$ defined in (1.4). Using (1.5), solution methods for positive definite systems can be applied throughout the iterative process of solving for the indefinite matrix $\mathcal{H}(W)$. Other possible numerical advantages of solving (1.2) or (1.3) instead of (1.1) are discussed in [7] and in other places. However, choosing W to make $M(W)$ easy to solve for may hurt the conditioning of $\mathcal{H}(W)$ and the choice of this parameter is delicate. For example, for $W = \gamma I$, as γ grows the conditioning of the saddle point matrix of (1.2) typically deteriorates at a rate proportional to γ^2 whereas the condition number of its associated (1,1) block deteriorates at a rate typically proportional to γ . It may be useful, then, to consider an approach of using a block preconditioner based on $M(W)$ rather than (or in addition to) modifying system (1.1) to (1.2) or (1.3).

Motivated by the above, our goal in this paper is to perform an algebraic study of the block diagonal positive definite preconditioner

$$(1.6) \quad \mathcal{M}(W) = \begin{pmatrix} A + BWB^T & 0 \\ 0 & B^T(A + BWB^T)^{-1}B \end{pmatrix}.$$

Forming the preconditioner (1.6) may be computationally expensive (in particular setting up the Schur complement and solving for it) unless A and B have a very special structure, and in practice cheaper alternatives must be sought. Nevertheless, understanding the spectral properties of (1.6) is useful since it can illuminate the behavior of preconditioners based on approximations of the components of $\mathcal{M}(W)$. For example, finite volume discretization of the Maxwell equation yields a semidefinite (1,1) block corresponding to the curl-curl operator, and in certain applications this operator is replaced by an operator of the form $A + BDB^T$ with D a diagonal scaling matrix [11]. This is not necessarily done in a saddle point system context, but such a formulation is also viable.

Our focus throughout this paper is on the analysis, and we stay within the algebraic framework without associating the problem with any specific application. In section 2 we provide analytical results that characterize the spectral distribution of the preconditioned matrix and indicate what can be expected in terms of convergence of minimum residual Krylov solvers such as MINRES [16]. Section 3 is devoted to the question of how to choose the parameter(s) involved, and we provide a perturbation analysis which may help in the choice of W . The idea here is that aiming to minimize the condition number of the augmented (1,1) block (and in practice look for cheap computational alternatives that keep it small) may prove effective. A few numerical examples are provided in section 4, and in section 5 we draw some conclusions.

2. The block diagonal preconditioner. In [13] it is shown that if A is nonsingular, then a matrix that in our notation here is equal to $\mathcal{M}(0)$ has the attractive property that the associated preconditioned matrix $\mathcal{M}^{-1}(0)\mathcal{H}$ has at most four distinct eigenvalues: $0, 1, \frac{1}{2} \pm \frac{\sqrt{5}}{2}$. Thus, when this matrix is nonsingular, a minimum residual Krylov solver terminates within three iterations (if roundoff errors are ignored). In the context of the linear system we discuss, if (1.1) is first reformulated as (1.3) and then $\mathcal{M}(W)$ is applied as a preconditioner, then we again obtain a preconditioned matrix with three distinct nonzero eigenvalues. However, it is possible that a particular choice of W that works well in solving for $\mathcal{M}(W)$ will not be the ideal choice in terms of the conditioning or the spectral distribution of $\mathcal{H}(W)$. Using $\mathcal{M}(W)$ as a preconditioner applied to (1.1) rather than (1.3) enhances flexibility in that it allows for a choice of W that makes the solve for $\mathcal{M}(W)$ effective without directly having to balance it with considerations related to the conditioning of the saddle point matrix.

Let us begin this part of the analysis by showing that if the matrix of (1.3) were to be used as a *preconditioner* for system (1.1), then a minimal residual Krylov solver converges almost immediately. In [7] it is shown that for general matrices A of size $n \times n$, B and C of size $n \times m$ with full column rank ($m \leq n$), and W of size $m \times m$, if $\tilde{\mathcal{H}}(W) = \begin{pmatrix} A+BWC^T & B \\ C^T & 0 \end{pmatrix}$ is nonsingular, then

$$\tilde{\mathcal{H}}^{-1}(W) = \tilde{\mathcal{H}}^{-1}(0) - \begin{pmatrix} 0 & 0 \\ 0 & W \end{pmatrix}.$$

The same result, for symmetric matrices, can be found in [5, Chap. 12]. From this we can obtain the following result.

PROPOSITION 2.1. *If $\mathcal{H}(W)$ (defined in (1.4)) is used as a preconditioner for solving the linear system (1.1), and a minimum residual Krylov subspace solver is applied, convergence is obtained (in the absence of roundoff errors) within two iterations.*

Proof. Since

$$\mathcal{H}^{-1}(W)\mathcal{H} = \left\{ \mathcal{H}^{-1} - \begin{pmatrix} 0 & 0 \\ 0 & W \end{pmatrix} \right\} \mathcal{H} = \begin{pmatrix} I & 0 \\ WB^T & I \end{pmatrix},$$

it follows that all eigenvalues are equal to 1 and the minimal polynomial is $p_2(z) = (1 - z)^2$. $\square$

From Proposition 2.1 it follows that a constraint preconditioner (see, e.g., [12] or [17] for definition and spectral analysis) based on taking $\mathcal{M}(W)$ as its (1,1) block behaves like a direct solver. Clearly, solving for $A + BWB^T$ may be a nontrivial computational task, and so the effectiveness of this approach depends on the numerical properties of the preconditioning matrix. The preconditioner $\mathcal{M}(W)$ defined in (1.6) is the middle matrix in the decomposition (1.5) applied to $\mathcal{H}(W)$. If it is to be used, then MINRES (which requires a positive definite preconditioner) could be applied.

2.1. Spectral properties of the preconditioned matrix. Consider the matrix $\mathcal{M} = \begin{pmatrix} M & 0 \\ 0 & B^T M^{-1} B \end{pmatrix}$, where M is a positive definite matrix. Let ν be an eigenvalue of $\mathcal{M}^{-1}\mathcal{H}$ with eigenvector $\begin{pmatrix} u \\ v \end{pmatrix}$. Then

$$\begin{pmatrix} A & B \\ B^T & 0 \end{pmatrix} \begin{pmatrix} u \\ v \end{pmatrix} = \nu \begin{pmatrix} M & 0 \\ 0 & B^T M^{-1} B \end{pmatrix} \begin{pmatrix} u \\ v \end{pmatrix}.$$

Substituting $v = \frac{1}{\nu}(B^T M^{-1} B)^{-1} B^T u$, we get a quadratic expression (in ν) for u :

$$[\nu^2 M - \nu A - B(B^T M^{-1} B)^{-1} B^T] u = 0.$$

Denoting $\tilde{u} = M^{\frac{1}{2}} u$ and $C = M^{-1/2} B$, the equation can be written as

$$(2.1) \quad (\nu^2 I - \nu K - P) \tilde{u} = 0,$$

where

$$K = M^{-1/2} A M^{-1/2}; \quad P = C(C^T C)^{-1} C^T.$$

Note that if A is positive definite and $M = A$, then $K = I$ and the result of [13] applies. P is an orthogonal projector of rank m , thus it has m eigenvalues equal to 1 and $n - m$ eigenvalues equal to 0. The matrix K has eigenvalues equal to those of the matrix $M^{-1} A$.

So far we have not relied on specific properties, and (2.1) holds for any positive definite M . We now discuss the specific choice $M(W)$, with W positive semidefinite. We start with $W = \gamma I$, for which we can show that there is an explicit formula that connects P and K .

PROPOSITION 2.2. *Suppose $W = \gamma I$. The projection matrix P is a polynomial of degree m in K , given explicitly by $P = I - p(K)$, where p is the Lagrange interpolant*

$$(2.2) \quad p(x) = \frac{\prod_{i=1}^m \left(x - \frac{\lambda_i}{\lambda_i + \gamma} \right)}{\prod_{i=1}^m \left(1 - \frac{\lambda_i}{\lambda_i + \gamma} \right)}$$

and $\{\lambda_i\}$ are the eigenvalues of the generalized eigenvalue problem $Ax = \lambda BB^T x$.

Proof. Let $L = \frac{1}{\gamma}(I - K)$. Then $L = CC^T = C(I)C^T$, and $L^p = C(C^T C)^{p-1} C^T$ for any positive integer p . Since $P = C(C^T C)^{-1} C^T$, and since by the Cayley-Hamilton theorem $(C^T C)^{-1}$ is a polynomial of degree m in $C^T C$, it follows that P is a polynomial of degree m in L and hence a polynomial of degree m in K .

The polynomial connection is now established, and we can seek an explicit expression. Denote the eigenvalues of K by $\{\mu_i\}$. We have $\text{eig}(K) = \text{eig}(M^{-1} A) = \text{eig}(A - \mu M)$. Thus $\{\mu_i\} = \text{eig}(A - \mu(A + \gamma BB^T)) = \text{eig}((1 - \mu)A - \mu\gamma BB^T)$, and the eigenvalues of K are related to the eigenvalues of the generalized eigenvalue problem $(A - \lambda BB^T)x = 0$ as follows: $\lambda_i = \frac{\mu_i \gamma}{1 - \mu_i}$. Since $\text{rank}(BB^T) = m$, the $n - m$ infinite generalized eigenvalues λ_i map onto $\mu_i = 1$. There are m finite generalized eigenvalues λ_i ; the m eigenvalues of K that are not equal to 1 thus satisfy $0 \leq \mu_i < 1$.

As for the matrix P , since $n - m$ of its eigenvalues are equal to 1, we know that the function p defined in the statement of this theorem satisfies $p(1) = 1$ and $p(\frac{\lambda_i}{\lambda_i + \gamma}) = 0$, and thus p is the Lagrange interpolant of degree m given by (2.2). $\square$

If $W \neq \gamma I$, a polynomial connection does not necessarily exist (in particular if W is singular), but we can still make an observation on the clustering of the eigenvalues.

LEMMA 2.3. *For $M = M(W)$, P and K share the same eigenvectors.*

Proof. We have

$$I - CW C^T = M^{-1/2}(M - BW B^T)M^{-1/2} = M^{-1/2} A M^{-1/2} = K,$$

from which it follows that

$$PK = C(C^T C)^{-1} C^T (I - C W C^T) = KP = P + K - I.$$

Since K and P are commuting normal matrices, they share the same eigenvectors. $\square$

THEOREM 2.4. *Suppose A and W are symmetric positive semidefinite and $M(W)$ is symmetric positive definite. Then the eigenvalues of K , $\{\mu_i\}$, are given as follows.*

1. $n - m$ eigenvalues are equal to 1; the corresponding eigenvectors span the null space of B^T .
2. $\text{nullity}(A)$ eigenvalues are equal to 0; the corresponding eigenvectors span the null space of A .
3. The rest of the eigenvalues are all strictly between 0 and 1, and are given explicitly in terms of the generalized eigenvalue problem $Ax = \lambda B W B^T x$ as follows:

$$\mu_i = \frac{\lambda_i}{\lambda_i + 1}, \quad i = 1, \dots, 2m - \text{nullity}(A).$$

Proof. Suppose $Kx = \mu x$. Then μ is also an eigenvalue of $M^{-1}A$, with eigenvector $\tilde{x} = M^{-1/2}x$, and we have

$$(2.3) \quad A\tilde{x} = \mu(A + B W B^T)\tilde{x}.$$

Proof of statement 1. If z is a vector in the null space of B^T , namely $B^T z = 0$, then from nonsingularity of $\mathcal{H}$ it follows that $Az \neq 0$. Thus z is an eigenvector of (2.3), with eigenvalue $\mu = 1$. Since B has full column rank, this eigenvalue is of multiplicity $n - m$.

Proof of statement 2. Suppose A is semidefinite, and let $z \neq 0$ be a null vector. Since $A + B W B^T$ is positive definite, it follows that $B W B^T z \neq 0$. From this it follows that $\mu = 0$ is a corresponding eigenvalue with multiplicity equal to $\text{nullity}(A)$.

Proof of statement 3. If $\mu \neq 0, 1$ we have, by (2.3), $A\tilde{x} = \frac{\mu}{1-\mu} B W B^T \tilde{x}$. Since A and $B W B^T$ are positive semidefinite, the generalized eigenvalues $\lambda = \frac{\mu}{1-\mu}$ must be nonnegative and hence $0 < \mu = \frac{\lambda}{\lambda+1} < 1$. $\square$

THEOREM 2.5. *If A is positive semidefinite, the eigenvalues of the preconditioned matrix $\mathcal{M}^{-1}(W)\mathcal{H}$ are bounded within the two intervals*

$$\left[-1, \frac{1-\sqrt{5}}{2}\right] \cup \left[1, \frac{1+\sqrt{5}}{2}\right].$$

The eigenvalue 1 is of algebraic multiplicity $n - m + \text{nullity}(A)$. Also, $\text{nullity}(A)$ eigenvalues are equal to -1 .

Proof. Since $\text{null}(B^T)$ is the space of eigenvectors associated with the $n - m$ zero eigenvalues of P and the $n - m$ eigenvalues of K that are equal to 1, and since K and P commute by Lemma 2.3, we get that $n - m$ eigenvalues of the preconditioned matrix $\mathcal{M}^{-1}(W)\mathcal{H}$ satisfy $\nu^2 - \nu = 0$, and when the preconditioned matrix is nonsingular those eigenvalues are all equal to 1.

For the rest of the eigenvalues, by (2.1) we have the connection $\nu^2 - \nu\mu = 1$, which leads to

$$\nu_i = \frac{1}{2} \left(\mu_i \pm \sqrt{\mu_i^2 + 4} \right).$$

The zero eigenvalues of A are mapped onto $\nu = \pm 1$. On the other hand, from what we know about the eigenvalues of K (namely, that they are all bounded between 0 and 1; see Theorem 2.4) we can deduce that

$$1 \leq \frac{1}{2} \left(\mu_i + \sqrt{\mu_i^2 + 4} \right) \leq \frac{1}{2} (1 + \sqrt{5}) \approx 1.618$$

and

$$-0.618 \approx \frac{1}{2} (1 - \sqrt{5}) \geq \frac{1}{2} \left(\mu_i - \sqrt{\mu_i^2 + 4} \right) \geq -1. \quad \square$$

All the eigenvalues of the preconditioned matrix are thus within two narrow intervals whose ends are fixed. We note that some of the bounds presented above can be obtained by applying [18, Lemma 2.1] or [17, Proposition 2] to the matrix $\mathcal{M}^{-1/2} \mathcal{H} \mathcal{M}^{-1/2}$; however at least one of the bounds obtained here is tighter, since we use the special connection between K and P , observed in Lemma 2.3, which does not hold in more general cases.

2.2. Numerical properties of the Schur complement. The preconditioner we are studying involves forming the Schur complement

$$-S(W) = -B^T (A + BWB^T)^{-1} B.$$

The same Schur complement arises if (1.3) is to be solved using a block Gaussian elimination procedure. An analysis of the Schur complement is useful for evaluating the merits of the general approach, even if in practice one may seek cheaper alternatives.

When A is positive definite $S(0) = B^T A^{-1} B$ is defined; for simplicity of notation let us denote it simply by S . If A is semidefinite, consider a small perturbation of A , e.g., $A(\varepsilon) \equiv A + \varepsilon BB^T$ for some small ε , so that the Schur complement is still defined, and in this case we refer to $A(\varepsilon)$ and $S(\varepsilon)$ simply as A and S , and both are symmetric positive definite.

Using the Sherman–Morrison formula [8, p. 50] in its block version, we have

$$\begin{aligned} S(W) &= B^T (A + BWB^T)^{-1} B \\ &= B^T A^{-1} B - B^T A^{-1} B (I + WB^T A^{-1} B)^{-1} WB^T A^{-1} B \\ &= B^T A^{-1} B (I + WB^T A^{-1} B)^{-1} (I + WB^T A^{-1} B - WB^T A^{-1} B) \\ &= S(I + WS)^{-1}, \end{aligned}$$

and we have the following commuting relation.

PROPOSITION 2.6. *If S and W commute, then S and $S(W)$ commute.*

Proof. If $WS = SW$, then we have

$$\begin{aligned} S \cdot S(W) &= S^2 (I + WS)^{-1} = S(S^{-1} + WSS^{-1})^{-1} \\ &= S(S^{-1} + S^{-1}SW)^{-1} = S(I + WS)^{-1} S = S(W) \cdot S. \quad \square \end{aligned}$$

This leads to the following conclusion with regard to the conditioning of the Schur complement.

PROPOSITION 2.7. *Suppose that S and W commute, and let $\{\varphi_i\}$ be the eigenvalues of $S(W)$, $\{\beta_i\}$ the eigenvalues of S , and $\{\alpha_i\}$ the eigenvalues of W . Then*

$$(2.4) \quad \varphi_i = \frac{\beta_i}{\beta_i \alpha_i + 1}.$$

Proof. From Proposition 2.6 it follows that S , $S(W)$, and W share the same eigenvectors. Ordering the eigenvalues so that the i th eigenvalue of each of the matrices corresponds to the same eigenvector, (2.4) readily follows. $\square$

COROLLARY 2.8. *Let $S(\gamma)$ denote $S(W)$ with $W = \gamma I$. Then $\kappa_2(S(\gamma)) \leq \kappa_2(S)$ for any $\gamma > 0$.*

Proof. By Proposition 2.7 we have $S(\gamma) = S(I + \gamma S)^{-1}$ and the eigenvalues of $S(\gamma)$ are $\varphi_i = \frac{\beta_i}{1+\gamma\beta_i}$. The desired result readily follows, by monotonicity. $\square$

Proposition 2.7 has limited practical value, since S and W rarely commute (even if W is a simple nonconstant diagonal matrix there is no commuting). However, even when the commuting condition does not hold, it may still nearly hold.

3. Minimizing the condition number of the augmented (1,1) block.
Suppose that

$$W = \text{diag}(w_j), \quad w_j \geq 0.$$

The (1,1) block can be expressed as a sum of rank-1 perturbations of A as follows:

$$M(W) = A + \sum_{j \in \mathcal{S}} w_j b_j b_j^T,$$

where b_j is the j th column of B , and $\mathcal{S}$ is the set of indices j for which $w_j \neq 0$. The number of nonzero entries of W is equal to at least the nullity of A if positive definiteness of the modified (1,1) block is to be obtained.

A possible way of selecting W is to base it on condition number minimization of $A + BWB^T$. See [15] and the references therein for a survey of eigenvalue optimization problems; see also [10] for minimizing the condition number given a prescribed sparsity pattern.

Since $M(W)$ defined above is a sum of A and a sequence of rank-1 contributions, we can simplify the discussion by considering a single rank-1 perturbation:

$$M(w) = A + wbb^T,$$

where A is positive definite and $w \geq 0$.

PROPOSITION 3.1. *Suppose that A is symmetric positive semidefinite with nullity 1, and suppose that $(0, u_n)$ and (λ_1, u_1) are the eigenpairs with the smallest and the largest eigenvalues of A (known to be simple), and $\lambda_{n-1} > 0$ is the smallest positive eigenvalue of A . Then the function $\kappa_2(A + wu_n u_n^T)$ is strictly decreasing in w for $0 = \lambda_n < w < \lambda_{n-1}$, is constant for $\lambda_{n-1} \leq w \leq \lambda_1$, is strictly increasing for $w > \lambda_1$, and is explicitly given in terms of the eigenvalues of A as follows:*

$$(3.1) \quad \kappa_2(A + wu_n u_n^T) = \frac{\max(\lambda_1, w)}{\min(\lambda_{n-1}, w)}.$$

The proof follows straightforwardly from orthogonality of the eigenvectors and is omitted. In [4, Theorem 5.3] it is shown that for saddle point systems rank-1 perturbations of a (1,1) block with nullity 1 change the zero eigenvalue to a positive one and perturb the eigenvectors, and the norm of the perturbation is related to the angle formed between the null vector of the matrix and the vector used for the rank-1 perturbation. The angle between a column of B and the null vectors of A is relevant in our discussion here, in which the condition number is considered.

Let the eigenvalues of A be $\lambda_1 > \lambda_2 \geq \cdots \geq \lambda_n > 0$, and let us define

$$\kappa(w) \equiv \kappa_2(A + wbb^T); \quad \lambda_i(w) \equiv \lambda_i(A + wbb^T).$$

Also, we denote by $u_i(w)$ the eigenvectors of $A + wbb^T$. The eigenvalues of $M(w)$ and A interlace:

$$\lambda_n \leq \lambda_n(w) \leq \lambda_{n-1} \leq \cdots \leq \lambda_1 \leq \lambda_1(w).$$

In general, each $\lambda_k(w)$ is a piecewise smooth function, with “corners” possible at multiple eigenvalues. In order to avoid discontinuities with the extremal eigenvalues λ_1 and λ_n , we assumed above that they were *simple* eigenvalues of A ; then $\lambda_1(w)$, $\lambda_n(w)$, and hence $\kappa(w) = \lambda_1(w)/\lambda_n(w)$ are all smooth functions. Note also that only $\lambda_1(w)$ increases without limit as $w \rightarrow \infty$; the other eigenvalues of $A + wbb^T$ are all bounded by the next eigenvalue of A . For $A = D = \text{diag}(\lambda_k)$, the limiting values are the roots of $\sum_{k=1}^n \frac{b_k^2}{(\lambda - \lambda_k)} = 0$ [8, section 8.5.3].

We are interested in the behavior of $\kappa(w)$ for $w > 0$, in particular whether it has a minimum. To this end, we need expressions for the derivatives of the eigenvalues, which can be derived from perturbation theory.

PROPOSITION 3.2. *Let $\lambda_k(w)$ be a simple eigenvalue of $M(w)$. Then*

$$\lambda'_k(w) = [b^T u_k(w)]^2,$$

where $u_k(w)$ is the corresponding (normalized) eigenvector.

Proof. A simple perturbation expansion [19] for $\lambda_k(w)$ about w gives this immediately, since our perturbation matrix is bb^T . $\square$

Applying Proposition 3.2 to $\lambda_1(w)$ and $\lambda_n(w)$ gives

$$(3.2) \quad \frac{\kappa'(w)}{\kappa(w)} = \frac{\lambda'_1(w)}{\lambda_1(w)} - \frac{\lambda'_n(w)}{\lambda_n(w)} = \frac{(b^T u_1(w))^2}{\lambda_1(w)} - \frac{(b^T u_n(w))^2}{\lambda_n(w)}.$$

In general, $\kappa(w)$ does not necessarily have a minimum for $w > 0$. However, we can state when it does. Below we will again (as we did in section 2.2) assume that A is positive definite, and when it is not we will consider a small perturbation $A(\varepsilon)$ that is positive definite as the matrix for which the following result is stated.

PROPOSITION 3.3. *When λ_1 and λ_n are simple, a necessary and sufficient condition for $\kappa(A + wbb^T)$ to have at least one minimum for $w > 0$ is*

$$\frac{|b^T u_1|}{|b^T u_n|} < \sqrt{\kappa_2(0)}.$$

Proof. Since $\kappa(w) \rightarrow \infty$ as $w \rightarrow \infty$, it will have at least one minimum if and only if $\kappa'(0) < 0$. From (3.2), this holds if and only if $|b^T u_1(0)|^2 < \kappa(0)|b^T u_n(0)|^2$. $\square$

From Proposition 3.3 it follows that it may be desirable that b form a small angle with the eigenvector of A that corresponds to the smallest eigenvalue (a null vector if A is rank deficient). This is a reasonable condition, which can be concluded also from the analysis in [4, section 5]. We can in fact show more, as follows.

THEOREM 3.4. *When λ_1 and λ_n are simple, $\kappa(A + wbb^T)$ has at most one minimum for $w > 0$.*

Proof. We shall show that $\kappa(w)$ is convex: $\kappa''(w) > 0$. To see this, consider (3.2). Multiplying by $\kappa(w)$ on both sides and differentiating with respect to w yield

$$\kappa'' = \kappa' \cdot \left(\frac{\lambda'_1}{\lambda_1} - \frac{\lambda'_n}{\lambda_n} \right) + \kappa \cdot \left(\frac{\lambda_1 \lambda''_1 - (\lambda'_1)^2}{\lambda_1^2} - \frac{\lambda_n \lambda''_n - (\lambda'_n)^2}{\lambda_n^2} \right).$$

Simplifying, we get

$$\frac{\kappa''}{\kappa} = \frac{\lambda_1''}{\lambda_1} - \frac{\lambda_n''}{\lambda_n} + 2 \frac{\lambda_n'}{\lambda_n} \left(\frac{\lambda_n'}{\lambda_n} - \frac{\lambda_1'}{\lambda_1} \right).$$

From [19, p. 20] we can get an expression for the second derivative. For a general perturbation $A + wB$,

$$\lambda_k'' = 2 \sum_{i \neq k} \frac{(u_i^T B u_k)^2}{\lambda_k - \lambda_i}.$$

Here we have $B = bb^T$, so

$$\lambda_1'' = 2(b^T u_1)^2 \sum_{i=2}^n \frac{(b^T u_i)^2}{\lambda_1 - \lambda_i}, \quad \lambda_n'' = 2(b^T u_n)^2 \sum_{i=1}^{n-1} \frac{(b^T u_i)^2}{\lambda_n - \lambda_i}.$$

Now define $c_k = b^T u_k$. Then

$$\frac{\kappa''}{2\kappa} = \frac{c_1^2}{\lambda_1} \sum_{i=2}^n \frac{c_i^2}{\lambda_1 - \lambda_i} - \frac{c_n^2}{\lambda_n} \sum_{i=1}^{n-1} \frac{c_i^2}{\lambda_n - \lambda_i} + \frac{c_n^2}{\lambda_n} \left(\frac{c_n^2}{\lambda_n} - \frac{c_1^2}{\lambda_1} \right).$$

Notice that the terms in the first sum are positive and in the second sum are negative. Collecting terms, we have

$$\frac{\kappa''}{2\kappa} = \frac{c_1^2}{\lambda_1} \sum_{i=2}^n \frac{c_i^2}{\lambda_1 - \lambda_i} + \frac{c_n^2}{\lambda_n} \left[c_1^2 \left(\frac{1}{\lambda_1 - \lambda_n} - \frac{1}{\lambda_1} \right) + \frac{c_n^2}{\lambda_n} + \sum_{i=2}^{n-1} \frac{c_i^2}{\lambda_i - \lambda_n} \right],$$

and all terms of the right are positive. $\square$

As stated in Proposition 3.1, for $A + wu_n u_n^T$ any choice of the parameter w between the minimal positive eigenvalue and the maximal eigenvalue of A will obtain the minimized condition number. The upper bound of that range corresponds to taking $w = \|A\|_2 = \|A\|_2 / \|u_n\|_2^2$. Given b , if it forms a small angle with the null vector of A we could make the choice $w = \|A\|_2 / \|b\|_2^2$. This choice is relatively easy to compute or estimate. For the choice $A + wBB^T$ we can apply a similar strategy and set $w = \|A\|_2 / \|B\|_2^2$, and if 2-norms are difficult to calculate, then norm estimators could be used. This establishes a computationally inexpensive way of choosing the parameter w in the general case.

Finally, we mention that other choices of W are possible. For example, setting $W = (B^T B)^{-1}$ (see, for example, [5]) has the advantage that the matrix $B(B^T B)^{-1} B^T$ is an orthogonal projector onto the range of B , which is orthogonal to the null space of B^T , and since the null space of A does not intersect with the null space of B^T either, such a choice of W is viable.

4. Numerical examples. We have tested with a set of matrices, including a few from the CUTE/CUTEr collection [9], with self-generated right-hand-side vectors. The matrices were used merely for testing purposes, without considering the type of constraints in the underlying optimization problem.

Example 4.1. This example validates some of the analytical observations made in section 3. The **genhs28** problem from the CUTE collection is a tiny 28×28 saddle point matrix, where A is 10×10 . A is tridiagonal, almost Toeplitz, with

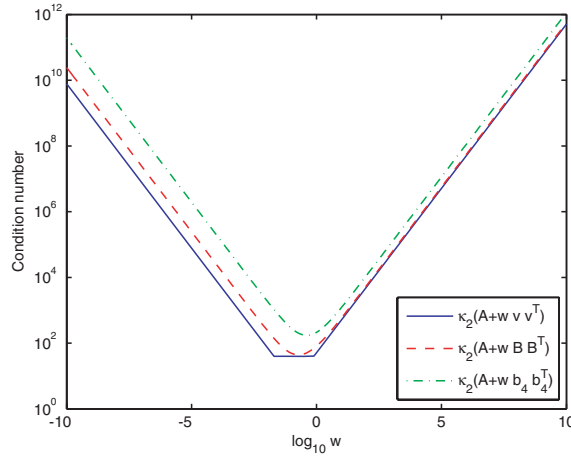


FIG. 4.1. Example 4.1: Condition numbers of $A + wvv^T$, $A + wBB^T$, and $A + wb_4b_4^T$ for the **genhs28** matrix. Here v is the null vector of A and b_4 is the fourth column of B .

2, 4, 2 along its superdiagonal, main diagonal, and subdiagonal, respectively, except $A_{1,1} = A_{10,10} = 2$. The matrix is thus singular with nullity 1, and its null vector is $v = \alpha \cdot (-1, 1, -1, 1, -1, 1, -1, 1, -1, 1)^T$. B is 10×8 with values 1, 2, and 3 along its main diagonal, first subdiagonal, and second subdiagonal, respectively.

The eigenvalues of A are $8\sin^2(\frac{\pi j}{2n})$, $j = 0, \dots, 9$, and so $\lambda_{\min} = 0$, $\lambda_{\max} = 7.8042\dots$, and the smallest positive eigenvalue is $0.1958\dots$.

The inner products of the eight columns of B with v are equal in magnitude and alternate in sign. Figure 4.1 shows the condition numbers of the perturbation of the matrix A by multiples of vv^T , the matrix obtained by perturbing A by a multiple of $b_4b_4^T$, where b_4 is the fourth column of B (arbitrary choice), and the matrix obtained by adding a multiple of BB^T to A . The range of values of w for which a nearly optimal condition number is obtained is fairly large and is similar for the three matrices. The condition numbers were computed for $\log_{10} w = -10 : 0.1 : 10$ (201 values for each of the matrices). We obtained $\min_w \kappa_2(A + wbb^T) = 172.3\dots$, $\min_w \kappa_2(A + wBB^T) = 44.2\dots$, and $\min_w \kappa_2(A + wvv^T) = 39.9\dots$. The choice $w = \|A\|/\|B\|^2$ gives $\log_{10} w = -0.6367\dots$, which is well within the range of values for which the condition numbers in the plot are close to the minimum, and for this value we obtained $\kappa_2(A + wBB^T) = 44.6\dots$, which is extremely close to the minimum stated above. Finally, we mention that $\kappa_2(A + B(B^TB)^{-1}B^T) = 25.5\dots$, which is superior to all the other choices and may justify this as a viable choice, as briefly discussed at the end of section 3.

Example 4.2. We took a version of the CUTE matrix **cvxqp1** of size 1500×1500 ; A is 1000×1000 with rank 986, and B is 1000×500 . The smallest positive eigenvalue of A is $9.4756\dots \times 10^{-7}$. The rank of the saddle point matrix is 1499, and thus $A + BWB^T$ is rank deficient with nullity of (at least) 1 for any choice of W . We set $W = (\|A\|_2/\|B\|_2^2)I$ and generated the right-hand side vector so that the solution was a vector of all 1's.

Figure 4.2 shows the history of residual norms using MINRES with preconditioners $\mathcal{M}$, for five choices of the (1,1) block M . In the top two curves M is the identity matrix and the diagonal of A . The bottom three are related to the shift by a scalar multiple of BB^T : we set $w = \|A\|/\|B\|^2 = 156.13\dots$, and take $A + wBB^T$,

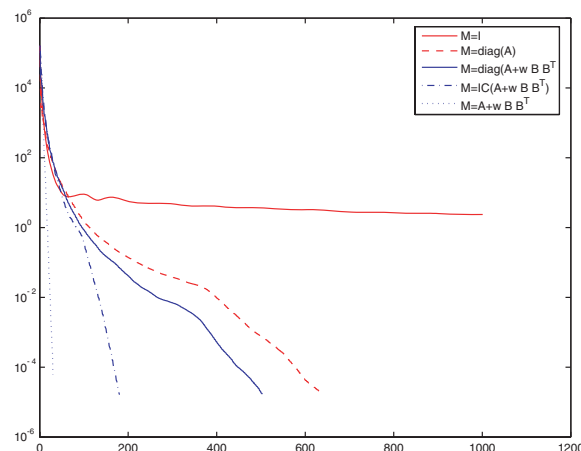


FIG. 4.2. Example 4.2: Convergence of the preconditioned MINRES with various positive definite block diagonal preconditioners.

its diagonal part, and its incomplete Cholesky decomposition as our choices for M . The number of nonzeros of A is 6968, whereas $A + wBB^T$ has 9950 nonzeros; thus matrix-vector products and linear solves with the latter are more computationally expensive. However the reduced iteration count more than compensates for that, and the overall computational work is reduced when $A + wBB^T$ or its approximations are used. We note that A is sparse but not narrow banded, and is not structured in a way that allows for easily exploiting its nonzero pattern.

Example 4.3. The CUTE matrix **gouldqp3** is 1048×1048 , with A of size 699×699 and rank 697. The smallest positive eigenvalue of A is approximately $4.0514 \cdot 10^{-5}$, and its largest eigenvalue is approximately 5.236. The rank-2 null space of A is spanned by a vector of all 1's (or any other nonzero constant) except the last entry, which is zero, and a vector of all 0's except the last, nonzero entry. Since this null space is explicitly available, the observations of section 3 can be used. We find the columns of B that form the smallest angle with the null vectors of A . The first and the last columns of B are valid (though not unique) choices; let us denote them by b_1 and b_m , respectively. We define $V = w_1 b_1 + w_2 b_m$, with $w_1 = \|A\|/\|b_1\|^2$ and $w_2 = \|A\|/\|b_2\|^2$. We then set $M_1 = A + VV^T$ and $M_2 = A + wBB^T$, with $w = \|A\|/\|B\|^2$.

TABLE 4.1
Number of nonzeros in A , B , $A + VV^T$, and $A + BB^T$ for the **gouldqp3** matrix.

$\text{nnz}(A)$	2092
$\text{nnz}(B)$	1047
$\text{nnz}(A + VV^T)$	2105
$\text{nnz}(A + BB^T)$	4185

The number of nonzeros of all the matrices involved is given in Table 4.1. As is evident, M_1 is very close in sparsity to A . The advantage of forming a rank-2 perturbation is that the eigenvalues are all nearly equal to the three eigenvalues of the case described in [13], due to interlacing and the high algebraic multiplicity of each of the eigenvalues. This is illustrated in Figure 4.3. For the choice $A + wBB^T$ the analysis of section 2 holds, and all eigenvalues are within the intervals stated in

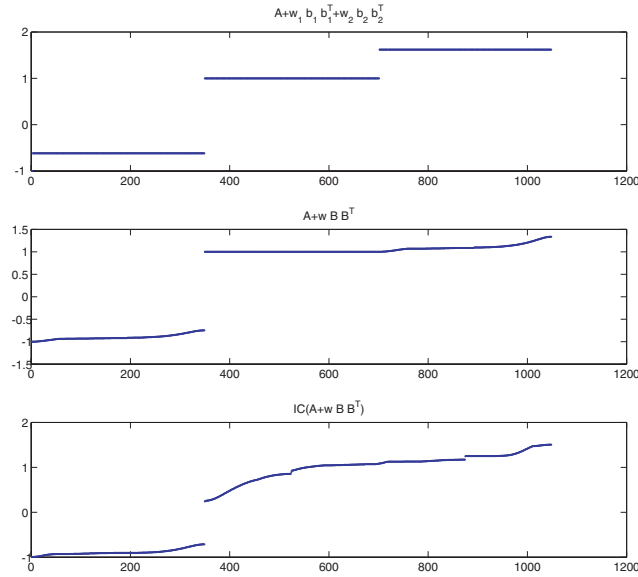


FIG. 4.3. Example 4.3: Eigenvalue distribution of the preconditioned matrix `gouldqp3` using various positive definite block diagonal preconditioners.

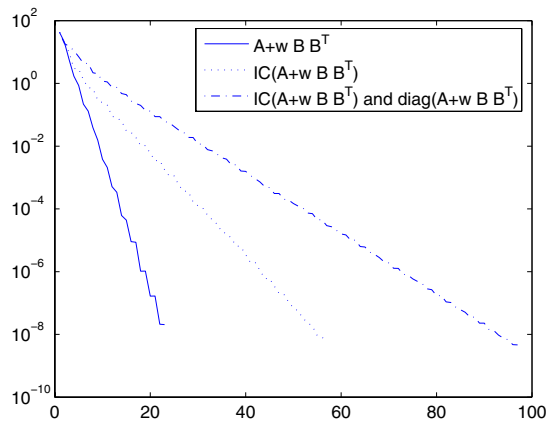


FIG. 4.4. Example 4.3: Convergence of preconditioned MINRES with various positive definite block diagonal preconditioners. The dot-dashed line refers to a preconditioner whose (1,1) block is the incomplete Cholesky factorization of $A + wBB^T$ and whose (2,2) block is $B^T \text{diag}(A + wBB^T)^{-1} B$.

Theorem 2.5, with the expected algebraic multiplicities. Finally, for the incomplete Cholesky factorization the result of the theorem does not hold, but the eigenvalues of the preconditioned matrix are still fairly close to the eigenvalues in the intervals stated in Theorem 2.5, and none of them is close to zero. Numerical experiments show that the eigenvalue structure well predicts the convergence behavior of preconditioned MINRES, as demonstrated in Figure 4.4.

5. Conclusions. We have performed an algebraic analysis for solving (1.1) by using the block diagonal positive definite preconditioner (1.6), whose (1,1) block is $A + BWB^T$. While the preconditioner in that precise form may be computationally expensive, it forms a basis for a preconditioning approach (based on approximations of the components of (1.6)) that may be useful in practice.

Our analysis shows that the eigenvalues of the preconditioned matrix are clustered in fixed intervals whose ends are fixed far from the origin. Moreover, the eigenvalue 1 is of algebraic multiplicity at least $n - m$, and interestingly, the multiplicity goes up with the nullity of A . In section 3 we discussed an approach of minimizing the condition number of $A + BWB^T$. Our analysis shows that columns of B that form a small angle with the null vectors of A should be considered in the selection of nonzero entries of W . In practice obtaining the null space of A is a computationally nontrivial task for large scale problems, and we have provided analytical justification for using all the columns of B and choosing a scaling parameter of the form $w = \|A\|/\|B\|^2$. This involves a minimal computational effort when cheap norm or extremal eigenvalue estimators are used.

Acknowledgments. We are grateful to the two referees, whose helpful suggestions have substantially improved this paper. We dedicate this paper to Michael Saunders, a kind gentleman and outstanding scientist, for the tireless devotion and unflagging patience with which he spends countless hours with colleagues, helping to improve the quality of their work and writing. Among his many contributions to numerical linear algebra and optimization, Michael has played a pivotal role in the software implementation of robust algorithms for solving the type of indefinite linear systems discussed in this paper.

REFERENCES

- [1] K. ARROW, L. HURWICZ, AND H. UZAWA, *Studies in Nonlinear Programming*, Stanford University Press, Stanford, CA, 1958.
- [2] K. J. ARROW AND R. M. SOLOW, *Gradient methods for constrained maxima, with weakened assumptions*, in *Studies in Linear and Nonlinear Programming*, K. J. Arrow, L. Hurwicz, and H. Uzawa, eds., Stanford University Press, Stanford, CA, 1958, pp. 166–176.
- [3] M. BENZI, G. H. GOLUB, AND J. LIESEN, *Numerical solution of saddle-point systems*, in *Acta Numerica*, Acta Numer. 14, Cambridge University Press, Cambridge, UK, 2005, pp. 1–137.
- [4] P. BOCHEV AND R. B. LEHOUCQ, *On the finite element solution of the pure Neumann problem*, SIAM Rev., 47 (2005), pp. 50–66.
- [5] R. FLETCHER, *Practical Methods of Optimization*, 2nd ed., Wiley, Chichester, UK, 1987.
- [6] M. FORTIN AND R. GLOWINSKI, *Augmented Lagrangian Methods: Applications to the Numerical Solution of Boundary-Value Problems*, North-Holland, Amsterdam, 1983.
- [7] G. H. GOLUB AND C. GREIF, *On solving block-structured indefinite linear systems*, SIAM J. Sci. Comput., 24 (2003), pp. 2076–2092.
- [8] G. H. GOLUB AND C. F. VAN LOAN, *Matrix Computations*, 3rd ed., Johns Hopkins University Press, Baltimore, MD, 1996.
- [9] N. I. M. GOULD, D. ORBAN, AND PH. L. TOINT, *CUTEr, a Constrained and Unconstrained Testing Environment, Revisited*, <http://cuter.rl.ac.uk/cuter-www/>.
- [10] A. GREENBAUM AND G. H. RODRIGUE, *Optimal preconditioners for a given sparsity pattern*, BIT, 29 (1990), pp. 610–634.
- [11] R. HIPTMAIR, *Finite elements in computational electromagnetism*, in *Acta Numerica*, Acta Numer. 11, Cambridge University Press, Cambridge, UK, 2002, pp. 237–339.
- [12] C. KELLER, N. I. M. GOULD, AND A. J. WATHEN, *Constraint preconditioning for indefinite linear systems*, SIAM J. Matrix Anal. Appl., 21 (2000), pp. 1300–1317.
- [13] M. F. MURPHY, G. H. GOLUB, AND A. J. WATHEN, *A note on preconditioning for indefinite linear systems*, SIAM J. Sci. Comput., 21 (2000), pp. 1969–1972.
- [14] J. NOCEDAL AND S. WRIGHT, *Numerical Optimization*, Springer-Verlag, New York, 1999.
- [15] M. L. OVERTON, *Large-scale optimization of eigenvalues*, SIAM J. Optim., 2 (1992), pp. 88–120.

- [16] C. C. PAIGE AND M. A. SAUNDERS, *Solution of sparse indefinite systems of linear equations*, SIAM J. Numer. Anal., 12 (1975), pp. 617–629.
- [17] I. PERUGIA AND V. SIMONCINI, *Block-diagonal and indefinite symmetric preconditioners for mixed finite element formulations*, Numer. Linear Algebra Appl., 7 (2000), pp. 585–616.
- [18] T. RUSTEN AND R. WINTHER, *A preconditioned iterative method for saddlepoint problems*, SIAM J. Matrix Anal. Appl., 13 (1992), pp. 887–904.
- [19] J. H. WILKINSON, *The Algebraic Eigenvalue Problem*, Clarendon Press, Oxford, UK, 1965.

A LOWER BOUND ON THE C -NUMERICAL RADIUS OF NILPOTENT MATRICES APPEARING IN COHERENT SPECTROSCOPY*

CHI-KWONG LI[†] AND HUGO J. WOERDEMAN[‡]

Abstract. We provide a lower bound for the efficiency of polarization or coherence transfer between quantized states under unitary transformations. Mathematically the problem is the determination of the C -numerical radius of A for certain nilpotent matrices C and A . The presented lower bound is conjectured to be exact as it coincides with numerical data provided in [U. Helmke et al., *J. Global Optim.*, 23 (2002), pp. 283–308].

Key words. C -numerical radius, NMR spectroscopy, nilpotent matrices

AMS subject classifications. 15A60, 81V99

DOI. 10.1137/040613330

1. Introduction. In the study of the efficiency of polarization or coherence transfer between quantized states under unitary transformations (see [2, 3, 4, 6, 7]), one is interested in determining or estimating the quantity

$$(1.1) \quad b(A_n, C_n) := \max_{UU^* = \mathbb{I}} |\operatorname{tr}(C_n^* U A_n U^*)|,$$

where U^* denotes the Hermitian transpose of U , and A_n and C_n are given matrices derived from a spin system, defined as follows:

$$(1.2) \quad A_n = \begin{pmatrix} N_n & 0 \\ 0 & N_n \end{pmatrix}, \quad C_n = \begin{pmatrix} 0 & 0 \\ \mathbb{I}_{2^n} & 0 \end{pmatrix},$$

with $\mathbb{I}_m$ the $m \times m$ identity matrix and N_n given inductively by

$$N_0 = (0), \quad N_n = \begin{pmatrix} N_{n-1} & 0 \\ \mathbb{I}_{2^{n-1}} & N_{n-1} \end{pmatrix}.$$

Note that the matrices A_n and C_n are of size $2^{n+1} \times 2^{n+1}$. In matrix analysis literature, the quantity $b(A_n, C_n)$ is called the C_n^* -numerical radius of A_n . In [4] the authors proved that $b(A_1, C_1) = 2$ and $b(A_2, C_2) = 4$. Based on numerical methods, they conjectured the values shown in Table 1.1.

TABLE 1.1

n	3	4	5	6
$b(A_n, C_n)$	$4(1 + \sqrt{3})$	$8(1 + \sqrt{3})$	$16(1 + \sqrt{3}) + 4\sqrt{5}$	$32(1 + \sqrt{3}) + 8\sqrt{5}$

*Received by the editors August 12, 2004; accepted for publication (in revised form) by U. Helmke April 14, 2005; published electronically January 27, 2006. The research of both authors was supported in part by the National Science Foundation.

<http://www.siam.org/journals/simax/27-3/61333.html>

[†]Department of Mathematics, The College of William and Mary, P.O. Box 8795, Williamsburg, VA 23185-8795 (ckli@math.wm.edu).

[‡]Department of Mathematics, Drexel University, 3141 Chestnut Street, Philadelphia, PA 19104 (hugo@math.drexel.edu).

In this note, we show that there is a systematic way to extrapolate these values for general n , and construct unitary matrices U_n such that $\text{tr}(C_n^* U_n A_n U_n^*)$ attains these values. The key idea in our proof is a reduction of A_n to a weighted Jordan form $\tilde{A}_n$ using unitary similarity transforms. One can then get our proposed bounds using the off-diagonal entries of $\tilde{A}_n$.

In the next section we present our main result, and in the last section we discuss some open problems.

2. Main result. Recall that two matrices A and $\tilde{A}$ are unitarily similar (notation: $A \sim \tilde{A}$) if there is a unitary matrix U so that $A = U^* \tilde{A} U$. Clearly, when $A \sim \tilde{A}$ and $C \sim \tilde{C}$, then $b(A, C) = b(\tilde{A}, \tilde{C})$. If we let $\tilde{C}_n$ be the direct sum of 2^n copies of $\begin{pmatrix} 0 & 0 \\ 1 & 0 \end{pmatrix}$, then $C_n \sim \tilde{C}_n$. Thus, to compute $b(A_n, C_n) = b(A_n, \tilde{C}_n)$, we need to find a unitary U_n to maximize the quantity

$$(2.1) \quad \left| \text{tr}(\tilde{C}_n^* U_n^* A_n U_n) \right| = \left| \sum_{j=1}^{2^n} a_{2j, 2j-1} \right|, \quad \text{where } U_n^* A_n U_n = (a_{ij}).$$

Therefore, we need to focus on those unitary matrices U such that $U^* A_n U$ has large positive values on the subdiagonal. Proposition 2.1 will give the right tools. We need some notation to describe the result.

Consider the $(j+1) \times (j+1)$ matrices B_j , S_j , J_j , and the $(j+1) \times j$ matrix Z_j :

$$B_j = \begin{pmatrix} 0 & 0 & 0 & \cdots & 0 \\ \sqrt{j}\sqrt{1} & 0 & 0 & \cdots & 0 \\ 0 & \sqrt{j-1}\sqrt{2} & 0 & \cdots & 0 \\ \vdots & \vdots & \ddots & \ddots & \vdots \\ 0 & 0 & \cdots & \sqrt{1}\sqrt{j} & 0 \end{pmatrix},$$

$$S_j = \begin{pmatrix} \sqrt{0} & & \bigcirc \\ & \sqrt{1} & \\ & & \ddots \\ \bigcirc & & & \sqrt{j} \end{pmatrix}, \quad J_j = \begin{pmatrix} \bigcirc & & 1 \\ & 1 & \\ & & \ddots \\ 1 & & & \bigcirc \end{pmatrix}, \quad Z_j = \begin{pmatrix} 0 & \cdots & 0 \\ 1 & & \bigcirc \\ \bigcirc & \ddots & \\ & & 1 \end{pmatrix}.$$

Here the big circles indicate that the unspecified entries are zeros. Note that $B_0 = (0)$.

PROPOSITION 2.1. *The $(2j+2) \times (2j+2)$ matrix*

$$(2.2) \quad U_j = \frac{1}{\sqrt{j+1}} \begin{pmatrix} -S_j Z_j & J_j Z_{j+1}^* S_{j+1} J_{j+1} \\ J_j S_j Z_j J_{j-1} & Z_{j+1}^* S_{j+1} \end{pmatrix}$$

is unitary and the following equality holds:

$$(2.3) \quad \begin{pmatrix} B_j & 0 \\ \mathbb{I} & B_j \end{pmatrix} = U_j \begin{pmatrix} B_{j-1} & 0 \\ 0 & B_{j+1} \end{pmatrix} U_j^*.$$

Note that U_j is a real matrix with a simple structure. For instance,

$$U_1 = \frac{1}{\sqrt{2}} \begin{pmatrix} 0 & \sqrt{2} & 0 & 0 \\ -1 & 0 & 1 & 0 \\ 1 & 0 & 1 & 0 \\ 0 & 0 & 0 & \sqrt{2} \end{pmatrix}, \quad U_2 = \frac{1}{\sqrt{3}} \begin{pmatrix} 0 & 0 & \sqrt{3} & 0 & 0 & 0 \\ -1 & 0 & 0 & \sqrt{2} & 0 & 0 \\ 0 & -\sqrt{2} & 0 & 0 & 1 & 0 \\ \sqrt{2} & 0 & 0 & 1 & 0 & 0 \\ 0 & 1 & 0 & 0 & \sqrt{2} & 0 \\ 0 & 0 & 0 & 0 & 0 & \sqrt{3} \end{pmatrix}.$$

Proof of Proposition 2.1. Note that each column of U_j has at most two nonzero entries. It is straightforward to check that the columns of U_j form an orthorormal family. Also, it is straightforward to check that

$$\begin{pmatrix} B_j & 0 \\ \mathbb{I} & B_j \end{pmatrix} U_j = U_j \begin{pmatrix} B_{j-1} & 0 \\ 0 & B_{j+1} \end{pmatrix}. \quad \square$$

As $B_1 = N_1$, (2.3) yields that

$$N_2 = \begin{pmatrix} N_1 & 0 \\ \mathbb{I} & N_1 \end{pmatrix} \sim B_0 \oplus B_2.$$

Here $A \oplus B$ stands for the direct sum of A and B , i.e., $A \oplus B = \begin{pmatrix} A & 0 \\ 0 & B \end{pmatrix}$. Clearly, $A \oplus B \sim B \oplus A$, but in general $A \oplus B \neq B \oplus A$. Next, again using (2.3), we have

$$N_3 = \begin{pmatrix} N_2 & 0 \\ \mathbb{I} & N_2 \end{pmatrix} \sim \begin{pmatrix} B_0 \oplus B_2 & 0 \\ \mathbb{I} & B_0 \oplus B_2 \end{pmatrix} \sim \begin{pmatrix} B_0 & 0 \\ \mathbb{I} & B_0 \end{pmatrix} \oplus \begin{pmatrix} B_2 & 0 \\ \mathbb{I} & B_2 \end{pmatrix} \sim B_1 \oplus (B_1 \oplus B_3).$$

We shall abbreviate $B_1 \oplus B_1 \oplus B_3$ as $2B_1 \oplus B_3$. In other words,

$$nB_j := B_j \oplus \cdots \oplus B_j,$$

where B_j appears n times in the right-hand side. Continuing this way, we get

$$\begin{aligned} N_4 &\sim 2 \begin{pmatrix} B_1 & 0 \\ \mathbb{I} & B_1 \end{pmatrix} \oplus \begin{pmatrix} B_3 & 0 \\ \mathbb{I} & B_3 \end{pmatrix} \sim 2(B_0 \oplus B_2) \oplus (B_2 \oplus B_4) \sim 2B_0 \oplus 3B_2 \oplus B_4, \\ N_5 &\sim 2 \begin{pmatrix} B_0 & 0 \\ \mathbb{I} & B_0 \end{pmatrix} \oplus 3 \begin{pmatrix} B_2 & 0 \\ \mathbb{I} & B_2 \end{pmatrix} \oplus \begin{pmatrix} B_4 & 0 \\ \mathbb{I} & B_4 \end{pmatrix} \\ &\sim 2B_1 \oplus 3(B_1 \oplus B_3) \oplus (B_3 \oplus B_5) \sim 5B_1 \oplus 4B_3 \oplus B_5, \end{aligned}$$

and so on. In general, we have

$$(2.4) \quad N_{2m+1} \sim a_1^{(2m+1)} B_1 \oplus a_3^{(2m+1)} B_3 \oplus \cdots \oplus a_{2m+1}^{(2m+1)} B_{2m+1}$$

and

$$(2.5) \quad N_{2m} \sim a_0^{(2m)} B_0 \oplus a_2^{(2m)} B_2 \oplus \cdots \oplus a_{2m}^{(2m)} B_{2m},$$

where for $j \geq i \geq 0$ the numbers $a_i^{(j)}$ are integers satisfying

$$(2.6) \quad a_j^{(j)} = 1, \quad a_i^{(j)} = a_{i-1}^{(j-1)} + a_{i+1}^{(j-1)} \quad \text{if } i+j \text{ is even,}$$

$$a_i^{(j)} = 0 \quad \text{if } i+j \text{ is odd.}$$

The numbers $a_i^{(j)}$ are uniquely determined by the above conditions. The values are tabulated in Table 2.1.

We derive the following closed formula for the numbers $a_i^{(j)}$ in terms of binomial coefficients $\binom{r}{s}$. As usual we let $\binom{r}{s} = 0$ when $s < 0$ or $s > r$, and $\binom{0}{0} = 1$.

PROPOSITION 2.2. *For $j \geq i \geq 0$, the following formulas hold:*

$$(2.7) \quad a_i^{(j)} = \binom{j-1}{\frac{j-i}{2}} - \binom{j-1}{\frac{j-i}{2}-2}, \quad i+j \text{ is even,}$$

TABLE 2.1
Values for $a_i^{(j)}$.

$j \setminus i$	0	1	2	3	4	5	6	7	...
1		1							
2	1		1						
3		2		1					
4	2		3		1				
5		5		4		1			
6	5		9		5		1		
7		14		14		6		1	
8	14		28		20		7		1
⋮									

and $a_i^{(j)} = 0$, otherwise.

Proof. First observe that $a_j^{(j)} = 1 = \binom{0}{0} - \binom{0}{-2}$. Since the numbers $a_i^{(j)}$ are uniquely determined by the recurrence relation (2.6), it suffices to show that our proposed formula in (2.7) satisfies (2.6) if $i + j$ is even. Using the Pascal identity on binomial coefficients $\binom{k}{l-1} + \binom{k}{l} = \binom{k+1}{l}$, we have

$$\begin{aligned} a_{i-1}^{(j-1)} + a_{i+1}^{(j-1)} &= \binom{j-2}{\frac{j-i}{2}} - \binom{j-2}{\frac{j-i}{2}-2} + \binom{j-2}{\frac{j-i}{2}-1} - \binom{j-2}{\frac{j-i}{2}-3} \\ &= \binom{j-1}{\frac{j-i}{2}} - \binom{j-1}{\frac{j-i}{2}-2} = a_i^{(j)}. \quad \square \end{aligned}$$

Our main result is the following.

THEOREM 2.3. *Suppose n is a positive integer. We have*

$$(2.8) \quad b(A_{n+1}, C_{n+1}) \geq 2b(A_n, C_n).$$

Moreover, if $n = 2m + 1$ is odd, then

$$(2.9) \quad b(A_n, C_n) \geq 2 \sum_{k=0}^m \left\{ \left(\binom{2m}{k} - \binom{2m}{k-2} \right) \left(\sum_{q=1}^{m+1-k} \sqrt{2(m+1-k-q)+1} \sqrt{2q-1} \right) \right\}.$$

Proof. Suppose $U = \begin{pmatrix} U_{11} & U_{12} \\ U_{21} & U_{22} \end{pmatrix}$ is a $2^{n+1} \times 2^{n+1}$ unitary matrix such that $U_{11}, U_{12}, U_{21}, U_{22}$ are $2^n \times 2^n$ matrices, and $b(A_n, C_n) = |\text{tr}(C_n^* U A_n U^*)|$. Let

$$\tilde{U} = \begin{pmatrix} U_{11} & 0 & U_{12} & 0 \\ 0 & U_{11} & 0 & U_{12} \\ U_{21} & 0 & U_{22} & 0 \\ 0 & U_{21} & 0 & U_{22} \end{pmatrix}.$$

Then

$$b(A_{n+1}, C_{n+1}) \geq |\text{tr}(C_{n+1}^* \tilde{U} A_{n+1} \tilde{U}^*)| = 2b(A_n, C_n).$$

So, (2.8) holds.

Next, consider the $2r \times 2r$ matrix

$$L_r = \begin{pmatrix} 0 & 0 \\ \mathbb{I}_r & 0 \end{pmatrix}.$$

Then $C_n = L_{2^n}$ is unitarily similar to $L_{r_1} \oplus \cdots \oplus L_{r_k}$ as long as $r_1 + \cdots + r_k = 2^n$. In particular, we have

$$C_n \sim C_{n-1} \oplus C_{n-1}$$

and

$$C_{n-1} \sim a_1^{(2m+1)} L_1 \oplus a_3^{(2m+1)} L_2 \oplus \cdots \oplus a_{2m+1}^{(2m+1)} L_{m+1}.$$

Thus, combining this with (2.4),

$$(2.10) \quad \begin{aligned} b(A_n, C_n) &\geq 2b(N_n, C_{n-1}) \\ &\geq 2[a_1^{(2m+1)} b(B_1, L_1) + a_3^{(2m+1)} b(B_3, L_2) + \cdots + a_{2m+1}^{(2m+1)} b(B_{2m+1}, L_{m+1})]. \end{aligned}$$

Also, observe that by a permutation

$$(2.11) \quad B_{2m+1} \sim \begin{pmatrix} 0 & S_m \\ T_m & 0 \end{pmatrix},$$

where T_m and S_m are the $(m+1) \times (m+1)$ matrices

$$T_m = \begin{pmatrix} \sqrt{2m+1}\sqrt{1} & & & \circ \\ & \sqrt{2m-1}\sqrt{3} & & \\ & & \ddots & \\ \circ & & & \sqrt{1}\sqrt{2m+1} \end{pmatrix},$$

$$S_m = (0) \oplus \begin{pmatrix} \sqrt{2m}\sqrt{2} & & & \circ \\ & \sqrt{2m-2}\sqrt{4} & & \\ & & \ddots & \\ \circ & & & \sqrt{2}\sqrt{2m} \end{pmatrix}.$$

Thus

$$(2.12) \quad b(B_{2m+1}, L_{m+1}) \geq \sum_{q=1}^{m+1} \sqrt{2(m+1-q)+1} \sqrt{2q-1}.$$

Combining (2.10) with (2.12) yields the lower bound in (2.9). $\square$

An alternative way to compute the quantity on the right-hand side of (2.9) is the following. Compute the singular values of the matrix A_n . These numbers are square roots of even and odd integers. When one adds up all the singular values that are square roots of odd integers, one arrives at the quantity on the right-hand side of (2.9). For the unitary matrix one may take either the unitary matrix consisting of the left singular vectors of A_n or the unitary matrix consisting of the right singular vectors of A_n . As it turns out, a permutation connects the left and right singular vectors. Indeed, this follows immediately from the fact that A_n is unitarily similar to a direct

sum decomposition of B_j 's (see (2.4) and (2.5)) and from the unitary similarity given in (2.11).

Keeping track of the unitary similarities to achieve the bound in (2.9) gives a way to construct the corresponding unitary U . For instance, when $n = 3$, we have $N_3 = \begin{pmatrix} N_2 & 0 \\ \mathbb{I} & N_2 \end{pmatrix}$. As

$$N_2 = \begin{pmatrix} B_1 & 0 \\ \mathbb{I} & B_1 \end{pmatrix} = U_1(B_0 \oplus B_2)U_1^*,$$

we get that

$$\begin{aligned} N_3 &= (U_1 \oplus U_1) \begin{pmatrix} B_0 & 0 & 0 & 0 \\ 0 & B_2 & 0 & 0 \\ \mathbb{I} & 0 & B_0 & 0 \\ 0 & \mathbb{I} & 0 & B_2 \end{pmatrix} (U_1^* \oplus U_1^*) \\ &= (U_1 \oplus U_1) Y_2 \begin{pmatrix} B_0 & 0 & 0 & 0 \\ \mathbb{I} & B_0 & 0 & 0 \\ 0 & 0 & B_2 & 0 \\ 0 & 0 & \mathbb{I} & B_2 \end{pmatrix} Y_2^* (U_1^* \oplus U_1^*) \\ &= (U_1 \oplus U_1) Y_2 (U_0 \oplus U_2) (2B_1 \oplus B_3) (U_0^* \oplus U_2^*) Y_2^* (U_1^* \oplus U_1^*), \end{aligned}$$

where

$$Y_k = \begin{pmatrix} \mathbb{I}_k & 0 & 0 & 0 \\ 0 & 0 & \mathbb{I}_k & 0 \\ 0 & \mathbb{I}_k & 0 & 0 \\ 0 & 0 & 0 & \mathbb{I}_k \end{pmatrix}.$$

Furthermore, $C_3 = Y_4(C_2 \oplus C_2)Y_4^*$, $C_2 = V(2L_1 \oplus L_2)V^*$, $Y_1L_2Y_1^* = L_1 \oplus L_1$, where

$$V = \begin{pmatrix} 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \end{pmatrix}.$$

So

$$\mathrm{tr}(C_3^* U A_3 U^*) = 4\sqrt{3} + 4$$

for

$$U = Y_4(V \oplus V)(\mathbb{I}_4 \oplus Y_1^* \oplus \mathbb{I}_4 \oplus Y_1^*)(U_0^* \oplus U_2^* \oplus U_0^* \oplus U_2^*)(Y_2^* \oplus Y_2^*)4U_1^*.$$

It is still a challenge to give an easy formula for the unitary matrix in the general case. By the way, it is interesting to note that in [5] the matrix $2B_1 \oplus B_3$ appears (up to a permutation) as the matrix O_- .

3. Open problems. As mentioned in the introduction, the right-hand side of (2.9) coincides with the numerical computation of $b(A_n, C_n)$, $n = 1, \dots, 6$, in [4]. In addition, it is straightforward to check that the U attaining the right-hand side of (2.9) satisfies the optimality condition of Lagrange multipliers that

$$C_n U A_n U^* - U A_n U^* C_n$$

is Hermitian. These observations lead us to believe that equality holds in (2.9). To prove this it remains to prove the inequality “ $\leq$ ” in (2.9).

From a practical NMR viewpoint it is a very important question whether the optimal U_n can actually be realized in experiments. Indeed, in NMR spectroscopy there are a limited number of manipulations that one can perform that mathematically result in a unitary transformation. In other words, only a limited number of unitary matrices correspond to physically possible experiments. The possible unitaries are formed by cascades of operations e^{iH} , where the self-adjoint matrix H can be one of the following:

$$\begin{aligned} H_{rf} &= aI_{kx} + bI_{ky}, H_{Jhe} = 2cI_{kz}S_z, H_{Jho} = c(I_{kz}S_z + I_{kx}S_x + I_{ky}S_y), \\ H_{shift} &= dI_{kz} + eS_z, H_{dhe} = 2fI_{kz}S_z, H_{dho} = g(2I_{kz}S_z - I_{kx}S_x - I_{ky}S_y), \\ H_{planar} &= h(2I_{kx}S_x + sI_{ky}S_y), H_{iso} = i(2I_{kx}S_x + 2I_{ky}S_y + 2I_{kz}S_z), \end{aligned}$$

with

$$S_\alpha = I_\alpha \otimes \mathbb{I} \otimes \dots \otimes \mathbb{I}, \quad I_{k\alpha} = \mathbb{I} \otimes \dots \otimes \mathbb{I} \otimes I_\alpha \otimes \mathbb{I} \otimes \dots \otimes \mathbb{I}, \quad \alpha \in \{x, y, z\},$$

where in the last equality I_α appears in the $(k+1)$ st position. Here $\otimes$ stands for the Kronecker product, and

$$I^+ = \begin{pmatrix} 0 & 1 \\ 0 & 0 \end{pmatrix}, I^- = \begin{pmatrix} 0 & 0 \\ 1 & 0 \end{pmatrix}, I_x = \frac{1}{2} \begin{pmatrix} 0 & 1 \\ 1 & 0 \end{pmatrix}, I_y = \frac{i}{2} \begin{pmatrix} 0 & -1 \\ 1 & 0 \end{pmatrix}, I_z = \frac{1}{2} \begin{pmatrix} 1 & 0 \\ 0 & -1 \end{pmatrix}.$$

These choices of H correspond to typical experiments, but other experiments may be possible as well. Note that the above operations yield a subgroup of the unitaries.

We also would like to mention the problem of finding a physical explanation for our result. As remarked earlier, the matrices A_n have singular values that are square roots of integers. In the case that n is odd, half of them are square roots of odd integers, and one simply needs to add all of these to obtain the right-hand side of (2.9). Is there a physical explanation for this phenomenon?

Finally, the physical problem we discussed in this paper concerns the transformation from the so-called -1 quantum coherence of the I spins to the -1 quantum coherence of the S spin (see [3]). Other transformations are of interest as well; namely, in [3] the transfer between $A = F^-$ and $C = 2F_z S^-$ is mentioned. This leads to the same A_n 's as before, but now C_n is given by

$$C_n = \begin{pmatrix} 0 & 0 \\ M_n & 0 \end{pmatrix},$$

where $M_0 = 0$ and

$$M_j = M_{j-1} \otimes \mathbb{I}_2 + \mathbb{I}_{2^{j-1}} \otimes \begin{pmatrix} 1 & 0 \\ 0 & -1 \end{pmatrix}.$$

Also in this case one would like to determine $b(A_n, C_n)$.

Note added in proof. After the acceptance of the paper, we learned about the paper of Dirr, Helmke, and Kleinstaubler [1], where the same problem is treated.

Acknowledgments. We wish to thank Professors John Delos (The College of William and Mary) and Robert L. Vold (The College of William and Mary) for useful discussions involving the underlying physics, and N. C. Nielsen (University of Aarhus) for commenting on our manuscript and mentioning the open problem of checking whether the unitary we found can be implemented experimentally. Moreover, Professor Uwe Helmke (University of Würzburg) was helpful in providing us with his manuscript [4].

REFERENCES

- [1] G. DIRR, U. HELMKE, AND M. KLEINSTEUBER, *Lie algebra representations, nilpotent matrices, and the C -numerical range*, Linear Algebra Appl., to appear.
- [2] R. R. ERNST, G. BODENHAUSEN, AND A. WOKAUN, *Principles of Nuclear Magnetic Resonance in One and Two Dimensions*, Oxford University Press, New York, 1987.
- [3] S. J. GLASER, T. SCHULTE-HERBRÜGGEN, M. SIEVEKING, O. SCHEDLETZKY, N. C. NIELSEN, O. W. SORESENSEN, AND C. GRIESINGER, *Unitary control in quantum ensembles: Maximizing signal intensity in coherent spectroscopy*, Science, 280 (1998), pp. 421–424.
- [4] U. HELMKE, K. HÜPER, J. B. MOORE, AND T. SCHULTE-HERBRÜGGEN, *Gradient flows computing the C -numerical range with applications in NMR spectroscopy*, J. Global Optim., 23 (2002), pp. 283–308.
- [5] N. C. NIELSEN AND O. W. SORESENSEN, *Conditional bounds on polarization transfer*, J. Magn. Reson. Ser. A, 114 (1995), pp. 24–31.
- [6] O. W. SORESENSEN, *Polarization transfer experiments in high-resolution NMR spectroscopy*, Progress in NMR Spectroscopy, 21 (1989), pp. 503–569.
- [7] J. STOUSTRUP, O. SCHEDLETZKY, S. J. GLASER, C. GRIESINGER, N. C. NIELSEN, AND O. W. SORESENSEN, *Generalized bound on quantum dynamics: Efficiency of unitary transformations between non-Hermitian states*, Phys. Rev. Lett., 74 (1995), pp. 2921–2924.

COMPUTING THE CONDITION NUMBER OF TRIDIAGONAL AND DIAGONAL-PLUS-SEMISEPARABLE MATRICES IN LINEAR TIME*

GARETH I. HARGREAVES†

Abstract. For an $n \times n$ tridiagonal matrix we exploit the structure of its QR factorization to devise two new algorithms for computing the 1-norm condition number in $O(n)$ operations. The algorithms avoid underflow and overflow, and are simpler than existing algorithms since tests are not required for degenerate cases. An error analysis of the first algorithm is given, while the second algorithm is shown to be competitive in speed with existing algorithms. We then turn our attention to an $n \times n$ diagonal-plus-semiseparable matrix, A , for which several algorithms have recently been developed to solve $Ax = b$ in $O(n)$ operations. We again exploit the QR factorization of the matrix to present an algorithm that computes the 1-norm condition number in $O(n)$ operations.

Key words. condition number, tridiagonal matrix, diagonal-plus-semiseparable matrix, QR factorization

AMS subject classifications. 15A09, 65F35, 15A12

DOI. 10.1137/040606636

1. Introduction. Consider a nonsingular matrix $A \in \mathbb{R}^{n \times n}$ and the linear system $Ax = b$. The condition number of A ,

$$\kappa(A) = \|A\| \|A^{-1}\|,$$

is often computed or estimated since it provides a measure of the sensitivity of the solution to perturbations in A and b . The condition number depends on the choice of matrix norm. We will consider the matrix 1-norm

$$\|A\|_1 = \max_j \sum_i |a_{ij}|.$$

Various techniques exist for estimating the condition number of a general matrix in $O(n^2)$ operations, given a suitable factorization of the matrix.

Some matrices with a special structure allow the linear system to be solved in $O(n)$ operations rather than the $O(n^3)$ operations required for a general matrix. Techniques for estimating the condition number of such matrices typically reduce to $O(n)$ operations. However, the structure of the inverse may make it possible to compute the condition number exactly in $O(n)$ operations. Two types of matrices for which this can be achieved are tridiagonal and diagonal-plus-semiseparable (dpss) matrices.

Tridiagonal matrices occur in many areas of numerical analysis. The inverse of a tridiagonal matrix is a semiseparable matrix, which is the sum of the strictly upper triangular part of a rank-1 matrix and the lower triangular part of another rank-1 matrix. Another link between tridiagonal and semiseparable matrices is that a symmetric matrix can be reduced to either of these forms using orthogonal transformations. This has led to new algorithms for solving the symmetric eigenvalue problem by reduction to semiseparable form [4] instead of tridiagonal form.

*Received by the editors April 13, 2004; accepted for publication (in revised form) by I. S. Dhillon April 27, 2005; published electronically January 27, 2006. This work was supported by an Engineering and Physical Sciences Research Council Ph.D. Studentship.

<http://www.siam.org/journals/simax/27-3/60663.html>

†School of Mathematics, University of Manchester, Manchester M13 9PL, England (hargreaves@ma.man.ac.uk, <http://www.ma.man.ac.uk/~hargreaves/>).

A dpss matrix is the sum of a diagonal matrix and a semiseparable matrix. Recently several algorithms have been developed to solve $Ax = b$, where A is a dpss matrix in $O(n)$ operations [2], [6], [8], [17]. Applications of solving linear systems of this form include boundary value problems for ordinary differential equations [10], [16], [18], integral equations [15], and orthogonal rational functions [3].

The semiseparable structure of the inverse of a tridiagonal matrix is exploited by Higham to give several algorithms for computing the 1-norm of the inverse in $O(n)$ operations. However, the algorithms for a general tridiagonal matrix are prone to underflow and overflow. The algorithm for the positive definite case, which is implemented in LAPACK [1], is shown not to have such problems.

Dhillon [7] describes four algorithms based on LDU factorizations. The first of these is less prone to underflow and overflow than Higham's algorithms but has restrictions on the tridiagonal matrix. These restrictions are eliminated using IEEE arithmetic [14] in the second algorithm, and various tests to guard against underflow and overflow problems are added to give the third and fourth algorithms. The resulting codes are complicated as they contain various tests to deal with degenerate cases.

Unlike tridiagonal matrices, we know of no existing methods for computing the condition number of a dpss matrix exactly in $O(n)$ operations.

In section 2, we use the properties of the QR factorization of a tridiagonal matrix to present two new algorithms for computing the 1-norm of the inverse of a tridiagonal matrix in $O(n)$ operations. The algorithms are more elegant and simpler than Dhillon's algorithms since they do not require tests for degenerate cases. Numerical experiments show that the second algorithm is marginally slower than the quickest of Dhillon's algorithms, but is faster than his recommended algorithm. A rounding error analysis of the first algorithm is given.

In section 3, we extend the techniques developed for the tridiagonal case to the dpss case to present an algorithm that computes the 1-norm of a dpss matrix in $O(n)$ operations. The condition number of a dpss matrix can instead be estimated in $O(n)$ operations by adapting the LAPACK [1] condition number estimator to take advantage of the structure of the dpss matrix; we show, however, that our new algorithm is quicker than the possibly inaccurate estimate.

2. Tridiagonal matrices. Let the tridiagonal matrix

$$(2.1) \quad T = \begin{bmatrix} \alpha_1 & \gamma_1 & & & \\ \beta_1 & \alpha_2 & \gamma_2 & & \\ & \beta_2 & \ddots & \ddots & \\ & & \ddots & \alpha_{n-1} & \gamma_{n-1} \\ & & & \beta_{n-1} & \alpha_n \end{bmatrix} \in \mathbb{R}^{n \times n}.$$

Since we can solve a tridiagonal system in $O(n)$ operations, we would also like to compute the condition number at the same cost. Computing $\|T\|_1$ can trivially be done in $O(n)$ operations so the problem remains to compute $\|T^{-1}\|_1$ in $O(n)$ operations.

We present two new algorithms for computing $\|T^{-1}\|_1$ in $O(n)$ operations. The algorithms use the properties of QR factorizations of tridiagonal matrices and extend some of the ideas of Bini, Gemignani, and Tisseur [5] for computing the trace of the inverse of a tridiagonal matrix. The new algorithms attempt to avoid underflow and overflow without the need for tests to deal with degenerate cases as in [7].

2.1. A new algorithm to compute $\|\mathbf{T}^{-1}\|_1$. The QR factorization of T in (2.1) can be obtained using $n - 1$ Givens rotations, G_i , so that

$$(2.2) \quad Q^T T = R \quad \text{and} \quad Q^T = G_{n-1} \dots G_2 G_1,$$

where R is upper triangular. The Givens rotation G_i is equal to the identity except for rows and columns i and $i + 1$, where

$$(2.3) \quad G_i([i, i + 1], [i, i + 1]) = \begin{bmatrix} \phi_i & \psi_i \\ -\psi_i & \phi_i \end{bmatrix}, \quad \phi_i^2 + \psi_i^2 = 1.$$

Since T is tridiagonal, the upper triangular matrix R is of the form

$$R = \begin{bmatrix} r_1 & s_1 & t_1 & & \\ & \ddots & \ddots & \ddots & \\ & & r_{n-2} & s_{n-2} & t_{n-2} \\ & & & r_{n-1} & s_{n-1} \\ & & & & r_n \end{bmatrix}.$$

The following theorem of [9] describes the structure of Q^T , which allows us to compute the elements of T^{-1} in $O(n)$ operations.

THEOREM 2.1. *Let $T \in \mathbb{R}^{n \times n}$ be tridiagonal and unreduced and let $T = QR$ be its QR factorization computed according to (2.2). Define*

$$(2.4) \quad \begin{aligned} D &= \text{diag}(1, -\psi_1, \psi_1\psi_2, \dots, (-1)^{n-1}\psi_1\psi_2 \dots \psi_{n-1}), \\ u &= D^{-1}[1, \phi_1, \phi_2, \dots, \phi_{n-1}]^T, \\ v &= D[\phi_1, \phi_2, \dots, \phi_{n-1}, 1]^T. \end{aligned}$$

Then

$$(2.5) \quad Q^T = \begin{bmatrix} v_1 u_1 & \psi_1 & & & 0 \\ v_2 u_1 & v_2 u_2 & \psi_2 & & \\ \vdots & \vdots & \ddots & \ddots & \\ \vdots & & & v_{n-1} u_{n-1} & \psi_{n-1} \\ v_n u_1 & v_n u_2 & \dots & v_n u_{n-1} & v_n u_n \end{bmatrix}.$$

From Theorem 2.1, the (i, j) element of T^{-1} , η_{ij} , is given by

$$\eta_{ij} = e_i^T T^{-1} e_j = e_i^T R^{-1} Q^T e_j = u_j e_i^T R^{-1} v, \quad i \geq j.$$

Defining w as the solution of $Rw = v$, we have

$$(2.6) \quad \eta_{ij} = u_j e_i^T w = u_j w_i, \quad i \geq j.$$

We can therefore find the elements of T^{-1} in the lower triangle using (2.6).

We note that we can obtain the strictly upper triangular part of T^{-1} by applying the above procedure to

$$\tilde{T} = \begin{bmatrix} \alpha_n & \beta_{n-1} & & & \\ \gamma_{n-1} & \alpha_{n-1} & \beta_{n-2} & & \\ & \gamma_{n-2} & \ddots & \ddots & \\ & & \ddots & \alpha_2 & \beta_1 \\ & & & \gamma_1 & \alpha_1 \end{bmatrix},$$

so that

$$\tilde{T}^{-1} = \begin{bmatrix} \eta_{nn} & & \cdots & \eta_{n1} \\ & \ddots & & \vdots \\ \vdots & & \eta_{22} & \eta_{21} \\ \eta_{1n} & \cdots & \eta_{12} & \eta_{11} \end{bmatrix}.$$

The QR factorization of $\tilde{T}$ (which is essentially the QL factorization of T) is given by $\tilde{T} = \tilde{Q}\tilde{R}$, where $\tilde{Q}$ is made up of $n-1$ Givens rotations and $\tilde{R}$ is upper triangular. By Theorem 2.1, the lower triangular part of $\tilde{Q}^T$ is determined by vectors $\tilde{u}$ and $\tilde{v}$, where the components of $\tilde{u}$ and $\tilde{v}$ are defined by the $n-1$ Givens rotations. Hence the (i, j) element of $\tilde{T}^{-1}$ is given by

$$\tilde{\eta}_{ij} = e_i^T \tilde{T}^{-1} e_j = e_i^T \tilde{R}^{-1} \tilde{Q}^T e_j = \tilde{u}_j e_i^T \tilde{R}^{-1} \tilde{v} = \tilde{u}_j e_i^T \tilde{w} = \tilde{u}_j \tilde{w}_i, \quad i \geq j,$$

where $\tilde{w}$ is the solution of $\tilde{R}\tilde{w} = \tilde{v}$.

The (i, j) element of T^{-1} , $i < j$, is therefore given by

$$\eta_{ij} = \tilde{u}_{n-j+1} \tilde{v}_{n-i+1}.$$

We now return to the elements of T^{-1} in the lower triangle. The computation of u and v using (2.4) may cause underflow and overflow since the diagonal entries of D are products of ψ_i with $|\psi_i| \leq 1$. In [5] it was noted that a way of avoiding such problems for computing the diagonal elements of T^{-1} is to scale the triangular system $Rw = v$ with the diagonal matrix D . This gives $R'w' = v'$, where

$$(2.7) \quad R' = D^{-1}RD, \quad w' = D^{-1}w, \quad v' = D^{-1}v = [\phi_1, \dots, \phi_{n-1}, 1]^T.$$

Using this scaling avoids products of ψ_i . The entries of R' are given by

$$(2.8) \quad r'_i = r_i, \quad s'_i = -\psi_i s_i, \quad t'_i = \psi_i \psi_{i+1} t_i,$$

and since $|\psi_i| \leq 1$, these values cannot overflow.

Let

$$(2.9) \quad u' = [1, \phi_1, \phi_2, \dots, \phi_{n-1}]^T.$$

As $\phi_i^2 + \psi_i^2 = 1$, the components of u' are all bounded by 1 in modulus. Since $w = Dw'$ and $u = D^{-1}u'$, the diagonal elements of T^{-1} are given by

$$\eta_{ii} = u_i w_i = u'_i w'_i, \quad i = 1:n.$$

The strictly lower triangular elements are given by

$$\eta_{ij} = u_j w_i = \frac{d_i}{d_j} u'_j w'_i, \quad i > j,$$

and hence

$$|\eta_{ij}| = |\psi_j \dots \psi_{i-1} u'_j w'_i|, \quad i > j,$$

for which we clearly have possible underflow problems if η_{ij} is computed in this way.

Fortunately, this problem can be overcome by considering how to compute the 1-norm of the j th column of the strictly lower triangular part of T^{-1} , which is given by

$$\sigma_j = |u'_j|(|w'_{j+1}\psi_j| + |w'_{j+2}\psi_j\psi_{j+1}| + \cdots + |w'_n\psi_j \cdots \psi_{n-1}|), \quad j = 1:n-1.$$

If σ_j is computed in this way, $O(n^2)$ operations would be required to compute σ_j for $j = 1:n-1$. The underflow problem can be overcome and the operation count reduced by an order of magnitude using nested multiplication. We can rewrite σ_j as

$$\sigma_j = |u'_j|(((\cdots((|\psi_{n-1}w'_n| + |w'_{n-1}|)|\psi_{n-2}| + |w'_{n-2}|)\cdots)|\psi_{j+1}| + |w'_{j+1}|)|\psi_j|), \\ j = 1:n-1.$$

Therefore, σ_j can be computed using

$$\begin{aligned} & 1 \quad \sigma'_{n-1} = |\psi_{n-1}w'_n| \\ & 2 \quad \text{for } j = n-1:-1:2 \\ (2.10) \quad & 3 \quad \sigma'_{j-1} = (\sigma'_j + |w'_j|)|\psi_{j-1}| \\ & 4 \quad \text{end} \\ & 5 \quad \text{for } j = 1:n-1, \sigma_j = |u'_j|\sigma'_j, \text{ end} \end{aligned}$$

This method of computing σ_j avoids forming explicit products of ψ_i and allows σ_j , $j = 1:n-1$, to be computed in $O(n)$ operations. For underflow to occur $\sigma_j/|u'_j|$ must underflow, and since $\sigma_j/|u'_j| \geq \sigma_j$, this can occur only if σ_j , the 1-norm of the j th column of the strictly lower triangular part of T^{-1} , underflows. We note that this is very unlikely and that underflow does not cause the algorithm to break down.

When considering $\tilde{T}$, we can use a similar approach that also adds the absolute values of the diagonal elements of T^{-1} . This enables us to compute the 1-norm of the j th column of the upper triangular part of T^{-1} , denoted by $\tilde{\sigma}_j$, in $O(n)$ operations without the risk of underflow. Defining $\sigma_n = 0$ the 1-norm of T^{-1} is then given by

$$\|T^{-1}\|_1 = \max_{j=1:n}(\sigma_j + \tilde{\sigma}_j).$$

The pseudocode in the following algorithm summarizes the method described above. The function $(\phi, \psi, r) = \text{Givens}(a, b)$ computes $r = \sqrt{a^2 + b^2}$, $\phi = a/r$, and $\psi = b/r$, and guards against overflow.

ALGORITHM 2.1. Computes $\tau = \|T^{-1}\|_1$, where T is given by (2.1).

$$\begin{aligned} & 1 \quad \text{for } k = 1:2 \\ & 2 \quad a = \alpha_1, g = \gamma_1, u'_1 = 1 \\ & 3 \quad \text{for } i = 1:n-1 \\ & 4 \quad (\phi_i, \psi_i, r'_i) = \text{Givens}(a, \beta_i) \\ & 5 \quad \text{if } r'_i = 0, \tau = \infty, \text{ return, end} \\ & 6 \quad s'_i = -\psi_i(\phi_i g + \psi_i \alpha_{i+1}) \\ & 7 \quad a = -\psi_i g + \phi_i \alpha_{i+1}, u'_{i+1} = \phi_i, v'_i = \phi_i \\ & 8 \quad \text{if } i < n-1, t_i = \psi_i^2 \gamma_{i+1}, g = \phi_i \gamma_{i+1}, \text{ end} \\ & 9 \quad \text{if } i > 1, t'_{i-1} = \psi_i t'_{i-1}, \text{ end} \\ & 10 \quad \text{end} \\ & 11 \quad r'_n = a, v'_n = 1 \\ & 12 \quad \text{if } r'_n = 0, \tau = \infty, \text{ return, end} \\ & 13 \quad w'_n = v'_n / r'_n \end{aligned}$$

```

14    $w'_{n-1} = (v'_{n-1} - w'_n s'_{n-1})/r'_{n-1}$ 
15   for  $i = n-2:-1:1$ 
16        $w'_i = (v'_i - w'_{i+1} s'_i - w'_{i+2} t'_i)/r'_i$ 
17   end
18   if  $k = 1$ 
19       Compute  $\sigma_j$  using the code given in (2.10)
20        $\alpha = \alpha(n:-1:1)$ ,  $\delta = \beta$ ,  $\beta = \gamma(n-1:-1:1)$ ,  $\gamma = \delta(n-1:-1:1)$ 
21   else
22        $\tilde{\sigma}_n = |w'_n|$ 
23       for  $i = n-1:-1:1$ 
24            $\tilde{\sigma}_i = \tilde{\sigma}_{i+1} |\psi_i| + |w'_i|$ 
25       end
26       for  $i = 1:n-1$ ,  $\tilde{\sigma}_i = \tilde{\sigma}_i |u'_i|$ , end
27   end
28 end
29  $\tau = \max_{i=1:n} (\sigma_i + \tilde{\sigma}_{n-i+1})$ 

```

Cost. Assuming the function *Givens* requires 6 operations, the total cost is $51n$ operations.

Lines 2–12 compute the vectors r' , s' , t' in (2.8), u' in (2.9), and v' in (2.7) and lines 13–17 solve the linear system $R'w' = v'$. The 1-norms of the columns of the strictly lower and the upper triangular parts of T^{-1} are computed in lines 19 and 22–26, respectively. The only divisions in the algorithm are by r_i . Hence the test on line 5 prevents division by zero, which is possible if T is singular.

Algorithm 2.1 does not have to deal with the reduced case separately as in [12], or have numerous tests to deal with the reduced case as in [7]. It can also be easily adapted to compute $\|T^{-1}\|_1$ for $T \in \mathbb{C}^{n \times n}$ by using complex Givens rotations and the complex conjugate of v' in (2.7).

We note that there are certain similarities between Algorithm 2.1 and those in [7]. All the algorithms use factorizations: triangular factorizations in [7] and orthogonal factorizations in Algorithm 2.1. Also, Algorithm 2.1 and the algorithms in [7] both compute two factorizations, with the first proceeding from the top of T to the bottom and the second from the bottom of T to the top. In Algorithm 2.1 this takes the form of a QR factorization followed by a QL factorization, whereas in [7] the factorizations $T = L_+ D_+ U_+$ and $T = L_- D_- U_-$ are computed, where L_+ and L_- are lower bidiagonal, U_+ and U_- upper bidiagonal, and D_+ and D_- diagonal.

2.2. Reducing the operation count. The matrix T in (2.1) is said to be unreduced if $\beta_i \gamma_i \neq 0$ for $i = 1:n-1$. If T is unreduced, then it can be scaled using the diagonal matrix

$$D = \text{diag}(d_i), \quad d_1 = 1, \quad d_i = \frac{\beta_1 \cdots \beta_{i-1}}{\gamma_1 \cdots \gamma_{i-1}}, \quad i > 1,$$

so that $\tilde{T} = TD$ is symmetric. In this section we will use this property to obtain an algorithm for computing $\|T^{-1}\|_1$ that requires approximately half the number of operations of Algorithm 2.1.

We first note that it is also possible to use a similar scaling so that $\check{T} = \check{D}T$ is symmetric, where $\check{D}$ is diagonal and is defined by products of γ_i/β_i . A naive method of computing $\|T^{-1}\|_1$ would be to use this scaling to form $\check{T}$, compute the absolute column sums of $\check{T}^{-1}$ using the ideas of section 2.1 and making use of its symmetry, and then recover the absolute column sums of T^{-1} using $T^{-1} = \check{T}^{-1}\check{D}$. Explicitly

forming $\tilde{T}$ in this approach can lead to underflow and overflow due to the computation of the elements of $\tilde{D}$. We will use the scaling $\tilde{T} = TD$ without forming $\tilde{T}$ and use the structure of T^{-1} to deal with reduced T .

The QR factorization of $\tilde{T}$ has the form

$$Q^T \tilde{T} = RD,$$

where Q and R satisfy $Q^T T = R$ and can be computed using (2.4). The matrix Q is defined by (2.5). The (i, j) element of $\tilde{T}^{-1}$, $\tilde{\eta}_{ij}$, is therefore given by

$$\tilde{\eta}_{ij} = d_i^{-1} u_j w_i,$$

where w is the solution of the triangular system $Rw = v$.

In section 2.1, we showed how to scale the system $Rw = v$ to avoid underflow and overflow. Using the same procedure here we find that

$$|\tilde{\eta}_{ij}| = |d_i^{-1} \psi_j \dots \psi_{i-1} u'_j w'_i|, \quad i \geq j,$$

where u' is defined by (2.9) and w' is defined by (2.7).

Since $T^{-1} = D\tilde{T}^{-1}$ the (i, j) element of T^{-1} , η_{ij} , satisfies

$$|\eta_{ij}| = |\psi_j \dots \psi_{i-1} u'_j w'_i|, \quad i \geq j,$$

which is precisely what we found in section 2.1. Therefore, the 1-norm of the j th column of the strictly lower triangle of T^{-1} , σ_j , can be calculated as described in section 2.1.

Using the symmetry of $\tilde{T}^{-1}$ the upper triangular elements of $\tilde{T}^{-1}$ satisfy

$$|\tilde{\eta}_{ij}| = |d_j^{-1} \psi_i \dots \psi_{j-1} u'_i w'_j|, \quad i \leq j,$$

and therefore using $T^{-1} = D\tilde{T}^{-1}$,

$$(2.11) \quad \begin{aligned} |\eta_{ij}| &= \left| \frac{d_i}{d_j} \psi_i \dots \psi_{j-1} u'_i w'_j \right|, \quad i \leq j, \\ &= \left| \frac{\gamma_i \dots \gamma_{j-1}}{\beta_i \dots \beta_{j-1}} \psi_i \dots \psi_{j-1} u'_i w'_j \right|. \end{aligned}$$

Using nested multiplication, similar to that described for computing σ_j , we can compute the 1-norm of the j th column of the upper triangle of T^{-1} , $\tilde{\sigma}_j$, using

$$(2.12) \quad \begin{aligned} &1 \quad \tilde{\sigma}_1 = |u'_1| \\ &2 \quad \text{for } j = 1:n-1 \\ &3 \quad \tilde{\sigma}_{j+1} = \tilde{\sigma}_j |\gamma_j \psi_j / \beta_j| + |u'_{j+1}| \\ &4 \quad \text{end} \\ &5 \quad \text{for } j = 1:n, \tilde{\sigma}_j = |w'_j| \tilde{\sigma}_j, \text{ end} \end{aligned}$$

Computing $\tilde{\sigma}_j$ in this way allows us to make use of the lower triangular part of T^{-1} to find the upper triangular part and therefore approximately halves the number of operations required compared with the method of the previous section. We note that we encounter division by zero when computing $\tilde{\sigma}_j$ if $\beta_j = 0$ and hence the procedure described above is valid only for tridiagonal matrices with all $\beta_j \neq 0$. However, it is

possible to remove this restriction by considering the structure of T when $b_k = 0$ for some k .

If the tridiagonal matrix T has $\beta_k = 0$, then we can write it as

$$T = \begin{bmatrix} T_1 & C \\ 0 & T_2 \end{bmatrix}, \quad T_1 \in \mathbb{R}^{k \times k}, \quad T_2 \in \mathbb{R}^{(n-k) \times (n-k)}, \quad C \in \mathbb{R}^{k \times (n-k)},$$

where $C = \gamma_{k-1}e_k e_1^T$ and e_j denotes the j th unit vector. The inverse of the tridiagonal matrix is then given by

$$T^{-1} = \begin{bmatrix} T_1^{-1} & -T_1^{-1}CT_2^{-1} \\ 0 & T_2^{-1} \end{bmatrix}$$

and

$$(2.13) \quad T_1^{-1}CT_2^{-1} = \gamma_{k-1}(T_1^{-1}e_k)(T_2^{-1}e_1)^T.$$

We note that the k th absolute column sum of xy^T , where $x, y \in \mathbb{R}^n$, is $\|x\|_1|y_k|$ and therefore we can find the absolute column sums of (2.13) using the last absolute column sum of T_1^{-1} and the first row of T_2^{-1} .

Suppose we are using the procedure described in (2.12) to compute $\tilde{\sigma}_j$. Then if we encounter a β_j that is zero at line 3, the absolute last column sum of T_1^{-1} is given by $w'_j\tilde{\sigma}_j$. The absolute values of the entries of the upper triangular part of T_2^{-1} are given by (2.12). Since we know the last absolute column sum of T_1^{-1} and the absolute values of the first row of T_2^{-1} , we can combine these and the absolute values of the upper triangular part of T_2^{-1} to give the 1-norm of the j th column of the upper triangular part of T^{-1} for $j = k+1:n$. This is achieved by setting

$$\tilde{\sigma}_{j+1} = |\tilde{\sigma}_j w_j \gamma_j u'_{j+1}| + |u'_{j+1}|$$

instead of line 3. The $|u'_{j+1}|$ term corresponds to the 1-norm of the upper triangular part of T_2^{-1} and the other term corresponds to the 1-norm of the columns of (2.13). Clearly these ideas apply to T_1^{-1} and T_2^{-1} and, therefore, the procedure will work if there are several β_k equal to zero. We summarize this method of computing the absolute column sums of the upper triangular part of T^{-1} as follows:

$$(2.14) \quad \begin{array}{ll} 1 & \tilde{\sigma}_1 = |u'_1| \\ 2 & \text{for } j = 1:n-1 \\ 3 & \quad \text{if } \beta_j = 0 \\ 4 & \quad \quad \tilde{\sigma}_{j+1} = |\tilde{\sigma}_j w'_j \gamma_j u'_{j+1}| + |u'_{j+1}| \\ 5 & \quad \text{else} \\ 6 & \quad \quad \tilde{\sigma}_{j+1} = \tilde{\sigma}_j |\gamma_j \psi_j / \beta_j| + |u'_{j+1}| \\ 7 & \quad \text{end} \\ 8 & \text{end} \\ 9 & \text{for } j = 1:n, \tilde{\sigma}_j = |w'_j| \tilde{\sigma}_j, \text{ end} \end{array}$$

Defining $\sigma_n = 0$, we have $\|T^{-1}\|_1 = \max_{j=1:n}(\sigma_j + \tilde{\sigma}_j)$. The following algorithm computes $\|T^{-1}\|_1$ using the method described in this section.

ALGORITHM 2.2. *Computes $\tau = \|T^{-1}\|_1$, where T is given by (2.1).*

- 1 Use lines 2–12 and 19 in Algorithm 2.1 to compute σ_i .
- 2 Compute $\tilde{\sigma}_i$ using the code in (2.14).
- 3 $\tau = \max_{i=1:n}(\sigma_i + \tilde{\sigma}_i)$.

Cost. Assuming the function *Givens* requires 6 operations, the total cost is $31n$ operations.

2.3. Rounding error analysis. A rounding error analysis of Algorithm 2.1 is given. We have been unable to give a rounding error analysis of Algorithm 2.2 since when considering the upper triangular part of T^{-1} we obtain error results which are bounded by the norm of $\tilde{T} = TD$, which cannot be bounded a priori. However, extensive numerical experiments suggest that Algorithm 2.2 behaves in a stable way.

We will use the standard model for floating point arithmetic:

$$fl(x \text{ op } y) = (x \text{ op } y)(1 + \delta_1) = \frac{x \text{ op } y}{1 + \delta_2}, \quad |\delta_1|, |\delta_2| \leq u, \quad \text{op} = +, -, /, *,$$

where u is the unit roundoff of the computer. We denote by $\|\cdot\|_F$ the unitarily invariant Frobenius norm and make use of the following lemmas [13].

LEMMA 2.2 (see [13, Lem. 3.1]). *If $|\delta_i| \leq u$ and $\rho_i = \pm 1$ for $i = 1:n$, and $nu < 1$, then*

$$\prod_{i=1}^n (1 + \delta_i)^{\rho_i} = 1 + \theta_n,$$

where

$$|\theta_n| \leq \frac{nu}{1 - nu} =: \gamma_n.$$

We also make use of the constant

$$\tilde{\gamma}_n = \frac{cnu}{1 - cnu},$$

where c denotes a small integer constant whose exact value is unimportant.

LEMMA 2.3 (see [13, Lem. 6.6]). *Let $A, B \in \mathbb{R}^{n \times n}$. If $|A| \leq |B|$, then $\|A\|_F \leq \|B\|_F$.*

LEMMA 2.4. *Let $A \in \mathbb{R}^{n \times n}$. Then $\|A\|_1 \leq \sqrt{n}\|A\|_F$ and $\|A\|_F \leq \sqrt{n}\|A\|_1$.*

Given a tridiagonal matrix $T \in \mathbb{R}^{n \times n}$, the first step of the algorithm is to compute the QR factorization of T . From [13, Thm. 19.10] we have that there exists an orthogonal Q such that the computed upper triangular matrix $\hat{R}$ satisfies

$$(2.15) \quad T + \Delta T = Q\hat{R}, \quad \|\Delta T\|_F \leq \tilde{\gamma}_n \|T\|_F.$$

We make the simplifying assumption that the defining variables of the $n - 1$ Givens rotations, $\phi_1, \dots, \phi_{n-1}$ and $\psi_1, \dots, \psi_{n-1}$, are computed exactly. Using $D = \text{diag}(d_1, \dots, d_n)$, where D is given by (2.4), the components of the computed scaled upper triangular matrix $\hat{R}'$ (2.8) satisfy

$$\begin{aligned} \hat{r}'_i &= \hat{r}_i, \\ \hat{s}'_i &= \frac{d_{i+1}}{d_i} \hat{s}_i (1 + \Delta s_i), \quad |\Delta s_i| \leq \gamma_1, \\ \hat{t}'_i &= \frac{d_{i+2}}{d_i} \hat{t}_i (1 + \Delta t_i), \quad |\Delta t_i| \leq \gamma_2. \end{aligned}$$

We now consider solving the triangular system $\hat{R}'w' = \hat{v}'$ and the errors involved in solving the system using backward substitution. Since we have made the assumption that $\phi_1, \dots, \phi_{n-1}$ are computed exactly, the vectors $v' = D^{-1}v = [\phi_1, \dots, \phi_{n-1}, 1]^T$ and $u' = Du = [1, \phi_1, \dots, \phi_{n-1}]^T$ must also be computed exactly.

The last component of the computed solution to the triangular system therefore satisfies

$$\widehat{r}_n \widehat{w}'_n (1 + \theta_1) = \frac{v_n}{d_n},$$

which simplifies to

$$d_n \widehat{r}_n (1 + \widetilde{\Delta r}_n) \widehat{w}'_n = v_n, \quad |\widetilde{\Delta r}_n| \leq \gamma_1.$$

The computed component $\widehat{w}'_{n-1}$ satisfies

$$\widehat{r}_{n-1} \widehat{w}'_{n-1} (1 + \theta'_1) = \frac{v_{n-1}}{d_{n-1}} - \widehat{w}'_n \frac{d_n}{d_{n-1}} \widehat{s}_{n-1} (1 + \Delta s_{n-1}) (1 + \theta_3),$$

which simplifies to

$$d_{n-1} \widehat{r}_{n-1} (1 + \widetilde{\Delta r}_{n-1}) \widehat{w}'_{n-1} = v_n - \widehat{w}'_n d_n \widehat{s}_{n-1} (1 + \widetilde{\Delta s}_{n-1}),$$

where $|\widetilde{\Delta r}_{n-1}| \leq \gamma_1$ and $|\widetilde{\Delta s}_{n-1}| \leq \gamma_4$.

Similarly for $i = 1:n-2$, $\widehat{w}'_i$ satisfies

$$d_i \widehat{r}_i (1 + \widetilde{\Delta r}_i) \widehat{w}'_i = v_i - \widehat{w}'_{i+1} d_{i+1} \widehat{s}_i (1 + \widetilde{\Delta s}_i) - \widehat{w}'_{i+2} d_{i+2} \widehat{t}_i (1 + \widetilde{\Delta t}_i),$$

where

$$|\widetilde{\Delta r}_i| \leq \gamma_1, \quad |\widetilde{\Delta s}_i| \leq \gamma_4, \quad |\widetilde{\Delta t}_i| \leq \gamma_5.$$

Combining the above results for all the components of $\widehat{w}'$ and using Lemma 2.3 gives

$$(2.16) \quad (\widehat{R} + \widetilde{\Delta R}) D \widehat{w}' = v,$$

where

$$\|\widetilde{\Delta R}\|_F \leq \gamma_5 \|\widehat{R}\|_F \leq \gamma_5 \|T\|_F + O(u^2).$$

By considering the errors in computing σ_j using nested multiplication as described in section 2.1, the computed 1-norm of the j th column of the lower triangular part of T^{-1} is found to satisfy

$$(2.17) \quad \widehat{\sigma}_j = \left(\sum_{i=j}^n |\widehat{\eta}_{ij}| \right) (1 + \gamma_{2n-2}),$$

where

$$\begin{aligned} \widehat{\eta}_{ij} &= (-1)^{i+j} \psi_j \dots \psi_{i-1} \widehat{u}'_j \widehat{w}'_i \\ &= d_i u_j e_i^T \widehat{w}' \\ &= d_i u_j e_i^T D^{-1} (\widehat{R} + \widetilde{\Delta R})^{-1} v \quad \text{using (2.16)} \\ &= e_i^T (\widehat{R} + \widetilde{\Delta R})^{-1} Q^T e_j. \end{aligned}$$

Rearranging (2.15) gives $\widehat{R} = Q^T(T + \Delta T)$ and hence

$$\begin{aligned}\widehat{\eta}_{ij} &= e_i^T(Q^T(T + \Delta T) + \widetilde{\Delta R})^{-1}Q^T e_j \\ &= e_i^T(T + \Delta T + Q\widetilde{\Delta R})^{-1}e_j \\ &= e_i^T(T + \widetilde{\Delta T})^{-1}e_j,\end{aligned}$$

where

$$\|\widetilde{\Delta T}\|_F = \|\Delta T + Q\widetilde{\Delta R}\|_F \leq \widetilde{\gamma}_n \|T\|_F.$$

Using Lemma 2.4 we have

$$\|\widetilde{\Delta T}\|_1 \leq n\widetilde{\gamma}_n \|T\|_1.$$

Using the result [19]

$$\frac{\|A^{-1} - (A + E)^{-1}\|_1}{\|A^{-1}\|_1} \leq \frac{e}{1 - e}, \quad e = \kappa_1(A) \frac{\|E\|_1}{\|A\|_1} < 1,$$

and defining $(\widehat{T}^{-1})_{ij} = \widehat{\eta}_{ij}$ we have

$$\frac{\|\widehat{T}^{-1} - T^{-1}\|_1}{\|T^{-1}\|_1} = \frac{\|(T + \widetilde{\Delta T})^{-1} - T^{-1}\|_1}{\|T^{-1}\|_1} \leq \frac{e}{1 - e}, \quad e = \kappa_1(T) \frac{\|\widetilde{\Delta T}\|_1}{\|T\|_1} < 1.$$

Therefore,

$$(2.18) \quad \|\widehat{T}^{-1} - T^{-1}\|_1 \leq \kappa_1(T) \frac{\|\widetilde{\Delta T}\|_1 \|T^{-1}\|_1}{\|T\|_1} \leq n\widetilde{\gamma}_n \kappa_1(T) \|T^{-1}\|_1.$$

Using (2.17) and (2.18)

$$|\widehat{\sigma}_j - \sigma_j| \approx \left| \sum_{i=j}^n (|\widehat{\eta}_{ij}| - |\eta_{ij}|) \right| \leq \sum_{i=j}^n |\widehat{\eta}_{ij} - \eta_{ij}| \leq n\widetilde{\gamma}_n \kappa_1(T) \|T^{-1}\|_1.$$

A similar result holds for the 1-norm of the columns of the upper triangular part of T^{-1} . Hence if we denote by $\widehat{\tau}$ our approximation to $\|T^{-1}\|_1$, computed in floating point arithmetic, we have

$$(2.19) \quad \frac{|\widehat{\tau} - \|T^{-1}\|_1|}{\|T^{-1}\|_1} \leq 2n\widetilde{\gamma}_n \kappa_1(T).$$

This error result is the best we can expect, since it can be shown that the condition number of computing the condition number is the condition number [11].

2.4. Numerical experiments. We compare the accuracy of our new algorithms against Dhillon's recommended algorithm *nrminv_final* 2, Algorithm 4.2 from [12], and MATLAB's `cond`, which computes the condition number of a matrix in $O(n^3)$ operations. The speed of the new algorithms is tested with Dhillon's *nrminv_final* 2 but also the quicker algorithm *nrminv_final* 1. The test matrices are described in Table 2.1.

The results, given in Table 2.2, show that the new algorithms give the results up to four decimal places of accuracy for all but test matrix 7. Also, the new algorithms

TABLE 2.1
Test matrices.

Matrix type	Description
1	Nonsymmetric random tridiagonal, elements uniformly distributed in $[-1, 1]$.
2	<code>gallery('randsvd', 100, 1e15, 2, 1, 1)</code> in MATLAB, which creates a random tridiagonal matrix with all singular values close to 1 except for one that is of order 10^{-15} , so that the 2-norm condition number is 10^{15} .
3	<code>gallery('randsvd', 100, 1e15, 3, 1, 1)</code> in MATLAB, which creates a random tridiagonal matrix with geometrically distributed singular values and 2-norm condition number 10^{15} .
4	Tridiagonal with $\alpha_i = 10^8$, $\beta_i = \gamma_i = 1$.
5	Tridiagonal with $\alpha_i = 10^{-8}$, $\beta_i = \gamma_i = 1$.
6	<code>gallery('lesp', 100)</code> in MATLAB.
7	<code>gallery('dorr', 100, 1e-4)</code> in MATLAB.
8	Nonsymmetric tridiagonal, elements uniformly distributed in $[-1, 1]$ except β_{50} given by 1×10^{-50} multiplied by a random number in $[-1, 1]$.
9	Nonsymmetric tridiagonal, elements of α and γ uniformly distributed in $[-1, 1]$ and β_i given by 1×10^{-50} multiplied by a random number in $[-1, 1]$ for $i = 1:n$.
10	Symmetric tridiagonal with $\alpha_i = 0$, $\beta_i = \gamma_i = 1$.

TABLE 2.2
Computation of $\kappa_1(T)$ on test matrices. Test matrices 1–9 are of order 100 and test matrix 10 is of order 99.

Matrix type	Higham's algorithm	<i>nrminv_final</i> 2	Algorithm 2.1	Algorithm 2.2	MATLAB's <i>cond</i>
1	2.9946e+03	2.9946e+03	2.9946e+03	2.9946e+03	2.9946e+03
2	NaN	1.4979e+15	1.4979e+15	1.4979e+15	1.4979e+15
3	4.5336e+15	4.5336e+15	4.5336e+15	4.5336e+15	4.5336e+15
4	NaN	1.0000	1.0000	1.0000	1.0000
5	100.0000	100.0000	100.0000	100.0000	100.0000
6	NaN	67.1164	67.1164	67.1164	67.1164
7	2.3712e+19	2.7617e+19	1.9403e+19	2.1895e+19	3.5211e+19
8	1.0568e+04	1.0568e+04	1.0568e+04	1.0568e+04	1.0568e+04
9	NaN	6.1603e+10	6.1603e+10	6.1603e+10	6.1603e+10
10	∞	∞	∞	∞	∞

do not suffer from the underflow and overflow problems of Higham's algorithm, which breaks down for test matrices 2, 4, 6, and 9 due to overflow. All the algorithms correctly detect the singular test matrix 10. The difference in results for test matrix 7 are due to rounding errors, and in fact all the methods tested are inaccurate since the actual condition number is 8.885×10^{24} . As noted in section 2.3, the best error bound we can expect to obtain for computing the condition number is of the form (2.19). In this case the bound (2.19) is approximately 10^{34} , which suggests that the computed condition number may be inaccurate.

Before we consider the times taken by the various algorithms we first consider the cost. For Algorithms 2.1 and 2.2, this depends on how the function *Givens* is implemented. In our experiments we use the LAPACK [1] routine `DLARTG` in place of *Givens*, which requires at most 10 operations but typically 6. The maximum number of operations required by the algorithms for computing $\|T^{-1}\|_1$ are given in Table 2.3.

Table 2.4 shows the times taken for the new algorithms and Dhillon's algorithms to compute $\kappa_1(T)$ for nonsymmetric random tridiagonal matrices. The algorithms were implemented in Fortran 77 and compiled using the NAGWare Fortran 95 compiler with the normal optimization level (-O2) and IEEE arithmetic. The machine used

TABLE 2.3

The maximum number of operations required to compute the 1-norm of the inverse of an $n \times n$ tridiagonal matrix for Dhillon's algorithms and the algorithms presented here.

Algorithm	<i>nrminv_final</i> 1	<i>nrminv_final</i> 2	Algorithm 2.1	Algorithm 2.2
Total	$22n$	$27n$	$59n$	$35n$

TABLE 2.4

Time taken in seconds to compute the 1-norm of the inverse of a random $n \times n$ tridiagonal matrix for various n .

Dimension	10^6	2×10^6	4×10^6	6×10^6	8×10^6
<i>nrminv_final</i> 1	0.31	0.60	1.22	1.78	2.54
<i>nrminv_final</i> 2	0.37	0.73	1.45	2.17	3.08
Algorithm 2.1	0.60	1.20	2.37	3.54	4.87
Algorithm 2.2	0.33	0.65	1.26	1.93	2.72

was a 2010 MHz AMD Athlon machine running Linux.

The results show that *nrminv_final* 1 is the quickest as expected. However, Algorithm 2.2 runs much more quickly than the operation count would suggest, which could be due to the fewer if-statements required. As a result, Algorithm 2.2 is only marginally slower than *nrminv_final* 1. However, Algorithm 2.2 is quicker than *nrminv_final* 2, which is the recommended algorithm in [7] due to less element growth in its computation. Interestingly, Algorithm 2.2 is as quick as *nrminv_final* 1 if both are compiled with no optimization (-O0).

3. Diagonal-plus-semiseparable (dpss) matrices. A semiseparable matrix, S , is the sum of the strictly upper triangular part of a rank-1 matrix and the lower triangular part of another rank-1 matrix,

$$S = \text{tril}(qp^T, 0) + \text{triu}(xy^T, 1).$$

Here, $\text{tril}(A, i)$ denotes a matrix A with the entries above the i th diagonal set to zero, where $i = 0$ is the leading diagonal and $i > 0$, $i < 0$ is above and below the leading diagonal, respectively. Similarly $\text{triu}(A, i)$ denotes A with the entries below the i th diagonal set to zero.

A dpss matrix is the sum of a diagonal matrix and a semiseparable matrix

$$(3.1) \quad A := \text{diag}(z) + S = \begin{bmatrix} p_1 q_1 + z_1 & x_1 y_2 & \cdots & x_1 y_n \\ p_1 q_2 & \ddots & \ddots & \vdots \\ \vdots & \ddots & & x_{n-1} y_n \\ p_1 q_n & \cdots & p_{n-1} q_n & p_n q_n + z_n \end{bmatrix}.$$

We assume that A is nonsingular.

Assuming that the defining vectors p , q , x , y , and z are given, we show how to compute the condition number of A in $O(n)$ operations. This is achieved in a similar way to the tridiagonal case in section 2.1. We will again make use of the special structure of Q and R in the QR factorization of A , solve a scaled linear system, and form absolute column sums using methods that avoid underflow and overflow.

We first note that it is possible to compute $\|A\|_1$ in $O(n)$ operations since the i th absolute column sum can be computed using

$$|y_i|(|x_1| + |x_2| + \cdots + |x_{i-1}|) + |p_i|(|q_{i+1}| + |q_{i+2}| + \cdots + |q_n|) + |p_i q_i + z_i|.$$

By accumulating the sums $|x_1| + |x_2| + \cdots + |x_{i-1}|$ and $|q_{i+1}| + |q_{i+2}| + \cdots + |q_n|$ we can compute the absolute column sums and hence $\|A\|_1$ in $O(n)$ operations. The problem remains to compute $\|A^{-1}\|_1$.

3.1. Computing the 1-norm of the inverse of a dpss matrix. The QR factorization of a dpss matrix can be considered in two stages. First, it is shown in [2] that by applying Givens rotations and scaling we can find an orthogonal matrix $Q_1 \in \mathbb{R}^{n \times n}$ so that $Q_1^T A = H$, where H is upper Hessenberg. We require the condition $q_n \neq 0$; however, we later show that if $q_n = 0$, we need only consider a principal submatrix of A . The special structure of H and Q_1 is described in the following proposition which summarizes the result from [2].

PROPOSITION 3.1. *The upper Hessenberg matrix $H = Q_1^T A$ has an upper triangular part that is equal to the upper triangular part of the rank-2 matrix $ab^T + fg^T \in \mathbb{R}^{n \times n}$ and subdiagonal $h \in \mathbb{R}^{n-1}$, where*

$$(3.2) \quad a = \begin{bmatrix} 1 \\ q_1 \\ q_2 \\ \vdots \\ q_{n-1} \end{bmatrix}, \quad b = \begin{bmatrix} p_1\mu_1 + z_1q_1 \\ (q_1x_1)y_2 + p_2\mu_2 + z_2q_2 \\ (\sum_{i=1}^2 q_ix_i)y_3 + p_3\mu_3 + z_3q_3 \\ \vdots \\ (\sum_{i=1}^{n-1} q_ix_i)y_n + p_n\mu_n + z_nq_n \end{bmatrix},$$

$$(3.3) \quad f = \begin{bmatrix} 0 \\ -\mu_1x_1 \\ -(\sum_{i=1}^1 q_ix_i)q_2 - \mu_2x_2 \\ \vdots \\ -(\sum_{i=1}^{n-2} q_ix_i)q_{n-1} - \mu_{n-1}x_{n-1} \end{bmatrix}, \quad g = y, \quad h = \begin{bmatrix} -z_1\mu_2 \\ -z_2\mu_3 \\ \vdots \\ -z_{n-1}\mu_n \end{bmatrix}$$

with $\mu_i = q_i^2 + \cdots + q_n^2$. The orthogonal matrix Q_1^T is upper Hessenberg with an upper triangular part that is equal to the upper triangular part of the rank-1 matrix $rs^T \in \mathbb{R}^{n \times n}$ and subdiagonal $t \in \mathbb{R}^{n-1}$, where

$$(3.4) \quad r = \begin{bmatrix} 1 \\ q_1 \\ q_2 \\ \vdots \\ q_{n-1} \end{bmatrix}, \quad s = q, \quad t = \begin{bmatrix} -\mu_2 \\ -\mu_3 \\ \vdots \\ -\mu_n \end{bmatrix}.$$

We note that the defining vectors of H and Q_1^T can all be computed in $O(n)$ operations.

The second stage is to reduce H to upper triangular form, which can be achieved by $n-1$ Givens rotations, G_i , defined by (2.3).

The first Givens rotation G_1 is chosen to zero the first element of the subdiagonal of H , and is therefore defined by $\phi_1 = (a_1b_1 + f_1g_1)/\tau_1$ and $\psi_1 = h_1/\tau_1$ with $\tau_1 = \sqrt{(a_1b_1 + f_1g_1)^2 + h_1^2}$. Since G_1 zeros h_1 and the upper triangular part of H is the upper triangular part of $ab^T + fg^T$, we apply G_1 to a and f to give $a^{(1)} = G_1a$ and $f^{(1)} = G_1f$. Except for the first diagonal element, the upper triangular part of G_1H is the upper triangular part of the rank-2 matrix $a^{(1)}b^T + f^{(1)}g^T$. The first diagonal element is given by τ_1 .

This can be repeated to zero the i th subdiagonal element of $G_{i-1} \cdots G_1H$ by

choosing

$$\begin{aligned}\tau_i &= \sqrt{(a_i^{(i-1)}b_i + f_i^{(i-1)}g_i)^2 + h_i^2}, \\ \phi_i &= (a_i^{(i-1)}b_i + f_i^{(i-1)}g_i)/\tau_i, \quad \psi_i = h_i/\tau_i,\end{aligned}$$

and forming $a^{(i)} = G_i a^{(i-1)}$ and $f^{(i)} = G_i f^{(i-1)}$.

After the $n-1$ Givens rotations have been applied,

$$(3.5) \quad Q_2^T Q_1^T A = Q_2^T H = G_{n-1} \dots G_1 H = R,$$

where R is upper triangular. The strictly upper triangular part of R is the strictly upper triangular part of $a^{(n-1)}b^T + f^{(n-1)}g^T$, where

$$a^{(n-1)} = G_{n-1} \dots G_1 a \quad \text{and} \quad f^{(n-1)} = G_{n-1} \dots G_1 f,$$

and the i th diagonal of R is given by τ_i for $i = 1:n-1$ and the n th diagonal element is given by $\tau_n \equiv a_n^{(n-1)}b_n + f_n^{(n-1)}g_n$. Since Q_2^T is a product of $n-1$ Givens rotations applied from top to bottom the structure of Q_2^T is given by (2.5).

From (3.5), we have $A^{-1} = R^{-1}Q_2^T Q_1^T$. In order to compute $\|A^{-1}\|_1$ we first show how to find the lower triangular elements of $R^{-1}Q_2^T$ by using a similar approach to how the lower triangular elements of the inverse of a tridiagonal matrix are found in section 2.1.

The (i, j) element of $R^{-1}Q_2^T$ is

$$(R^{-1}Q_2^T)_{ij} = e_i^T R^{-1}Q_2^T e_j = u_j e_i^T R^{-1}v, \quad i \geq j,$$

where e_i denotes the i th unit vector. If we let w be the solution of the triangular system $Rw = v$, then

$$(R^{-1}Q_2^T)_{ij} = u_j w_i, \quad i \geq j.$$

In [2] it is shown that if the strictly upper triangular part of R is the strictly upper triangular part of a rank-2 matrix, then the triangular system $Rw = v$ can be solved in $O(n)$ operations. However, this method would require products of ψ_i in u and v which may cause underflow and overflow problems. Instead, as in section 2.1 we scale $Rw = v$ using the diagonal matrix D (2.4) in order to avoid products of ψ_i , to give $R'w' = v'$, where

$$R' = D^{-1}RD, \quad w' = D^{-1}w, \quad v' = D^{-1}v = [\phi_1, \dots, \phi_{n-1}, 1]^T.$$

The scaled system can also be solved in $O(n)$ operations due to the structure of D . Given R , D , and v' , the following solves $R'w' = v'$, where $R' = D^{-1}RD$:

$$(3.6) \quad \begin{aligned} 1 & \quad w'_n = v'_n / \tau_n \\ 2 & \quad s' = (-1)^{n-1} \psi_{n-1} w'_n [b_n \ g_n]^T \\ 3 & \quad w'_{n-1} = (v'_{n-1} - (-1)^{n-2} [a_{n-1} \ f_{n-1}] s') / \tau_{n-1} \\ 4 & \quad \text{for } i = n-2:-1:1 \\ 5 & \quad \quad s' = (s' + (-1)^i w'_{i+1} [b_{i+1} \ g_{i+1}]^T) \psi_i \\ 6 & \quad \quad w'_i = (v'_i - (-1)^{i-1} [a_i \ f_i] s') / \tau_i \\ 7 & \quad \text{end} \end{aligned}$$

Let

$$u' = [1, \phi_1, \dots, \phi_{n-1}]^T.$$

Since $w = Dw'$ and $u = D^{-1}u'$, the diagonal elements of $R^{-1}Q_2^T$ are given by

$$(3.7) \quad u_i w_i = u'_i w'_i, \quad i = 1:n.$$

The strictly lower triangular elements are given by

$$(3.8) \quad u_j w_i = (-1)^{i+j-1} \psi_j \dots \psi_{i-1} u'_j w'_i, \quad i > j.$$

The lower triangular part of $R^{-1}Q_2^T$ is therefore defined by the three vectors, u' , w' , and ψ .

The structure of $\text{tril}(R^{-1}Q_2^T, 0)$ and Q_1^T can now be exploited to find the j th absolute column sum of the strictly lower triangular part of $A^{-1} = R^{-1}Q_2^T Q_1^T$. Using (3.7), (3.8), and (3.4), the (i, j) , $i > j$, element of A^{-1} is given by

$$\begin{aligned} (A^{-1})_{ij} &= \sum_{k=1}^j \left((-1)^{k+i-1} \prod_{l=k}^{i-1} \psi_l \right) u'_k w'_i r_k s_j + \left((-1)^{j+i} \prod_{l=j+1}^{i-1} \psi_l \right) u'_{j+1} w'_i t_j \\ &= \left((-1)^{i+j} \prod_{l=j+1}^{i-1} \psi_l \right) w'_i \left(s_j \sum_{k=1}^j \left((-1)^{j+k} \prod_{l=k}^j \psi_l \right) u'_k r_k + u'_{j+1} t_j \right). \end{aligned}$$

The j th absolute column sum of the strictly lower triangular part of A^{-1} is therefore given by

$$\sigma_j = \left| \sum_{i=j+1}^n \left((-1)^{j+i} \prod_{l=j+1}^{i-1} \psi_l \right) w'_i \right| \cdot \left| s_j \sum_{k=1}^j \left((-1)^{j+k} \prod_{l=k}^j \psi_l \right) u'_k r_k + u'_{j+1} t_j \right|.$$

By considering

$$x'_j = \sum_{i=j+1}^n \left((-1)^{j+i} \prod_{l=j+1}^{i-1} \psi_l \right) w'_i$$

and

$$y'_j = s_j \sum_{k=1}^j \left((-1)^{j+k} \prod_{l=k}^j \psi_l \right) u'_k r_k + u'_{j+1} t_j$$

separately, given w' , u' , ψ , r , s , and t , we can compute σ_j for $j = 1:n-1$ in $O(n)$ operations as follows:

$$\begin{aligned} (3.9) \quad & 1 \quad \psi = -\psi \\ & 2 \quad y'_1 = \psi_1 u'_1 r_1 \\ & 3 \quad x'_n = |w'_n| \\ & 4 \quad \text{for } i = 2:n-1 \\ & 5 \quad \quad y'_i = (y'_{i-1} + u'_i r_i) \psi_i \\ & 6 \quad \quad x'_{n-i} = |x'_{n-i+1} \psi_{n-i+1} + w'_{n-i+1}| \\ & 7 \quad \text{end} \\ & 8 \quad \text{for } i = 1:n-1 \\ & 9 \quad \quad \sigma_i = x'_i |y'_i s_i + u'_{i+1} t_i| \\ & 10 \quad \text{end} \end{aligned}$$

In order to find the j th absolute column sum of the upper triangular part of A^{-1} , $\tilde{\sigma}_j$, we can consider $\tilde{A} = JAJ$, where J has only ones on the antidiagonal. By swapping p and q with y and x , respectively, reversing the order of the vectors z , p , q , x , and y , and setting $z_i = x_i y_i + z_i - p_i q_i$ for $i = 1:n$, we can repeat the above process on $\tilde{A}$ to find the absolute columns sums of the strictly lower triangular part of $\tilde{A}^{-1}$ and hence the absolute column sums of the strictly upper triangular part of A^{-1} .

We have shown how to find the strictly lower and strictly upper triangular parts of A^{-1} , and it remains to find the diagonal elements of A^{-1} . Given the strictly lower and strictly upper triangular parts of A^{-1} it is possible to use $AA^{-1} = I$ to find the diagonal elements of A^{-1} in $O(n)$ operations. However, this requires division by the diagonal elements of A , and the diagonal may contain zeros or elements close to zero that cause either breakdown of the algorithm or inaccurate results.

An alternative is to consider $RA^{-1} = Q_2^T Q_1^T$ and make use of

$$R(i, :)A^{-1}(:, i) = (Q_2^T Q_1^T)_{ii}.$$

The diagonal elements of A^{-1} , therefore, satisfy

$$(3.10) \quad (A^{-1})_{ii} = \begin{cases} \frac{(Q_2^T Q_1^T)_{ii} - R(i, i+1:n)A^{-1}(i+1:n, i)}{\tau_i}, & i < n, \\ \frac{(Q_2^T Q_1^T)_{ii}}{\tau_i}, & i = n. \end{cases}$$

It is possible to divide by $\tau_i \equiv R(i, i)$ since $\tau_i \neq 0$ as A is nonsingular. The structure of Q_1^T and Q_2^T is described in Proposition 3.1 and Theorem 2.1, respectively, and it is not difficult to check that, given that we have used the code in (3.6) and (3.9), the following computes $c_i := (Q_2^T Q_1^T)_{ii}$ for $i < n$ in $O(n)$ operations:

$$(3.11) \quad \begin{array}{ll} 1 & s' = w'_n[b_n \ g_n] \\ 2 & c_{n-1} = y'_{n-1}s'[a_{n-1} \ f_{n-1}]^T \\ 3 & \text{for } i = n-2:-1:1 \\ 4 & \quad s' = \psi_{i+1}s' + w'_{i+1}[b_{i+1} \ g_{i+1}] \\ 5 & \quad c_i = y'_i s'[a_i \ f_i]^T \\ 6 & \text{end} \end{array}$$

Similarly, by considering the structure of R and the strictly lower triangular part of A^{-1} , we can compute $\tilde{c}_i := R(i, i+1:n)A^{-1}(i+1:n, i)$ for $i \leq n$ in $O(n)$ operations as follows:

$$(3.12) \quad \begin{array}{ll} 1 & \tilde{c}_1 = u'_1 a_1 \\ 2 & \text{for } i = 2:n \\ 3 & \quad \tilde{c}_i = \tilde{c}_{i-1}\psi_{i-1} + u'_i a_i \\ 4 & \text{end} \\ 5 & \text{for } i = 1:n-1 \\ 6 & \quad \tilde{c}_i = \tilde{c}_i v_i s_i + \psi_i t_i \\ 7 & \text{end} \\ 8 & \tilde{c}_n = \tilde{c}_n v_n s_n \end{array}$$

The absolute column sums and the absolute value of the diagonal elements of A^{-1} can now be combined to give

$$\|A^{-1}\|_1 = \max_{i=1:n} (\sigma_i + \tilde{\sigma}_i + |(A^{-1})_{ii}|).$$

The following algorithm computes $\|A^{-1}\|_1$ in $O(n)$ operations.

ALGORITHM 3.1. *Computes $\xi = \|A^{-1}\|_1$, where A is an $n \times n$ dpss matrix given by (3.1) with $q_n \neq 0$ and $x_1 \neq 0$.*

- 1 Compute vectors in (3.2), (3.3), and (3.4) efficiently
- 2 for $i = 1:n-1$
- 3 $(\phi_i, \psi_i, \tau_i) = \text{Givens}(a_i b_i + f_i g_i, h_i)$ and let G_i be given by (2.3)
- 4 $a = G_i a, f = G_i f$
- 5 $u_{i+1} = \phi_i, v_i = \phi_i$
- 6 end
- 7 $\tau_n = a_n b_n + f_n g_n$
- 8 Solve $R'w' = v'$ using the code in (3.6).
- 9 Compute σ_i for $i < n$ using the code in (3.9).
- 10 Set $\tilde{\sigma} = \sigma$ and repeat above on $\tilde{A}$.
- 11 Compute $(Q_2^T Q_1^T)_{ii}$ using the code in (3.11).
- 12 Compute $R(i, i+1 : n)A^{-1}(i+1 : n, i)$ using the code in (3.12).
- 13 Compute $(A^{-1})_{ii}$ for $i \leq n$ using (3.10).
- 14 $\xi = \max_{i=1:n}(\tilde{\sigma}_i + \sigma_{n-i+1} + |(A^{-1})_{n-i+1, n-i+1}|)$.

Cost. Assuming the function *Givens* requires 6 operations, the total cost is $133n$ operations.

The restrictions $q_n \neq 0$ and $x_1 \neq 0$ in Algorithm 3.1 can easily be removed by considering the structure of A^{-1} . For example, if $q_i, \dots, q_n$ are all zero, then the last $n-i+1$ columns of the strictly lower triangular part of A^{-1} are zero and $(A^{-1})_{jj} = 1/z_i$ for $j = i:n$. Therefore, the absolute column sums of the strictly lower triangular part of A^{-1} can be computed by applying lines 1–9 on $A(1:i-1, 1:i-1)$.

3.2. Numerical experiments. To test Algorithm 3.1 we gave the defining vectors of a dpss matrix random elements in $[-1, 1]$. We tested the resulting dpss matrix and the matrices obtained by raising the components of the defining vectors to the powers 2–6 elementwise. This was repeated 20 times to give 120 test matrices with condition numbers varying from approximately 10^4 to 5×10^{25} . The condition numbers of these test matrices were computed using Algorithm 3.1 and by forming the dpss matrices and using MATLAB's `cond`. The quantity $\beta = (\xi - \kappa_1(A))/\kappa_1(A)^2$ was computed, where ξ denotes the condition number computed using Algorithm 3.1 and $\kappa_1(A)$ denotes the condition number computed using `cond`, and they are shown in Figure 3.1. We divide by $\kappa_1(A)^2$ since the condition number of computing $\kappa_1(A)$ is $\kappa_1(A)$ [11]. Thus we expect $\beta = O(u)$ for a forward stable method, where $u \approx 10^{-16}$ is the unit roundoff. The results show that Algorithm 3.1 performs in a forward stable way.

The test matrices described above are all full matrices. Dpss matrices with various components of the defining vectors set to zero were also tested for which accurate results were obtained.

The condition number of a general real matrix can be estimated in $O(n^2)$ operations using the LAPACK [1] routine `DLACON`. The algorithm estimates $\|B\|_1$ iteratively, for an arbitrary B , by computing the matrix-vector products Bx and $B^T y$ at each iteration for carefully chosen x and y . No more than five iterations are usually required [13, sect. 15.3]. Therefore, if we have a factorization of A such as an LU factorization or a QR factorization, then we can form the matrix-vector products for $B = A^{-1}$ in $O(n^2)$ operations and hence estimate $\|A^{-1}\|_1$ in $O(n^2)$ operations.

In [2], given the defining vectors of a dpss matrix, A , the QR factorization of A is considered to give an algorithm for solving the linear system $Ax = b$ in $O(n)$ operations. By storing the required vectors that define the QR factorizations of A and

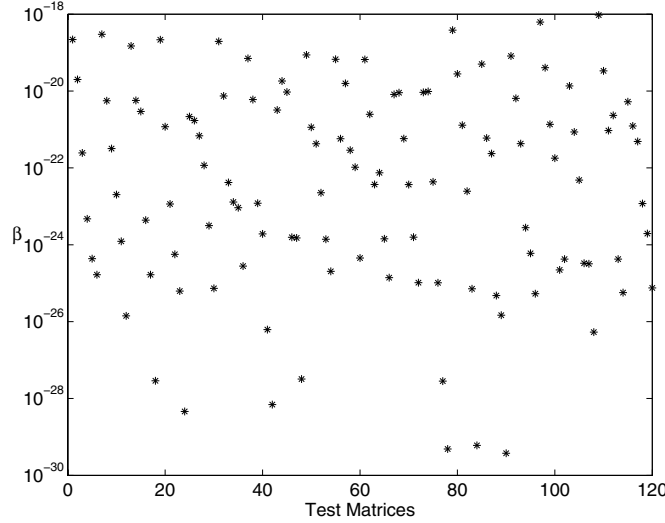


FIG. 3.1. $\beta = (\xi - \kappa_1(A))/\kappa_1(A)^2$ for 120 test matrices $A \in \mathbb{R}^{100 \times 100}$ with varying condition numbers.

TABLE 3.1

Time taken in seconds to compute the 1-norm of the inverse of a random $n \times n$ dpss matrix for various n .

Algorithm	$n = 5 \times 10^5$	$n = 10^6$	$n = 2 \times 10^6$	$n = 5 \times 10^6$
Algorithm 3.1	0.78	1.55	3.04	7.44
LAPACK style	1.15	2.23	4.52	10.83

A^T , the LAPACK algorithm can be adapted to estimate $\|A^{-1}\|_1$ in $O(n)$ operations. We emphasize that this method only *estimates* $\|A^{-1}\|_1$ and that inaccurate estimates can easily occur.

We have found experimentally that estimating the 1-norm of the inverse of a dpss matrix in this way requires two iterations, but this still requires $177n$ operations, which is $44n$ more than that required for Algorithm 3.1. Table 3.1 shows the times taken in seconds to estimate $\|A^{-1}\|_1$ using an adapted version of the LAPACK algorithm for dpss matrices and the times taken to compute $\|A^{-1}\|_1$ using Algorithm 3.1. The algorithms were implemented using Fortran 95 and run on a 2010 MHz AMD Athlon machine. The results show that Algorithm 3.1, which actually computes $\|A^{-1}\|_1$, is quickest.

4. Conclusion. We have presented two algorithms that compute the condition number of a tridiagonal matrix in $O(n)$ operations. The algorithms avoid underflow and overflow and do not require tests for degenerate cases as in [7]. The second algorithm is marginally slower than the quickest algorithm in [7], but is faster than the recommended algorithm in [7]. An $O(n)$ algorithm to compute the condition number of a dpss matrix has also been given. Not only does this compute the condition number exactly, but it is also significantly quicker than a specialized implementation of the LAPACK condition number estimator.

Acknowledgment. I would like to thank Nick Higham for his helpful comments on this work.

REFERENCES

- [1] E. ANDERSON, Z. BAI, C. BISCHOF, S. BLACKFORD, J. DEMMEL, J. DONGARRA, J. D. CROZ, A. GREENBAUM, S. HAMMARLING, A. MCKENNEY, AND D. SORENSEN, *LAPACK Users' Guide*, 3rd ed., SIAM, Philadelphia, PA, 1999.
- [2] M. VAN BAREL, E. VAN CAMP, AND N. MASTRONARDI, *Two fast algorithms for solving diagonal-plus-semiseparable linear systems*, J. Comput. Appl. Math., 164–165 (2004), pp. 731–747.
- [3] M. VAN BAREL, L. G. D. FASINO, AND N. MASTRONARDI, *Orthogonal rational functions and structured matrices*, SIAM J. Matrix Anal. Appl., 26 (2004), pp. 810–829.
- [4] M. VAN BAREL, R. VANDEBRIL, AND N. MASTRONARDI, *An orthogonal similarity reduction of a matrix into semiseparable form*, SIAM J. Matrix Anal. Appl., 27 (2005), pp. 176–197.
- [5] D. A. BINI, L. GEMIGNANI, AND F. TISSEUR, *The Ehrlich–Aberth method for the nonsymmetric tridiagonal eigenvalue problem*, SIAM J. Matrix Anal. Appl., 27 (2005), pp. 153–175.
- [6] S. CHANDRASEKARAN AND M. GU, *Fast and stable algorithms for banded plus semiseparable matrices*, SIAM J. Matrix Anal. Appl., 25 (2003), pp. 373–384.
- [7] I. DHILLON, *Reliable computation of the condition number of a tridiagonal matrix in $O(n)$ time*, SIAM J. Matrix Anal. Appl., 19 (1998), pp. 776–796.
- [8] Y. EIDELMAN AND I. GOHBERG, *A modification of the Dewilde–van der Veen method for inversion of finite structured matrices*, Linear Algebra Appl., 343–344 (2001), pp. 419–450.
- [9] P. E. GILL, G. H. GOLUB, W. MURRAY, AND M. A. SAUNDERS, *Methods for modifying matrix factorizations*, Math. Comp., 28 (1974), pp. 505–535.
- [10] L. GREENGARD AND V. ROKHLIN, *On the solution of two-point boundary value problems*, Comm. Pure Appl. Math., 44 (1991), pp. 419–452.
- [11] D. J. HIGHAM, *Condition numbers and their condition numbers*, Linear Algebra Appl., 214 (1995), pp. 193–213.
- [12] N. J. HIGHAM, *Efficient algorithms for computing the condition number of a tridiagonal matrix*, SIAM J. Sci. Statist. Comput., 7 (1986), pp. 150–165.
- [13] N. J. HIGHAM, *Accuracy and Stability of Numerical Algorithms*, 2nd ed., SIAM, Philadelphia, PA, 2002.
- [14] *IEEE Standard for Binary Floating Point Arithmetic*, Technical Report Std. 754–1985, IEEE/ANSI, New York, 1985.
- [15] S.-Y. KANG, I. KOLTRACHT, AND R. RAWITSCHER, *Nyström–Clenshaw–Curtis quadrature for integral equations with discontinuous kernels*, Math. Comp., 72 (2003), pp. 729–756.
- [16] J. LEE AND L. GREENGARD, *A fast adaptive numerical method for stiff two-point boundary value problems*, SIAM J. Sci. Comput., 18 (1997), pp. 403–429.
- [17] N. MASTRONARDI, S. CHANDRASEKARAN, AND S. VAN HUFFEL, *Fast and stable two-way algorithm for diagonal plus semi-separable systems of linear equations*, Numer. Linear Algebra Appl., 8 (2001), pp. 7–12.
- [18] H. P. STARR, *On the Numerical Solution of One-Dimensional Integral and Differential Equations*, Ph.D. thesis, Yale University, New Haven, CT, 1991.
- [19] G. W. STEWART, *Introduction to Matrix Computations*, Academic Press, New York, 1973.

STRUCTURED FACTORIZATIONS IN SCALAR PRODUCT SPACES*

D. STEVEN MACKEY[†], NILOUFER MACKEY[‡], AND FRANÇOISE TISSEUR[†]

Abstract. Let A belong to an automorphism group, Lie algebra, or Jordan algebra of a scalar product. When A is factored, to what extent do the factors inherit structure from A ? We answer this question for the principal matrix square root, the matrix sign decomposition, and the polar decomposition. For general A , we give a simple derivation and characterization of a particular generalized polar decomposition, and we relate it to other such decompositions in the literature. Finally, we study eigendecompositions and structured singular value decompositions, considering in particular the structure in eigenvalues, eigenvectors, and singular values that persists across a wide range of scalar products.

A key feature of our analysis is the identification of two particular classes of scalar products, termed unitary and orthosymmetric, which serve to unify assumptions for the existence of structured factorizations. A variety of different characterizations of these scalar product classes are given.

Key words. automorphism group, Lie group, Lie algebra, Jordan algebra, bilinear form, sesquilinear form, scalar product, indefinite inner product, orthosymmetric, adjoint, factorization, symplectic, Hamiltonian, pseudo-orthogonal, polar decomposition, matrix sign function, matrix square root, generalized polar decomposition, eigenvalues, eigenvectors, singular values, structure preservation

AMS subject classifications. 15A18, 15A21, 15A23, 15A57, 15A63

DOI. 10.1137/040619363

1. Introduction. The factorization of a general matrix into a product of structured factors plays a key role in theoretical and computational linear algebra. In this work we address the following question: if we apply one of the standard factorizations to a matrix that is already structured, to what extent do the factors have additional structure related to that of the original matrix?

Many applications generate structure, and there are potential benefits to be gained by exploiting it when developing theory and deriving algorithms. For example, algorithms that preserve structure may have reduced storage requirements and operation counts, may be more accurate, and may also provide solutions that are physically more meaningful in the presence of rounding and truncation errors.

The structured matrices we consider belong to the automorphism group $\mathbb{G}$, the Lie algebra $\mathbb{L}$, and the Jordan algebra $\mathbb{J}$ associated with a scalar product, that is, a nondegenerate bilinear or sesquilinear form on $\mathbb{K}^n$. These classes of matrices include linear structures such as complex symmetric, pseudosymmetric, and Hamiltonian matrices, as well as nonlinear structures such as complex orthogonal, pseudo-orthogonal and symplectic matrices. Section 2 introduces concepts and notation needed for our unified treatment of structured factorizations in scalar product spaces. We introduce two important special types of scalar products, termed unitary and orthosymmetric,

*Received by the editors November 22, 2004; accepted for publication (in revised form) by P. Benner June 30, 2005; published electronically January 27, 2006.

<http://www.siam.org/journals/simax/27-3/61936.html>

[†]School of Mathematics, The University of Manchester, Sackville Street, Manchester, M60 1QD, UK (smackey@ma.man.ac.uk, ftisseur@ma.man.ac.uk, <http://www.ma.man.ac.uk/~ftisseur/>). The work of the first and third authors was supported by Engineering and Physical Sciences Research Council grant GR/S31693. The work of the third author was also supported by grant GR/R45079.

[‡]Department of Mathematics, Western Michigan University, Kalamazoo, MI 49008 (nil.mackey@wmich.edu, <http://homepages.wmich.edu/~mackey/>). The work of this author was supported by Engineering and Physical Sciences Research Council Visiting Fellowship GR/S15563.

and describe several equivalent ways to characterize them. The proofs of these equivalences have been delegated to an appendix to avoid disrupting the flow of the paper. These characterizations are essential in clarifying existing results in the literature and in formulating simple, unified assumptions on the scalar product that guarantee the existence of structured factors.

The factorizations we study are the principal square root, the matrix sign decomposition, the polar decomposition, and the generalized polar decomposition (i.e., polar decomposition with respect to a general scalar product). Our results in sections 3 and 4 show that structured principal square roots and structured factors for the matrix sign decomposition exist in arbitrary scalar product spaces. As shown in sections 5 and 6, structured polar factors are guaranteed to exist whenever the scalar product is unitary, while the generalized polar decomposition always exists in an orthosymmetric scalar product space.

Much work has been done on structured spectral decompositions, Schur-like forms, and Jordan-like canonical forms for matrices arising in the context of a specific scalar product or restricted classes of scalar products. An overall unified structured eigendecomposition theory for all scalar products appears to be difficult. However, we show in section 7 that even in an arbitrary scalar product space, the eigenvalues and eigenvectors of matrices in $\mathbb{G}$, $\mathbb{L}$, and $\mathbb{J}$ have a significant level of structure.

Finally, we discuss in section 8 the extent to which the singular values of matrices in $\mathbb{G}$, $\mathbb{L}$, or $\mathbb{J}$ are structured and survey what is known about structured singular value decompositions (SVDs).

2. Scalar products and structured matrices. We begin with the basic definitions and properties of scalar products, and the structured classes of matrices associated with them.

2.1. Scalar products. Consider scalar-valued maps from $\mathbb{K}^n \times \mathbb{K}^n$ to $\mathbb{K}$: $(x, y) \mapsto \langle x, y \rangle$, where $\mathbb{K}$ denotes the field $\mathbb{R}$ or $\mathbb{C}$. When such maps are linear in each argument they are called *bilinear forms* on $\mathbb{K}^n$. For $\mathbb{K} = \mathbb{C}$, maps that are conjugate linear in the first argument and linear in the second are called *sesquilinear forms*.

A real or complex bilinear form has a unique matrix representation given by $\langle x, y \rangle = x^T M y$, while a sesquilinear form can be represented by $\langle x, y \rangle = x^* M y$. We will denote $\langle x, y \rangle$ by $\langle x, y \rangle_M$ as needed. Note that $\langle \cdot, \cdot \rangle_M$ is *nondegenerate* if and only if M is nonsingular. For brevity, the term *scalar product* will be used to refer to any nondegenerate bilinear or sesquilinear form on $\mathbb{K}^n$. The space $\mathbb{K}^n$ equipped with a fixed scalar product is said to be a *scalar product space*.

2.2. Adjoints, automorphisms, and algebras. To each scalar product there corresponds a notion of *adjoint*, generalizing the idea of transpose ^{T} and conjugate transpose * . That is, for any operator A there exists a unique operator $A^\star$, called the *adjoint of A* with respect to the scalar product $\langle \cdot, \cdot \rangle_M$, such that

$$(2.1) \quad \langle Ax, y \rangle_M = \langle x, A^\star y \rangle_M \quad \text{for all } x, y \in \mathbb{K}^n.$$

An explicit formula for the adjoint is given by

$$(2.2) \quad A^\star = \begin{cases} M^{-1} A^T M & \text{for bilinear forms,} \\ M^{-1} A^* M & \text{for sesquilinear forms.} \end{cases}$$

This immediately yields the following useful result. Here $A \sim B$ denotes similarity between matrices A and B .

LEMMA 2.1. *For real or complex bilinear forms, $A^\star \sim A$, while for sesquilinear forms, $A^\star \sim \overline{A}$.*

As one might expect, the adjoint has a number of properties analogous to those of transpose. The following basic properties hold for all scalar products. We omit the simple proofs.

$$(A + B)^\star = A^\star + B^\star, (AB)^\star = B^\star A^\star, (A^{-1})^\star = (A^\star)^{-1},$$

$$(\alpha A)^\star = \alpha A^\star \text{ for bilinear forms, } (\alpha A)^\star = \overline{\alpha} A^\star \text{ for sesquilinear forms.}$$

Notice the absence of the involutory property $(A^\star)^\star = A$ from this list. Only a restricted class of scalar products have an involutory adjoint, as discussed in Appendix A.

Associated to each scalar product are three classes of structured matrices: the automorphisms, self-adjoint matrices, and skew-adjoint matrices. A matrix G is said to be an *automorphism* (or isometry) of the scalar product $\langle \cdot, \cdot \rangle_M$ if $\langle Gx, Gy \rangle_M = \langle x, y \rangle_M$ for all $x, y \in \mathbb{K}^n$, or equivalently, $\langle Gx, y \rangle_M = \langle x, G^{-1}y \rangle_M$ for all $x, y \in \mathbb{K}^n$. In terms of adjoint this corresponds to $G^\star = G^{-1}$. Thus, the automorphism group of $\langle \cdot, \cdot \rangle_M$ is the set

$$\mathbb{G} = \{G \in \mathbb{K}^{n \times n} : G^\star = G^{-1}\}.$$

Matrices S that are *self-adjoint* with respect to the scalar product, i.e., $\langle Sx, y \rangle_M = \langle x, Sy \rangle_M$ for all $x, y \in \mathbb{K}^n$, form a Jordan algebra,

$$\mathbb{J} = \{S \in \mathbb{K}^{n \times n} : S^\star = S\},$$

and matrices K that are *skew-adjoint* with respect to the scalar product, i.e., $\langle Kx, y \rangle_M = -\langle x, Ky \rangle_M$ for all $x, y \in \mathbb{K}^n$, belong to the Lie algebra $\mathbb{L}$, defined by

$$\mathbb{L} = \{K \in \mathbb{K}^{n \times n} : K^\star = -K\}.$$

Some important instances of $\mathbb{G}$, $\mathbb{L}$ and $\mathbb{J}$ are listed in Table 2.1.

$\mathbb{G}$ always forms a multiplicative group (indeed a Lie group), although it is not a linear subspace. By contrast, the sets $\mathbb{L}$ and $\mathbb{J}$ are linear subspaces, but they are not closed under multiplication. Instead $\mathbb{L}$ is closed with respect to the Lie bracket $[K_1, K_2] = K_1 K_2 - K_2 K_1$, while $\mathbb{J}$ is closed with respect to the Jordan product $\{S_1, S_2\} = \frac{1}{2}(S_1 S_2 + S_2 S_1)$. Technical correctness dictates that the symbols $\star$, $\mathbb{G}$, $\mathbb{L}$, and $\mathbb{J}$ all bear the subscript M . For the sake of readability, however, we will usually suppress this subscript.

$\mathbb{G}$ is always closed under products and inverses, being a multiplicative group. The following closure properties of $\mathbb{L}$ and $\mathbb{J}$ will be useful; we omit the simple proofs.

LEMMA 2.2.

- (a) Inverses: *For nonsingular A , $A \in \mathbb{L} \Rightarrow A^{-1} \in \mathbb{L}$ and $A \in \mathbb{J} \Rightarrow A^{-1} \in \mathbb{J}$.*
- (b) Powers: *$A \in \mathbb{J} \Rightarrow A^n \in \mathbb{J}$ for all $n \in \mathbb{N}$. $A \in \mathbb{L} \Rightarrow A^n \in \begin{cases} \mathbb{L} & \text{for } n \text{ odd,} \\ \mathbb{J} & \text{for } n \text{ even.} \end{cases}$*
- (c) Polynomials: *For bilinear forms, $\mathbb{L}$ is closed under all odd polynomials over $\mathbb{K}$, while $\mathbb{J}$ is closed under all polynomials over $\mathbb{K}$. When the scalar product is sesquilinear, the same results hold only for real polynomials.*
- (d) Linear combinations: *For bilinear forms, $\mathbb{L}$ and $\mathbb{J}$ are closed under arbitrary $\mathbb{K}$ -linear combinations and hence are $\mathbb{K}$ -linear subspaces. However, for sesquilinear forms $\mathbb{L}$ and $\mathbb{J}$ are closed only under real linear combinations and hence are only $\mathbb{R}$ -linear subspaces.*

(e) Commuting products: *Suppose A and B commute. Then*

$$\begin{aligned} A, B \in \mathbb{L} &\implies AB \in \mathbb{J}, \\ A, B \in \mathbb{J} &\implies AB \in \mathbb{J}, \\ A \in \mathbb{L}, B \in \mathbb{J} &\implies AB \in \mathbb{L}, BA \in \mathbb{L}. \end{aligned}$$

In general, noncommuting products retain neither $\mathbb{L}$ nor $\mathbb{J}$ structure.

2.3. Unitary and orthosymmetric scalar products. We aim to present results on structured factorizations that use a minimum number of hypotheses on the scalar product defining the structures. As shown in the next sections, some results on structured factorizations need no assumptions on the scalar product at all, while others require at least one of the following properties:

- (a) $(A^*)^\star = (A^\star)^*$ for all $A \in \mathbb{K}^{n \times n}$.
- (b) Adjoint preserves unitarity; that is, U unitary $\Rightarrow U^\star$ is unitary.
- (c) Adjoint preserves Hermitian positive (semi)definite structure; that is, H Hermitian positive (semi)definite $\Rightarrow H^\star$ is Hermitian positive (semi)definite.
- (d) $M = \beta U$ for some unitary U and $\beta > 0$.
- (e) Adjoint with respect to $\langle \cdot, \cdot \rangle_M$ is involutory; that is, $(A^\star)^\star = A$ for all $A \in \mathbb{K}^{n \times n}$.
- (e') $(A^\star)^\star = A$ for all A in some CS-set¹ for $\mathbb{K}^{n \times n}$.
- (f) $\mathbb{K}^{n \times n} = \mathbb{L} \oplus \mathbb{J}$.
- (g) For bilinear forms, $M^T = \pm M$. For sesquilinear forms, $M^* = \alpha M$ with $\alpha \in \mathbb{C}$, $|\alpha| = 1$.
- (h) Vector orthogonality is a symmetric relation; that is,

$$\langle x, y \rangle_M = 0 \iff \langle y, x \rangle_M = 0 \quad \text{for all } x, y \in \mathbb{K}^n.$$

At first glance there is no obvious connection or relationship among these conditions. However, it turns out that they cluster together into two groups of equivalent properties,

$$(a) \Leftrightarrow (b) \Leftrightarrow (c) \Leftrightarrow (d) \quad \text{and} \quad (e) \Leftrightarrow (e') \Leftrightarrow (f) \Leftrightarrow (g) \Leftrightarrow (h).$$

These equivalences are proved in Appendix A, together with the equivalence of some additional properties. We can now identify two natural classes of scalar products.

DEFINITION 2.3. *A scalar product is said to be unitary if it satisfies any one (and hence all) of the properties (a), (b), (c), (d) above.*

DEFINITION 2.4. *A scalar product is said to be orthosymmetric if it satisfies any one (and hence all) of the properties (e), (e'), (f), (g), (h) above.*

The name “orthosymmetric” has been adopted in Definition 2.4 because of property (h). Note that all the “classical” examples of scalar products listed in Table 2.1 are both orthosymmetric and unitary.

We like to abbreviate property (a) by saying, “In unitary scalar products, the stars commute.” This commutativity leads to another useful closure property needed for the structured polar decomposition.

PROPOSITION 2.5. *For any unitary scalar product with automorphism group $\mathbb{G}$, Lie algebra $\mathbb{L}$, and Jordan algebra $\mathbb{J}$, we have $A \in \mathbb{S} \Rightarrow A^\star \in \mathbb{S}$, where $\mathbb{S} = \mathbb{G}$, $\mathbb{L}$ or $\mathbb{J}$.*

Proof. We consider only the case when $\mathbb{S} = \mathbb{G}$, as the proof for $\mathbb{S} = \mathbb{L}$ or $\mathbb{S} = \mathbb{J}$ is similar. Since the stars commute in a unitary scalar product space, $A \in \mathbb{G} \Rightarrow A^\star = A^{-1} \Rightarrow (A^\star)^* = (A^{-1})^* \Rightarrow (A^*)^\star = (A^*)^{-1} \Rightarrow A^* \in \mathbb{G}$. $\square$

¹A CS-set for $\mathbb{K}^{n \times n}$ is any set of matrices $\mathcal{S} \subseteq \mathbb{K}^{n \times n}$ such that $BS = SB$ for all $S \in \mathcal{S}$ implies $B = \alpha I$ for some $\alpha \in \mathbb{K}$. For example, any open subset $\mathcal{S} \subseteq \mathbb{K}^{n \times n}$ is a CS-set (see Lemma A.2(e)).

TABLE 2.1

A sampling of structured matrices associated with scalar products $\langle \cdot, \cdot \rangle_M$, where M is the matrix defining the scalar product.

Space	M	Automorphism group $\mathbb{G} = \{G : G^\star = G^{-1}\}$	Jordan algebra $\mathbb{J} = \{S : S^\star = S\}$	Lie algebra $\mathbb{L} = \{K : K^\star = -K\}$
Bilinear forms				
$\mathbb{R}^n$	I	real orthogonals	symmetrics	skew-symmetrics
$\mathbb{C}^n$	I	complex orthogonals	complex symmetrics	cplx. skew-symmetrics
$\mathbb{R}^n$	$\Sigma_{p,q}$	pseudo-orthogonals ^a	pseudosymmetrics	pseudoskew-symmetrics
$\mathbb{C}^n$	$\Sigma_{p,q}$	cplx. pseudo-orthogonals	cplx. pseudosymm.	cplx. pseudoskew-symm.
$\mathbb{R}^n$	R	real perplectics	persymmetrics	perskew-symmetrics
$\mathbb{R}^{2n}$	J	real symplectics	skew-Hamiltonians	Hamiltonians
$\mathbb{C}^{2n}$	J	complex symplectics	cplx. J -skew-symm.	complex J -symmetrics
Sesquilinear forms				
$\mathbb{C}^n$	I	unitaries	Hermitian	skew-Hermitian
$\mathbb{C}^n$	$\Sigma_{p,q}$	pseudo-unitaries ^b	pseudo-Hermitian	pseudoskew-Hermitian
$\mathbb{C}^{2n}$	J	conjugate symplectics	J -skew-Hermitian	J -Hermitian

Here, $R = \begin{bmatrix} & & 1 \\ & \cdot & \\ 1 & & \end{bmatrix}$, $J = \begin{bmatrix} 0 & I_n \\ -I_n & 0 \end{bmatrix}$, $\Sigma_{p,q} = \begin{bmatrix} I_p & 0 \\ 0 & -I_q \end{bmatrix} \in \mathbb{R}^{n \times n}$.

^aPhysicists refer to the pseudo-orthogonal group with $\Sigma_{3,1} = \begin{bmatrix} I_3 & \\ & -1 \end{bmatrix}$ as the Lorentz group.

^bPseudo-unitary matrices are sometimes called $\Sigma_{p,q}$ -unitaries, or hypernormal matrices in the signal processing literature.

This result allows us to say that in a unitary scalar product space, $\mathbb{G}$, $\mathbb{L}$, and $\mathbb{J}$ are all “closed under the stars.”

3. Structured square roots. A matrix $X \in \mathbb{K}^{n \times n}$ such that $A = X^2$ is said to be a square root of A . Much is known about the existence of matrix square roots and their computation; see [4], [21], [26], for example. In general, the number of square roots may be zero, finite, or infinite. When a square root does exist, the one of most significance for theory and computation is the principal square root, identified by having all its eigenvalues in the open right half-plane and denoted by $A^{1/2}$. It is well known that $A^{1/2}$ exists if and only if A has no eigenvalues on $\mathbb{R}^- := \{x \in \mathbb{R} : x \leq 0\}$, and that $A^{1/2}$ is unique whenever it exists.

If a structured matrix A has a principal square root with the same or related structure, then we can view the equation $A = A^{1/2} \cdot A^{1/2}$ as a structured factorization. Square roots with structure related to the original matrix have been studied in, for example, [17], [24]. In this section we address the question of whether the principal square root of a matrix in $\mathbb{G}$, $\mathbb{L}$, or $\mathbb{J}$ associated with an arbitrary scalar product is also structured. The answer will have a significant impact on many of the other structured factorizations considered in this paper.

LEMMA 3.1. *Suppose A has a principal square root, and let $A^\star$ denote the adjoint of A with respect to an arbitrary scalar product (bilinear or sesquilinear). Then*

- $(A^\star)^{1/2}$ and $(A^{-1})^{1/2}$ both exist,
- $(A^\star)^{1/2} = (A^{1/2})^\star$ and $(A^{-1})^{1/2} = (A^{1/2})^{-1}$.

Proof. (a) A has no eigenvalues on $\mathbb{R}^-$, so neither does $A^\star$, since by Lemma 2.1 $A^\star$ is similar to either A or $\bar{A}$. Now A^{-1} also has no eigenvalues on $\mathbb{R}^-$, since the eigenvalues of A^{-1} are the reciprocals of those of A . Hence $A^{1/2}$, $(A^\star)^{1/2}$, and $(A^{-1})^{1/2}$ all exist.

(b) $(A^{1/2})^\star$ is a square root for $A^\star$, since $(A^{1/2})^\star \cdot (A^{1/2})^\star = (A^{1/2} \cdot A^{1/2})^\star = A^\star$. But $(A^{1/2})^\star$, being similar to either $A^{1/2}$ or $\overline{A^{1/2}}$, has all its eigenvalues in the open right half-plane and hence must be the principal square root of $A^\star$. The proof of the second part of part (b) is similar. $\square$

THEOREM 3.2 (structured principal square roots). *Suppose $\mathbb{G}$, $\mathbb{L}$, and $\mathbb{J}$ are the automorphism group, Lie algebra, and Jordan algebra, respectively, of an arbitrary scalar product and A is a matrix that has a principal square root. Then*

- (a) $A \in \mathbb{G} \implies A^{1/2} \in \mathbb{G}$,
- (b) $A \in \mathbb{L} \implies A^{1/2}$ is not in $\mathbb{L}$,
- (c) $A \in \mathbb{J} \implies A^{1/2} \in \mathbb{J}$.

Proof. Using Lemma 3.1, we have

(a) $A \in \mathbb{G} \implies A^\star = A^{-1} \implies (A^\star)^{1/2} = (A^{-1})^{1/2} \implies (A^{1/2})^\star = (A^{1/2})^{-1} \implies A^{1/2} \in \mathbb{G}$.

(b) Suppose $A^{1/2} \in \mathbb{L}$. Then $A = (A^{1/2})^2 \in \mathbb{J}$, since the square of any matrix in $\mathbb{L}$ is in $\mathbb{J}$. But $\mathbb{L} \cap \mathbb{J} = \{0\}$; thus $A = 0$, contradicting the existence of $A^{1/2}$.

(c) $A \in \mathbb{J} \implies A^\star = A \implies (A^\star)^{1/2} = A^{1/2} \implies (A^{1/2})^\star = A^{1/2} \implies A^{1/2} \in \mathbb{J}$. $\square$

In section 5.2 we need to consider square roots of structured matrices that are *singular* and hence do not have a principal square root. These singular matrices, though, are Hermitian positive semidefinite and thus have a distinguished square root—the unique Hermitian positive semidefinite square root [25]. Note that whenever a matrix A has both a principal square root and a Hermitian positive semidefinite square root, these two roots coincide, and so without ambiguity we can use $A^{1/2}$ to denote this unique distinguished square root. In the following analogues of Lemma 3.1 and Theorem 3.2, we consider only unitary scalar products.

LEMMA 3.3. *Suppose A is Hermitian positive semidefinite, and let $A^\star$ denote the adjoint of A with respect to a unitary scalar product. Then*

- (a) $(A^\star)^{1/2}$ exists, and $(A^\star)^{1/2} = (A^{1/2})^\star$.
- (b) If A is nonsingular, then $(A^{-1})^{1/2}$ exists and $(A^{-1})^{1/2} = (A^{1/2})^{-1}$.

Proof. (a) A and $A^{1/2}$ are both Hermitian positive semidefinite; hence by Definition 2.3(c) so are both $A^\star$ and $(A^{1/2})^\star$. Thus $(A^\star)^{1/2}$ exists. Since $(A^{1/2})^\star \cdot (A^{1/2})^\star = (A^{1/2} \cdot A^{1/2})^\star = A^\star$, $(A^{1/2})^\star$ must be the unique Hermitian positive semidefinite square root of $A^\star$, i.e., $(A^\star)^{1/2} = (A^{1/2})^\star$.

(b) If A is nonsingular, then the Hermitian positive semidefinite square root is the same as the principal square root; thus Lemma 3.1 applies. $\square$

THEOREM 3.4 (structured Hermitian positive semidefinite square roots). *Suppose $\mathbb{G}$, $\mathbb{L}$, and $\mathbb{J}$ are the automorphism group, Lie algebra, and Jordan algebra, respectively, of a unitary scalar product, and A is Hermitian positive semidefinite. Then*

- (a) $A \in \mathbb{G} \implies A^{1/2} \in \mathbb{G}$,
- (b) $A \in \mathbb{L} \implies A^{1/2} \in \mathbb{L}$ only if $A = 0$,
- (c) $A \in \mathbb{J} \implies A^{1/2} \in \mathbb{J}$.

Proof. Using Lemma 3.3 in place of Lemma 3.1, the proofs are all formally identical to the corresponding ones in Theorem 3.2. The only difference is that $(\cdot)^{1/2}$ denotes the Hermitian positive semidefinite square root. $\square$

4. Structured matrix sign decomposition. The matrix sign function, defined for matrices $A \in \mathbb{C}^{n \times n}$ with no pure imaginary eigenvalues, was originally

introduced and developed by Roberts [47] as a tool to solve algebraic Riccati equations. In [20], Higham extended this notion to a factorization $A = SN$, where S is the matrix sign of A and $N := S^{-1}A$, and showed how this “matrix sign decomposition” is closely related to the polar decomposition. The next result shows that for a matrix A structured with respect to an arbitrary scalar product, the sign factors S and N are also structured.

THEOREM 4.1. *Let $\mathbb{S}$ be the automorphism group $\mathbb{G}$, the Lie algebra $\mathbb{L}$, or the Jordan algebra $\mathbb{J}$ of an arbitrary scalar product, and suppose $A \in \mathbb{K}^{n \times n}$ has a sign decomposition $A = SN$. Then*

$$(4.1) \quad A \in \mathbb{S} \implies S \in \mathbb{S} \text{ and } N \in \begin{cases} \mathbb{S} & \text{when } \mathbb{S} = \mathbb{G}, \mathbb{J}, \\ \mathbb{J} & \text{when } \mathbb{S} = \mathbb{L}. \end{cases}$$

Proof. In each case we focus first on the structure of the factor N and then from this deduce the structure of S , using the characterizations $N = (A^2)^{1/2}$ and $S = A(A^2)^{-1/2}$ from [20]. Note that A has a matrix sign if and only if A has no eigenvalues on the imaginary axis, so that A^2 then has no zero or negative eigenvalues, and hence $(A^2)^{1/2}$ exists.

First suppose that $A \in \mathbb{G}$. Then $A^2 \in \mathbb{G}$, so $N = (A^2)^{1/2} \in \mathbb{G}$ by Theorem 3.2. Since $\mathbb{G}$ is a group, it follows that $S = AN^{-1}$ is also in $\mathbb{G}$.

Next let $A \in \mathbb{L}$. Then $A^2 \in \mathbb{J}$ by Lemma 2.2(b); hence $N = (A^2)^{1/2} \in \mathbb{J}$ by Theorem 3.2, and thus $N^{-1} \in \mathbb{J}$ by Lemma 2.2(a). Finally, since A and N^{-1} commute, we have $S = AN^{-1} \in \mathbb{L}$ by Lemma 2.2(e).

Finally, consider $A \in \mathbb{J}$. Then $A^2 \in \mathbb{J}$, and thus $N = (A^2)^{1/2} \in \mathbb{J}$ and $N^{-1} \in \mathbb{J}$ just as above. With A and N^{-1} being commuting elements of $\mathbb{J}$, we have $S = AN^{-1} \in \mathbb{J}$ by Lemma 2.2(e). $\square$

Theorem 4.1 can also be proved using globally convergent structure-preserving iterations for the matrix sign. Such iterations $X_{k+1} = f(X_k)$ converge to $\text{sign}(X_0)$ for any $X_0 \in \mathbb{S}$ for which $\text{sign}(X_0)$ is defined, and have $X_k \in \mathbb{S}$ for all k . A family of such iterations is discussed in [23], but only in the context of preserving automorphism group structure. Two particular examples are the cubically converging $X_{k+1} = f(X_k) = X_k(3I + X_k^2)(I + 3X_k^2)^{-1}$ and the quintically converging $Y_{k+1} = g(Y_k) = Y_k(5I + 10Y_k^2 + Y_k^4)(I + 10Y_k^2 + 5Y_k^4)^{-1}$. That the matrix functions f and g preserve automorphism group structure is shown in [23] and [24], but by using Lemma 2.2 they can also be shown to preserve $\mathbb{L}$ and $\mathbb{J}$ structure. The well-known Newton iteration $Z_{k+1} = \frac{1}{2}(Z_k + Z_k^{-1})$ for matrix sign also preserves $\mathbb{L}$ and $\mathbb{J}$. The existence of these structure-preserving iterations gives perhaps a more direct insight into why the sign S of a matrix A should have the same structure as A . The fact that A , S , and N all commute then implies that $N \in \mathbb{J}$ whenever $A \in \mathbb{L}$ or $A \in \mathbb{J}$.

5. Structured polar decomposition. The polar decomposition of a matrix $A \in \mathbb{C}^{n \times n}$ factors A as a product $A = UH$, where U is unitary and H is Hermitian positive semidefinite. The Hermitian factor is always unique, whereas the unitary factor is unique only if A is nonsingular [25]. When A is real, so are U and H . We begin by considering the polar factors of matrices belonging to automorphism groups, Lie algebras, and Jordan algebras associated with unitary scalar products, and investigate the extent to which these factors have similar structure. In section 5.2 the question of the structure of polar factors is addressed for automorphism groups of more general scalar products.

5.1. Unitary scalar products. Several conditions equivalent to a scalar product being unitary were given in section 2.3. Two of them are key to establishing the

structure of the polar factors of matrices in $\mathbb{G}$, $\mathbb{L}$, and $\mathbb{J}$: in a unitary scalar product space, adjoint preserves both unitarity and Hermitian positive definite structure (Definition 2.3, properties (b) and (c)).

5.1.1. Automorphism groups.

THEOREM 5.1. *Suppose $\mathbb{G}$ is the automorphism group of a unitary scalar product, and $A \in \mathbb{G}$. Then in the polar decomposition $A = UH$ the factors U and H also belong to $\mathbb{G}$.*

Proof. The matrix A is nonsingular since $A \in \mathbb{G}$; thus U and H exist and are unique. We have

$$A \in \mathbb{G} \Leftrightarrow A = A^{-\star} = (UH)^{-\star} = U^{-\star}H^{-\star}.$$

By Definition 2.3, properties (b) and (c), $U^{\star}$ is unitary and $H^{\star}$ is Hermitian positive definite. Thus $A = U^{-\star}H^{-\star}$ exhibits another polar decomposition of A . Uniqueness of polar factors implies that $U = U^{-\star}$ and $H = H^{-\star}$; i.e., U and H are in $\mathbb{G}$. $\square$

Just as for the sign function, an alternate proof of this result can be given based on the existence of globally convergent structure-preserving iterations for the unitary polar factor of matrices in automorphism groups. Details about such iterations and their theoretical and numerical properties may be found in [23]. Connections between these polar factor iterations and other matrix iterations are explored and developed further in [24].

Note that Theorem 5.1 is known for some particular automorphism groups; for real symplectic matrices see [45], and for real pseudo-orthogonal matrices see [50].

5.1.2. Lie and Jordan algebras. Next we consider the polar factors of matrices in Lie and Jordan algebras associated with unitary scalar products. For a matrix A in $\mathbb{L}$ or $\mathbb{J}$, when can we expect the polar factors of A to be in $\mathbb{L}$ or $\mathbb{J}$? An added feature of this question that was not present for matrices in automorphism groups is that elements of $\mathbb{L}$ and $\mathbb{J}$ may be singular and thus have nonunique unitary polar factors. We look first at the unitary factor in the nonsingular case.

THEOREM 5.2. *Suppose A is nonsingular with polar decomposition $A = UH$. Let $\mathbb{S}$ denote either the Lie algebra $\mathbb{L}$ or the Jordan algebra $\mathbb{J}$ associated with a unitary scalar product. Then $A \in \mathbb{S} \Rightarrow U \in \mathbb{S}$.*

Proof. We prove only the case when $\mathbb{S} = \mathbb{L}$; the argument when $\mathbb{S} = \mathbb{J}$ is almost identical. Since $A \in \mathbb{L}$, $A^{\star} = -A$ and

$$A = UH \Rightarrow A^{\star} = H^{\star}U^{\star} \Rightarrow -A = H^{\star}U^{\star} \Rightarrow A = H^{\star}(-U^{\star}) = \tilde{H}\tilde{U}.$$

This displays a “backwards” polar decomposition for A , since $H^{\star}$ is Hermitian positive definite and $U^{\star}$ is unitary. But for nonsingular matrices, the “forward” and “backward” unitary factors are equal [25]. Hence $U = \tilde{U} = -U^{\star}$; i.e., $U \in \mathbb{L}$. $\square$

An alternative proof for Theorem 5.2 can be given using the Newton iteration $X_{k+1} = \frac{1}{2}(X_k + X_k^{-\star})$ for computing the unitary polar factor of the matrix X_0 . For unitary scalar products it can be shown that this iteration is structure-preserving for all nonsingular matrices in $\mathbb{L}$ or $\mathbb{J}$. The global convergence of the iteration on nonsingular matrices then implies the desired result. More on Newton-like iterations can be found in [23] and [24].

To handle the singular case we need the following lemma.

LEMMA 5.3. *Suppose $\mathcal{W} \subseteq \mathbb{K}^{n \times n}$ is a (real or complex) subspace of matrices. Let $\text{Inv}(\mathcal{W})$ denote the subset of invertible matrices in $\mathcal{W}$. Then either $\text{Inv}(\mathcal{W})$ is empty,*

or it is an open dense subset of $\mathcal{W}$ whose complement $\mathcal{W} \setminus \text{Inv}(\mathcal{W})$ is a set of measure zero in $\mathcal{W}$.

Proof. Let $\{W_1, W_2, \dots, W_\ell\}$ be a basis for $\mathcal{W}$, so that every $X \in \mathcal{W}$ is uniquely expressed as $X = a_1 W_1 + \dots + a_\ell W_\ell$. Then $p(a_1, a_2, \dots, a_\ell) = \det(a_1 W_1 + \dots + a_\ell W_\ell)$ is a polynomial in the unknowns $a_1, \dots, a_\ell$ that distinguishes the singular elements from the nonsingular elements of $\mathcal{W}$; the zero set of p in $\mathbb{K}^\ell$ can be identified with the set of singular matrices in $\mathcal{W}$. This zero set is not empty because $p(0, 0, \dots, 0) = 0$; thus it is either all of $\mathcal{W}$ (if and only if $p \equiv 0$), or it is a nontrivial algebraic variety in $\mathcal{W}$. But nontrivial varieties are always closed and nowhere dense (in the usual Euclidean topology) and are of measure zero in $\mathcal{W}$. $\square$

With this lemma we can now deal with singular matrices in $\mathbb{L}$ or $\mathbb{J}$. It is possible that $\mathbb{L}$ or $\mathbb{J}$ may contain *only* singular matrices, as the lemma indicates (consider, for example, the Lie algebra of $n \times n$ skew-symmetric matrices for n odd). In such a case, $\mathbb{L}$ (or $\mathbb{J}$) would contain no unitary matrices at all; this would be an insurmountable obstruction to the existence of a structured polar decomposition. Thus we must assume that $\mathbb{L}$ (or $\mathbb{J}$) contains at least one nonsingular matrix.

THEOREM 5.4. *Let $\mathbb{S}$ denote either the Lie algebra $\mathbb{L}$ or the Jordan algebra $\mathbb{J}$ of a unitary scalar product, and suppose $A \in \mathbb{S}$ is a singular matrix. If $\mathbb{S}$ contains at least one nonsingular matrix, then there exists a polar decomposition $A = UH$ such that $U \in \mathbb{S}$; however, even this polar decomposition with $U \in \mathbb{S}$ need not be unique.*

Proof. We consider only $A \in \mathbb{L}$, using a limit argument based on the already established result for the nonsingular case. The proof when $A \in \mathbb{J}$ is similar.

By assumption, $\mathbb{L}$ contains at least one nonsingular matrix; thus by Lemma 5.3 the nonsingular matrices are dense in $\mathbb{L}$. Let $A_n \rightarrow A$ with $A_n \in \mathbb{L}$ and A_n nonsingular. By Theorem 5.2 each A_n has a polar decomposition $A_n = U_n H_n$ with $U_n \in \mathbb{L}$. Since the unitary group is compact, we may pick a subsequence U_{n_k} that converges to a unitary U ; because $\mathbb{L}$ is closed, this U is also in $\mathbb{L}$. Then the sequence $H_{n_k} = U_{n_k}^{-1} A_{n_k}$ converges, say to H . The set of Hermitian positive semidefinite matrices is also a closed set, and thus H is Hermitian positive semidefinite. Then

$$A = \lim A_n = \lim A_{n_k} = \lim U_{n_k} \cdot \lim H_{n_k} = UH$$

displays a polar decomposition for A with $U \in \mathbb{L}$. The nonuniqueness assertion is concretely demonstrated by Example 5.5. $\square$

Example 5.5. Consider the matrix

$$A = \begin{bmatrix} 0 & -2 \\ 2 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix}$$

in the Lie algebra $\mathbb{L}$ of 4×4 real skew-symmetric matrices. For $A \in \mathbb{L}$ we have the polar decompositions

$$\begin{aligned} A &= U_1 H = \begin{bmatrix} 0 & -1 \\ 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} 2 & 0 \\ 0 & 2 \\ 0 & 0 \\ 0 & 0 \end{bmatrix} \text{ with } U_1 \notin \mathbb{L}, \\ A &= U_2 H = \begin{bmatrix} 0 & -1 \\ 1 & 0 & 0 & 0 \\ 0 & 0 & -1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} 2 & 0 \\ 0 & 2 \\ 0 & 0 \\ 0 & 0 \end{bmatrix} \text{ with } U_2 \in \mathbb{L}, \\ \text{and } A &= U_3 H = \begin{bmatrix} 0 & -1 \\ 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ -1 & 0 & 0 & 0 \end{bmatrix} \begin{bmatrix} 2 & 0 \\ 0 & 2 \\ 0 & 0 \\ 0 & 0 \end{bmatrix} \text{ with } U_3 \in \mathbb{L}, \end{aligned}$$

illustrating the fact that a singular $A \in \mathbb{L}$ may have polar decompositions with the unitary factor in $\mathbb{L}$ or not in $\mathbb{L}$, as well as having *more* than one polar decomposi-

tion with the unitary factor in $\mathbb{L}$. Similar examples may be constructed for singular matrices in the Jordan algebra of 4×4 real symmetric matrices.

What about the structure of the Hermitian factor H when A is in $\mathbb{L}$ or $\mathbb{J}$? Though H is always unique, regardless of whether A is singular or nonsingular, in general H does not seem to have any additional structure beyond Hermitian positive semidefiniteness. However, there is one special situation where H does have extra structure.

THEOREM 5.6. *Suppose A is in $\mathbb{L}$ or $\mathbb{J}$ of a unitary scalar product, and H is the Hermitian positive semidefinite polar factor of A . Then $H \in \mathbb{J}$ if and only if A is normal, i.e., $A^*A = AA^*$.*

Proof. We consider only the case $A \in \mathbb{L}$, since the proof for $A \in \mathbb{J}$ is almost identical.

($\Rightarrow$): By Lemma 2.2(b), $H = (A^*A)^{1/2} \in \mathbb{J} \Rightarrow A^*A \in \mathbb{J}$; i.e., $(A^*A)^* = A^*A$. So then $A^*A = (A^*A)^* = A^*(A^*)^* = A^*(A^*)^*$, since the stars commute in any unitary scalar product space (see Definition 2.3(a)). But $A \in \mathbb{L}$ means $A^* = -A$, and therefore $A^*A = A^*(A^*)^* = (-A)(-A)^* = AA^*$. Thus A is normal.

($\Leftarrow$): By Proposition 2.5, $A \in \mathbb{L} \Rightarrow A^* \in \mathbb{L}$. Then since A is normal, we have $A^*A \in \mathbb{J}$ by Lemma 2.2(e). By Theorem 3.4 we then have $H = (A^*A)^{1/2} \in \mathbb{J}$. $\square$

5.2. General scalar products. We now turn to the question of the polar factors of matrices in the automorphism group of a *general* scalar product. The results of section 3 on structured principal square roots allow us to give conditions under which an *individual* $A \in \mathbb{G}$ has both of its polar factors in $\mathbb{G}$, without making any global assumption about the scalar product or the automorphism group.

THEOREM 5.7. *Let $\mathbb{G}$ be the automorphism group of an arbitrary scalar product, and suppose $A \in \mathbb{G}$. Then the following are equivalent:*

- (a) $(A^*)^* = (A^*)^*$ (i.e., “the stars commute on A ”).
- (b) $A^* \in \mathbb{G}$.
- (c) A has a structured polar decomposition; i.e., $A = UH$ with $U, H \in \mathbb{G}$.

Proof. (a $\Rightarrow$ b): $A^* = A^{-1} \Rightarrow (A^*)^* = (A^{-1})^* \Rightarrow (A^*)^* = (A^*)^{-1} \Rightarrow A^* \in \mathbb{G}$.

(b $\Rightarrow$ c): Suppose $A = UH$ is the polar decomposition of A . Since $A^* \in \mathbb{G}$ we have $A^*A = H^2 \in \mathbb{G}$, and thus $H = (H^2)^{1/2} \in \mathbb{G}$ by Theorem 3.2. But then $U = AH^{-1} \in \mathbb{G}$ because $\mathbb{G}$ is a group, and thus A has a structured polar decomposition.

(c $\Rightarrow$ a): On one side, $(A^*)^* = ((UH)^*)^* = (HU^{-1})^* = U^{-*}H^* = UH^{-1}$, while on the other side, $(A^*)^* = ((UH)^*)^* = (H^{-1}U^{-1})^* = U^{-*}H^{-*} = UH^{-1}$. $\square$

In what follows, for any scalar product, we say that the automorphism group $\mathbb{G}$ “has structured polar decompositions” if for *every* $A \in \mathbb{G}$ we have $U, H \in \mathbb{G}$, where $A = UH$ is the polar decomposition of A .

COROLLARY 5.8. *An automorphism group $\mathbb{G}$ has structured polar decompositions if and only if $\mathbb{G}$ is closed under conjugate transpose.*

The next results identify two classes of scalar products whose automorphism groups have structured polar decompositions. Characterizing the set of *all* scalar products whose automorphism groups have structured polar decompositions is still an open question.

COROLLARY 5.9. *The automorphism group of any unitary scalar product has structured polar decompositions.*

Proof. This has already been proved in Theorem 5.1. But it is also a consequence of Corollary 5.8, together with the fact that $\mathbb{G}$ is closed under conjugate transpose when the scalar product is unitary (Proposition 2.5). $\square$

Corollary 5.10 extends a result of Horn, Merino, and Serre [28], who considered scalar products $\langle \cdot, \cdot \rangle_M$ defined by real M with $M^2 = \pm I$. In this corollary, note that

M may be complex in the sesquilinear case.

COROLLARY 5.10. *Let $\langle \cdot, \cdot \rangle_M$ be a scalar product on $\mathbb{K}^n$ such that $M^2 = \alpha I$ for some nonzero $\alpha \in \mathbb{K}$. In the complex bilinear case, further assume that M is real. Then the associated automorphism group $\mathbb{G}$ has structured polar decompositions.*

Proof. From (2.2) it follows that

$$(5.1) \quad (A^*)^\star = \begin{cases} M^{-1}\overline{A}M, & \text{bilinear forms,} \\ M^{-1}AM, & \text{sesquilinear forms.} \end{cases}$$

Observe that $M^2 = \alpha I$ implies

$$(5.2) \quad M^{-1}AM = MAM^{-1} \quad \text{for all } A \in \mathbb{K}^{n \times n}.$$

Our strategy will be to show that $\mathbb{G}$ is closed under conjugate transpose.

Complex sesquilinear forms: $A \in \mathbb{G} \Rightarrow A^\star A = I \Rightarrow M^{-1}A^\star M A = I$. Thus $M^{-1}A^\star$ and MA are inverses and hence commute, giving us $MAM^{-1}A^\star = I$. Using (5.2), this becomes $M^{-1}AM A^\star = I$, or $(A^\star)^\star A^\star = I$ by (5.1). Hence $A^\star \in \mathbb{G}$.

Real or complex bilinear forms with M real: Proceeding along similar lines we obtain $A \in \mathbb{G} \Rightarrow M^{-1}AM A^T = I$. Taking conjugates and using the fact that M is real gives $M^{-1}\overline{A}M A^\star = I$, or $(A^\star)^\star A^\star = I$, and thus $A^\star \in \mathbb{G}$. $\square$

The example below illustrates that for complex bilinear forms $\langle \cdot, \cdot \rangle_M$, the hypothesis $M^2 = \alpha I$ by itself is not sufficient to guarantee that $\mathbb{G}$ has structured polar decompositions. Hence the necessity for some additional hypothesis like M is real in Corollary 5.10.

Example 5.11. Consider the bilinear form on $\mathbb{C}^2$ defined by $M = \begin{bmatrix} 1 & i \\ 0 & -1 \end{bmatrix}$. Note that $M = \begin{bmatrix} 1 & \beta \\ 0 & -1 \end{bmatrix}$ satisfies $M^2 = I$ for any $\beta \in \mathbb{C}$. A direct calculation shows that $A \in \mathbb{G}$ if and only if A is of the form $\begin{bmatrix} a & b \\ b & a+ib \end{bmatrix}$ with $a, b \in \mathbb{C}$ and $\det A = +1$. Hence $\mathbb{G}$ is far from being closed under conjugate transpose. Indeed, both A and $A^\star$ belong to $\mathbb{G}$ only when $A = \pm I$; therefore these are the *only* elements of $\mathbb{G}$ with structured polar decompositions.

6. Generalized polar decomposition. In this section we look at a factorization that is somewhat different in nature from all the others considered in this paper. In other sections we take as starting point a matrix A in either $\mathbb{G}$, $\mathbb{L}$, or $\mathbb{J}$ of a scalar product space and then study the extent to which some standard factorization produces factors of A that are also in $\mathbb{G}$, $\mathbb{L}$, or $\mathbb{J}$. Here we begin instead with a *general* matrix $A \in \mathbb{K}^{n \times n}$ and seek to decompose it as $A = WS$, where W is in the automorphism group and S is in the Jordan algebra of some scalar product on $\mathbb{K}^n$. This notion generalizes the standard polar decomposition, where the scalar product is the standard inner product $\langle x, y \rangle = x^*y$ on $\mathbb{C}^n$, the automorphism group is the set of unitary matrices, and the Jordan algebra is the set of Hermitian matrices. Since there are scalar products for which the spectra of matrices in $\mathbb{J}$ can be arbitrary finite subsets of $\mathbb{C}$ (e.g., $\langle x, y \rangle = x^T y$ on $\mathbb{C}^n$, with $\mathbb{J}$ the set of complex symmetric matrices), it is not immediately clear what might constitute an appropriate replacement for the positive semidefiniteness condition on the Hermitian factor in the usual polar decomposition. We will return to this issue after briefly surveying some of the literature on this factorization.

The idea of generalizing the polar decomposition in this way is not new; it has been studied independently by several groups of authors using various notation and terminology, and in various contexts. An extensive body of work on this topic is due to Bolshakov, Reichstein, and others, starting with [5], [6], and [7], and continuing

through the recent [33] and [42]. In these papers the basic setting is a scalar product space defined by a Hermitian sesquilinear form on $\mathbb{C}^n$, termed an “indefinite inner (or scalar) product space.” Questions of existence of factorizations $A = WS$ ($W \in \mathbb{G}$, $S \in \mathbb{J}$) are considered with S satisfying various types of extra conditions, and even no extra condition at all, with technical emphasis on the more difficult case when A is singular. Kintzel [33] studies the doubly structured existence question: Given two scalar products with associated groups $\mathbb{G}_1, \mathbb{G}_2$ and Jordan algebras $\mathbb{J}_1, \mathbb{J}_2$, when can a matrix A be factored as $A = WS$ with $W \in \mathbb{G}_1 \cap \mathbb{G}_2$ and $S \in \mathbb{J}_1 \cap \mathbb{J}_2$? Another recurring theme in these papers is the characterization of special classes of matrices (e.g., matrices that are normal with respect to the given scalar product) for which every A in the class can be factored as $A = WS$. Complementary to this work are papers of Kaplansky [31] and Ikramov [29]; Kaplansky considers symmetric bilinear forms on $\mathbb{K}^n$ for algebraically closed fields $\mathbb{K}$, while Ikramov treats only the symplectic bilinear form on $\mathbb{C}^{2n}$.

In [27], Horn and Merino investigate a matrix factorization which they term the Φ_S -polar decomposition. Here the setting is *not* a scalar product space but rather the complex matrices $\mathbb{C}^{n \times n}$ equipped with a linear operator $\Phi_S(A) := SA^T S^{-1}$ for some fixed nonsingular symmetric or skew-symmetric $S \in \mathbb{C}^{n \times n}$. Matrices A such that $\Phi_S(A) = A$ are called Φ_S -symmetric, and matrices A such that $\Phi_S(A) = A^{-1}$ are Φ_S -orthogonal. Then A is said to have a Φ_S -polar decomposition if $A = XY$ for some Φ_S -orthogonal X and Φ_S -symmetric Y . This way of generalizing the usual polar decomposition (on $\mathbb{R}^{n \times n}$) is easily seen to be a special case of the generalized polar decomposition in scalar product spaces described above; if $M = S^{-1}$ is used to define a bilinear form $\langle x, y \rangle_M = x^T M y$ on $\mathbb{C}^n$, then $\Phi_S(A)$ is the same as the adjoint $A^\star$ with respect to this scalar product, and $\mathbb{G} = \{\Phi_S\text{-orthogonals}\}$ while $\mathbb{J} = \{\Phi_S\text{-symmetrics}\}$. Horn and Merino show that for any such Φ_S , every nonsingular A has a Φ_S -polar decomposition, and also give necessary and sufficient conditions for a general A to have a Φ_S -polar decomposition.

Yet a different approach to this factorization uses methods from Lie theory. Here the setting is a Lie group (or Lie semigroup) G equipped with an involutive homomorphism $\sigma : G \rightarrow G$; i.e., $\sigma^2 = \text{id}$ but $\sigma \neq \text{id}$. A generalized polar decomposition (or Ol’shanskii decomposition) of an element $z \in G$ in this setting is a factorization $z = xy$, where x is an element of the fixed point set $G^\sigma = \{x \in G : \sigma(x) = x\}$ and y is an element of the anti-fixed point set $G_\sigma = \{y \in G : \sigma(y) = y^{-1}\}$. See [35] for a development of the basic theory of this decomposition. A fundamental example is the group $G = GL(n, \mathbb{R})$ of all nonsingular real $n \times n$ matrices with the map $\sigma(A) = A^{-T}$. In this example G^σ is the group of real orthogonal matrices and G_σ is the set of all nonsingular symmetric matrices, and thus we are back to the usual notion of polar decomposition. Indeed, for any orthosymmetric scalar product on $\mathbb{K}^n$, the map $\sigma(A) = A^{-\star}$ is an involutive homomorphism on the group $G = GL(n, \mathbb{K})$, with $G^\sigma = \mathbb{G}$ and G_σ the set of all nonsingular elements of $\mathbb{J}$.

A disadvantage of this Lie group setting is that singular matrices are not considered at all; however, there are compensating advantages. The most important advantage is that many more examples than just the scalar product space notion of generalized polar decomposition can be encompassed in a unified development, including both matrix groups equipped with involutions σ that do not arise from any scalar product structure, as well as nonmatrix Lie groups associated with the study of differential equations. For more on such examples and a further development of the theory, see [46]. This Lie group notion of generalized polar decomposition has also recently been exploited in [51] and [30] to develop new structure-preserving algorithms

for computing the matrix exponential.

We return now to consider the question of an appropriate generalization for the positive definiteness condition on the self-adjoint (Hermitian) factor in the usual polar decomposition. Several such replacements have been suggested in the literature, but to us the simplest approach is to restrict the eigenvalues of the self-adjoint factor $S \in \mathbb{J}$ to some region of the complex plane containing the positive real axis, so as to compatibly generalize the usual positive definiteness condition. The most natural such region for the eigenvalues of S is the open right half-plane; this eigenvalue condition is concisely expressed as $\text{sign}(S) = I$. Bhatia [3] and Kintzel [33] use this condition: Bhatia only for the bilinear form $\langle x, y \rangle = x^T y$ on $\mathbb{C}^n$, and Kintzel for Hermitian sesquilinear and real symmetric bilinear forms.

Further justifying the choice of this particular condition is that it gives a generalized polar decomposition (GPD) that is strongly connected with the matrix sign function and the principal square root, both of which are well known to be computable by matrix iterations (see [21] and [32]). This connection is elaborated on and expanded in [24] to show how large families of matrix iterations for the matrix sign function and the principal square root are related to each other and to iterations for the GPD with the $\text{sign}(S) = I$ condition (see also [33] for one particular iteration from this family). One may reasonably view this GPD as the “computable” generalized polar decomposition, and thus we adopt the following definition, both for this paper and for [24], where iterations to compute this GPD are derived and analyzed.

DEFINITION 6.1 (“computable” GPD). *Suppose $\mathbb{K}^n$ is a scalar product space with automorphism group $\mathbb{G}$ and Jordan algebra $\mathbb{J}$. Then for a matrix $A \in \mathbb{K}^{n \times n}$, a generalized polar decomposition (GPD) with respect to this scalar product is a factorization $A = WS$, where $W \in \mathbb{G}$, $S \in \mathbb{J}$, and $\text{sign}(S) = I$.*

Although the condition $\text{sign}(S) = I$ immediately restricts us to considering only nonsingular A , it also leads to a connection between this notion of a GPD and the principal square root, as will be seen in the proof of Theorem 6.2. This theorem deals with the existence and uniqueness of a (“computable”) GPD for an individual matrix, with respect to an arbitrary scalar product.

THEOREM 6.2. *Let $\mathbb{K}^n$ be an arbitrary scalar product space. Then a matrix $A \in \mathbb{K}^{n \times n}$ has a GPD with respect to this scalar product if and only if $(A^\star)^\star = A$ and $A^\star A$ has no eigenvalues on $\mathbb{R}^- := \{x \in \mathbb{R} : x \leq 0\}$. When such a factorization exists it is unique.*

Proof. ($\Rightarrow$) Note first that if the factorization exists, then

$$(A^\star)^\star = (S^\star W^\star)^\star = (SW^{-1})^\star = W^{-\star} S^\star = (W^{-1})^{-1} S = WS = A.$$

Also we must have

$$(6.1) \quad A^\star A = S^\star W^\star WS = S^\star S = S^2.$$

But if $\text{sign}(S) = I$ is to hold, then the only possible choice for S is the principal square root $S = (A^\star A)^{1/2}$, and this square root exists only if $A^\star A$ has no eigenvalues on $\mathbb{R}^-$.

($\Leftarrow$) By hypothesis, $A^\star A$ has a principal square root. Letting $S = (A^\star A)^{1/2}$, the condition $\text{sign}(S) = I$ is automatically satisfied. It remains to show that $S^\star = S$. Using Lemma 3.1(b) together with the assumption that $(A^\star)^\star = A$, we have

$$S^\star = ((A^\star A)^{1/2})^\star = ((A^\star A)^\star)^{1/2} = (A^\star A)^{1/2} = S.$$

Finally, since $S^2 = A^\star A$, the uniquely defined matrix $W = AS^{-1}$ satisfies

$$W^\star W = (AS^{-1})^\star (AS^{-1}) = S^{-\star} (A^\star A) S^{-1} = S^{-1} (S^2) S^{-1} = I,$$

and thus $W \in \mathbb{G}$. $\square$

A matrix factorization is most useful in practice if it exists for a large set of matrices, certainly for at least an *open* subset of $\mathbb{K}^{n \times n}$. With this in mind, we examine the two conditions in Theorem 6.2 to see which scalar products are capable of supporting a generalized polar decomposition (with or without the condition $\text{sign}(S) = I$) that exists on some open subset of matrices.

For any scalar product the matrix $A = I$ produces an A^*A that has no eigenvalues on $\mathbb{R}^-$. The continuity of the eigenvalues of A^*A under perturbation of A then implies that there will always be some (perhaps very large) open neighborhood of the identity in $\mathbb{K}^{n \times n}$ where the eigenvalue condition $\sigma(A^*A) \cap \mathbb{R}^- = \emptyset$ holds for all A in the neighborhood. Thus the eigenvalue condition on A^*A in Theorem 6.2 puts no constraint on which scalar products can support a useful generalized polar decomposition.

Turning next to the involutory property $(A^*)^* = A$, we will see that the situation is somewhat different. For a general scalar product we can always expect there to be a large set of matrices for which $(A^*)^* = A$ holds, as the next result shows, but this set need not always contain an open set.

LEMMA 6.3. *For a scalar product space $\mathbb{K}^n$, $(A^*)^* = A$ holds for all A in the $\mathbb{K}$ -subalgebra of $n \times n$ matrices generated by $\mathbb{G} \cup \mathbb{L} \cup \mathbb{J}$.*

Proof. The straightforward proof is omitted. $\square$

However, from property (e') in Definition 2.4 we know that the involutory property of adjoint holds in some open set of matrices if and only if the scalar product is orthosymmetric. Thus we are led to the following theorem.

THEOREM 6.4. *Let $\langle \cdot, \cdot \rangle$ be a scalar product on $\mathbb{K}^n$. Then there exists some open² subset $\mathcal{U} \subseteq \mathbb{K}^{n \times n}$ of matrices such that every $A \in \mathcal{U}$ can be factored as $A = WS$ with $W \in \mathbb{G}$ and $S \in \mathbb{J}$ if and only if the scalar product $\langle \cdot, \cdot \rangle$ is orthosymmetric. (Note that we make no assumption here on S other than $S \in \mathbb{J}$, nor is the factorization $A = WS$ assumed to be unique. See Theorem A.5 for a further strengthening of this result.)*

Proof. ($\Rightarrow$) For all $A \in \mathcal{U}$ we have

$$(A^*)^* = (S^*W^*)^* = (SW^{-1})^* = W^{-*}S^* = (W^{-1})^{-1}S = WS = A.$$

Thus by Lemma A.2(e) and Theorem A.5(a') we know that $\langle \cdot, \cdot \rangle$ must be orthosymmetric.

($\Leftarrow$) From Theorem A.5(a), Theorem 6.2, and the earlier discussion of the spectrum of A^*A for A near I , we see that there is some open neighborhood of I in which every matrix A has a GPD. $\square$

Theorem 6.4 provides a compelling reason why one might choose to consider only orthosymmetric scalar products when studying generalized polar decompositions of any kind, even those without the condition $\text{sign}(S) = I$, or those that lack uniqueness of the factorization. Furthermore, this result also clarifies the *mathematical necessity* of the choices of scalar product made in previous work, which might otherwise appear to be unnecessarily restrictive or motivated only by a particular application. For orthosymmetric scalar products, then, we have the following simplification of Theorem 6.2.

COROLLARY 6.5. *For $\mathbb{K}^n$ equipped with an orthosymmetric scalar product, a matrix $A \in \mathbb{K}^{n \times n}$ has a GPD if and only if A^*A has no eigenvalues on $\mathbb{R}^-$. Whenever this GPD exists it is unique.*

²This is with respect to the standard Euclidean topology on $\mathbb{K}^{n \times n}$.

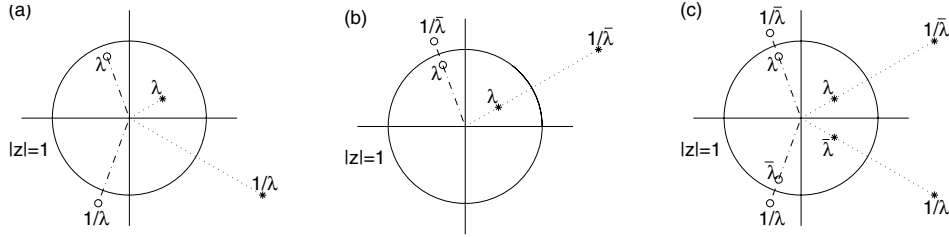


FIG. 7.1. Eigenvalue pairings relative to the unit circle in (a) general bilinear, (b) sesquilinear, and (c) real bilinear cases.

7. Eigenvalue and eigenvector structure. Even when no restrictions are imposed on the scalar product, matrices in the corresponding automorphism group and Lie and Jordan algebras have structure in their eigenvalues and eigenvectors. Some of this structure has been observed and exploited before, but usually in specific cases. In the context of a fixed scalar product or a restricted class of scalar products, the pursuit of spectral decompositions, Jordan canonical forms, and other condensed forms under structure-preserving similarities or general similarities has been the focus of intense study. The literature on this subject is extensive; see, for example, [1], [2], [12], [15], [18], [36], [38], [40], [41], [44], [43], and the references therein.

In this section we give a simple, unified presentation of the common eigenstructure shared by matrices in $\mathbb{G}$, $\mathbb{L}$, or $\mathbb{J}$ associated with an *arbitrary* scalar product. Our aim is to shed light on the extent to which the structure of eigenvalues and eigenvectors is independent of the particular scalar product defining the classes $\mathbb{G}$, $\mathbb{L}$, and $\mathbb{J}$.

7.1. Eigenvalues of automorphisms. The determinant and eigenvalues that are possible for matrices in automorphism groups are restricted.

PROPOSITION 7.1. *Let $A \in \mathbb{G}$. Then $\det A = \pm 1$ for any bilinear form, while for sesquilinear forms $|\det A| = 1$.*

Proof. For bilinear forms, $A \in \mathbb{G} \Rightarrow A^*A = I \Rightarrow M^{-1}A^TMA = I \Rightarrow (\det A)^2 = 1 \Rightarrow \det A = \pm 1$, while for sesquilinear forms, $A^*A = I \Rightarrow M^{-1}A^*MA = I \Rightarrow \det \bar{A} \det A = 1 \Rightarrow |\det A| = 1$. $\square$

In some automorphism groups, the determinant can be even more restricted; for example, real and complex symplectic matrices have only $+1$ determinant, and -1 is never realized. For a collection of proofs of this result, see [37].

Next we turn to the pairing structure found in the eigenvalues of matrices in $\mathbb{G}$.

THEOREM 7.2. *Let $A \in \mathbb{G}$. The eigenvalues of A come in pairs λ and $1/\lambda$ for bilinear forms and in pairs λ and $1/\bar{\lambda}$ for sesquilinear forms. In both cases these pairs have the same Jordan structure and hence the same algebraic and geometric multiplicities.*

Proof. From Lemma 2.1 we have $A^{-1} = A^* \sim A$ for bilinear forms, whereas $A^{-1} = A^* \sim \bar{A}$ for sesquilinear forms. The result is now immediate. $\square$

These eigenvalue pairings are illustrated in Figure 7.1(a) and (b). Instances of Theorem 7.2 for specific automorphism groups can be found, for example, in [8] and [22].

The automorphism group of a real bilinear form can be viewed as a restriction to real matrices of either the automorphism group of a complex bilinear form or the automorphism group of a complex sesquilinear form. Hence both eigenvalue structure results apply to real automorphisms, yielding the following corollary.

COROLLARY 7.3. *Let $A \in \mathbb{G}$, where $\mathbb{G}$ is the automorphism group of a real bilinear form. Then the eigenvalues of A come in quartets $\lambda, 1/\lambda, \bar{\lambda}, 1/\bar{\lambda}$. Figure 7.1(c) depicts such quartets of eigenvalues relative to the unit circle in $\mathbb{C}$.*

7.2. Characteristic polynomials of automorphisms. We next describe a nice property of the characteristic polynomials of matrices in automorphism groups. For a polynomial $p(x) = a_n x^n + a_{n-1} x^{n-1} + \cdots + a_1 x + a_0$ of degree n (i.e., $a_n \neq 0$), we define the “reversal of $p(x)$ ” to be the polynomial $\text{rev} p(x) = a_0 x^n + a_1 x^{n-1} + \cdots + a_{n-1} x + a_n$, that is, the polynomial p with its coefficients in reverse order. It is easy to see that

$$\text{rev} p(x) = x^n p(1/x).$$

The “conjugate of $p(x)$ ” is the polynomial $\bar{p}(x) = \bar{a}_n x^n + \bar{a}_{n-1} x^{n-1} + \cdots + \bar{a}_1 x + \bar{a}_0$.

DEFINITION 7.4. *We say that $p(x)$ is*

- palindromic if $\text{rev} p(x) = p(x)$,
- anti-palindromic if $\text{rev} p(x) = -p(x)$,
- conjugate palindromic if $\text{rev} p(x) = \bar{p}(x)$,
- conjugate anti-palindromic if $\text{rev} p(x) = -\bar{p}(x)$.

THEOREM 7.5. *Suppose $A \in \mathbb{G}$. Let $p_A(x) = \det(xI - A)$ denote the characteristic polynomial of A , and let $s := (-1)^n \det(A)$.*

- (a) *For bilinear forms, $s = \pm 1$; $p_A(x)$ is palindromic if $s = 1$ and anti-palindromic if $s = -1$.*
- (b) *For sesquilinear forms, $|s| = 1$, and $q_A(x) := \overline{s^{1/2}} p_A(x)$ is conjugate palindromic. Thus $p_A(x)$ is a scalar multiple of a conjugate palindromic polynomial.*

Proof. These results can be derived as consequences of the eigenvalue structure proved in Theorem 7.2, but we include a more direct proof here.

(a) As shown in the proof of Theorem 7.2, we have $A \sim A^{-1}$. From Proposition 7.1 we know that $\det A = \pm 1$. Thus,

$$\begin{aligned} p_A(x) &= p_{A^{-1}}(x) = (\det A)^2 \det(xI - A^{-1}) \\ &= (\det A) \det(xA - I) = (\det A) \det((-xI)(x^{-1}I - A)) \\ &= (\det A)(-x)^n p_A(x^{-1}) = (-1)^n (\det A) \text{rev} p_A(x) \\ &= (\pm 1) \text{rev} p_A(x). \end{aligned}$$

Hence $p_A(x)$ is either palindromic or anti-palindromic, depending on the sign of $s = (-1)^n \det(A)$.

(b) In this case we have $A \sim \bar{A}^{-1}$ and $(\det A) \overline{(\det \bar{A})} = 1$. Since $\det A$ can now have complex values other than ± 1 , it will not be as straightforward to see palindromicity here. We have

$$\begin{aligned} p_A(x) &= p_{\bar{A}^{-1}}(x) = (\det A) \overline{(\det \bar{A})} \det(xI - \bar{A}^{-1}) \\ &= (\det A) \det(x\bar{A} - I) = (\det A) \det((-xI)(x^{-1}I - \bar{A})) \\ (7.1) \quad &= (\det A)(-x)^n p_{\bar{A}}(x^{-1}) = (-1)^n (\det A) \text{rev} \bar{p}_A(x). \end{aligned}$$

So if $\det A = \pm 1$, then $p_A(x)$ is already conjugate palindromic or conjugate anti-palindromic. But in general $s := (-1)^n \det A = e^{i\theta}$, and $p_A(x)$ itself will not usually be conjugate (anti)palindromic. However, it is always possible to “symmetrize” $p_A(x)$ by

an appropriate complex scalar multiple that makes the resulting polynomial conjugate palindromic. Define $q_A(x) = \overline{s^{1/2}} p_A(x) = e^{-i\theta/2} p_A(x)$. Then using (7.1) we get

$$q_A(x) = e^{-i\theta/2} e^{i\theta} \operatorname{rev} \overline{p}_A(x) = e^{i\theta/2} \operatorname{rev} \overline{p}_A(x) = \operatorname{rev} \overline{q}_A(x).$$

Thus $q_A(x)$ is conjugate palindromic. $\square$

7.3. Eigenvalues of matrices in Lie and Jordan algebras. Analogous to the situation for automorphism groups, the relationship between matrices in $\mathbb{L}$ or $\mathbb{J}$ and their adjoints also leads to a pairing of eigenvalues and their corresponding Jordan blocks, as the next result shows.

THEOREM 7.6. *Let $A \in \mathbb{L}$ or $A \in \mathbb{J}$. Then the eigenvalues of A occur in pairs as shown below, with the same Jordan structure for each eigenvalue in a pair.*

	Bilinear	Sesquilinear
$A \in \mathbb{L}$	$\lambda, -\lambda$	$\lambda, -\bar{\lambda}$
$A \in \mathbb{J}$	“no pairing”	$\lambda, \bar{\lambda}$

Proof. When $A \in \mathbb{L}$, then $A^\star = -A$ by definition. Lemma 2.1 now implies $A \sim -A$ in the bilinear case and $A \sim -\bar{A}$ in the sesquilinear case, thus forcing eigenvalue pairings $\lambda, -\lambda$ and $\lambda, -\bar{\lambda}$, respectively, as well as the identical Jordan block structure for each eigenvalue in a pair.

On the other hand, when $A \in \mathbb{J}$, $A^\star = A$. In the bilinear case Lemma 2.1 puts no constraints on the eigenvalues. However, in the sesquilinear case we get $A \sim \bar{A}$, forcing the pairing $\lambda, \bar{\lambda}$ as eigenvalues of A with the same Jordan structure. $\square$

It is well known that any finite subset of $\mathbb{C}$ can be realized as the spectrum of a complex symmetric matrix [25, Thm. 4.4.9]. Thus there can be no eigenvalue structure property that holds in general for Jordan algebras of all bilinear forms; this explains the “no pairing” entry in the table in Theorem 7.6. On the other hand, for certain special classes of $\mathbb{L}$ or $\mathbb{J}$ there may be additional structure in the eigenvalues for reasons other than those considered in Theorem 7.6. For example, it is known that the eigenvalues of any real or complex skew-Hamiltonian matrix all have even multiplicity [16], [17], [29]. This is a special case of the following result.

PROPOSITION 7.7. *Let $\mathbb{J}$ be the Jordan algebra of any skew-symmetric bilinear form on $\mathbb{K}^n$. Then for any $A \in \mathbb{J}$, the eigenvalues of A all have even multiplicity. Moreover, for any $m > 0$ and eigenvalue λ , the number of $m \times m$ Jordan blocks corresponding to λ in the Jordan form for A is even.*

Proof. Suppose M is the $n \times n$ matrix defining the given skew-symmetric bilinear form; note that n is even since M is nonsingular. Then $A \in \mathbb{J} \Rightarrow A^\star = A \Rightarrow M^{-1} A^T M = A \Rightarrow A^T M = M A \Rightarrow (M A)^T = -M A$. Thus $A = M^{-1} \cdot (M A)$ expresses A as the product of two skew-symmetric matrices, the first of which is nonsingular. But any such product has an even number of $m \times m$ Jordan blocks for any eigenvalue λ , by Propositions 3 and 4 in [29]. The even multiplicity of all eigenvalues of A now follows immediately. $\square$

7.4. Eigenvector structure. The eigenvectors of matrices in $\mathbb{G}$, $\mathbb{L}$, or $\mathbb{J}$ cannot in general be arbitrarily chosen from $\mathbb{K}^n$, nor can they be arbitrarily matched up with scalars in $\mathbb{K}$ to form eigenpairs. In this section we will see what these restrictions on eigenvectors are and how they can sometimes feed back to put extra constraints on the spectra of structured matrices.

To concisely describe the relationships among the eigenvectors and their corresponding eigenvalues, we introduce various eigenvalue “pairing functions” $\wp$ to sum-

marize the eigenvalue structure results of Theorems 7.2 and 7.6. The function $\wp$ depends on the type of structured matrix under consideration, and associates to a scalar λ the eigenvalue “partner” appropriate to that structure.

Scalar product	$\mathbb{G}$	$\mathbb{L}$	$\mathbb{J}$
Bilinear	$\wp(\lambda) := 1/\lambda$	$\wp(\lambda) := -\lambda$	“no pairing”
Sesquilinear	$\wp(\lambda) := 1/\bar{\lambda}$	$\wp(\lambda) := -\bar{\lambda}$	$\wp(\lambda) := \bar{\lambda}$

We will also need to recall the notion of isotropic vector. A nonzero vector v in a scalar product space $\mathbb{K}^n$ is said to be *isotropic* if $\langle v, v \rangle_{\mathbb{M}} = 0$, and *nonisotropic* if $\langle v, v \rangle_{\mathbb{M}} \neq 0$. Note that the set of *all* isotropic vectors in any scalar product space can be only one of the following three types: empty, all of $\mathbb{K}^n \setminus \{0\}$, or a nontrivial algebraic variety in $\mathbb{K}^n \setminus \{0\}$ (and hence a set of measure zero in $\mathbb{K}^n$). Then we have the following result.

THEOREM 7.8. *Suppose $A \in \mathbb{S}$, where $\mathbb{S}$ is any one of $\mathbb{G}$, $\mathbb{L}$, or $\mathbb{J}$ for a scalar product space $\mathbb{K}^n$. Let $x, y \in \mathbb{K}^n$ be eigenvectors for A with $Ax = \lambda x$ and $Ay = \alpha y$. Then*

- (a) $\lambda \neq \wp(\alpha) \Rightarrow \langle x, y \rangle_{\mathbb{M}} = 0$ (x and y are “ M -orthogonal”),
- (b) $\lambda \neq \wp(\lambda) \Rightarrow x$ is an isotropic vector.

Proof. We give the proof only for $\mathbb{G}$ belonging to a sesquilinear form. The arguments for the other four cases are similar. We have

$$\langle \lambda x, y \rangle_{\mathbb{M}} = \langle Ax, y \rangle_{\mathbb{M}} = \langle x, A^{\star} y \rangle_{\mathbb{M}} = \langle x, A^{-1} y \rangle_{\mathbb{M}} = \langle x, (1/\alpha) y \rangle_{\mathbb{M}}.$$

Thus $(\bar{\lambda} - \frac{1}{\alpha}) \langle x, y \rangle_{\mathbb{M}} = 0$ or, equivalently, $(\lambda - \wp(\alpha)) \overline{\langle x, y \rangle_{\mathbb{M}}} = 0$ from the table above. Part (a) now follows immediately, and letting $x = y$ gives part (b). $\square$

Thus eigenvectors of structured matrices can be nonisotropic only if their eigenvalues are on the fixed point set of the corresponding pairing function. For sesquilinear forms these fixed point sets are the unit circle for $\mathbb{G}$, the imaginary axis for $\mathbb{L}$, and the real axis for $\mathbb{J}$. When a scalar product space contains *no* isotropic vectors at all, eigenvalues of structured matrices are all *forced* to be on these fixed point sets, constraining the eigenvalues even more than is generally the case. The simplest and best known example of this is the standard inner product $\langle x, y \rangle = x^* y$ on $\mathbb{C}^n$, where $\mathbb{G}$ is the unitary group with all eigenvalues on the unit circle, $\mathbb{L}$ is the skew-Hermitian matrices with all imaginary eigenvalues, and $\mathbb{J}$ is the Hermitian matrices with all real eigenvalues.

8. Structured singular value decomposition. The SVD is one of the most important factorizations in all of linear algebra. The SVD factors a matrix $A \in \mathbb{K}^{n \times n}$ into $A = U \Sigma V^*$, where $U, \Sigma, V \in \mathbb{K}^{n \times n}$, U and V are unitary, and Σ is a diagonal matrix with only nonnegative entries [19], [25]. The factors U , Σ , and V are never unique, although uniqueness is often forced on Σ by adopting the convention that its diagonal entries are in nonincreasing order. For the structured matrices considered in this paper it is sometimes useful to relax this ordering convention or replace it by a different ordering convention.

In this section we will first discuss the extent to which the singular values of matrices in $\mathbb{G}$, $\mathbb{L}$, and $\mathbb{J}$ are structured and then survey what is known about structured SVDs.

8.1. Singular values of automorphisms. The singular values of matrices in many (but not all) automorphism groups have a pairing structure analogous to the eigenvalue pairings described in Theorem 7.2.

THEOREM 8.1. *Suppose $\mathbb{G}$ is the automorphism group of a scalar product and $A \in \mathbb{G}$ has a structured polar decomposition. Then A has reciprocally paired singular values; i.e., the singular values of A come in pairs, σ and $1/\sigma$, with the same multiplicity.*

Proof. The singular values of A are the same as the eigenvalues of the Hermitian factor H in the structured polar decomposition $A = UH$. Since $H \in \mathbb{G}$, by Theorem 7.2 the eigenvalues of H , which are real and positive, have the reciprocal pairing property. $\square$

The converse of Theorem 8.1 does not hold. There are examples of automorphism groups $\mathbb{G}$ for which every $A \in \mathbb{G}$ has reciprocally paired singular values, but not every $A \in \mathbb{G}$ has a structured polar decomposition.

Example 8.2. Consider the automorphism group $\mathbb{G}$ of the bilinear form on $\mathbb{R}^2$ defined by $M = \begin{bmatrix} 1 & 0 \\ 0 & 4 \end{bmatrix}$. Since $\det A = \pm 1$ for every $A \in \mathbb{G}$ by Proposition 7.1, this forces every $A \in \mathbb{G}$ to have reciprocally paired singular values. However, one can easily verify that $A = \begin{bmatrix} 0 & 2 \\ \frac{1}{2} & 0 \end{bmatrix}$ is in $\mathbb{G}$ while A^* is not, thus showing (by Theorem 5.7) that A does *not* have a structured polar decomposition. More generally, for any scalar product on $\mathbb{R}^n$ defined by $M = \begin{bmatrix} I_{n-1} & 0 \\ 0 & k \end{bmatrix}$ with $k > 0$, it can be shown that every $A \in \mathbb{G}$ has reciprocally paired singular values, but when $k \neq 1$ almost every $A \in \mathbb{G}$ *fails* to have a structured polar decomposition.

As an immediate consequence of Theorem 8.1 together with the results of section 5, we have the following large classes of automorphism groups in which every matrix has reciprocally paired singular values.

COROLLARY 8.3. *Let $\mathbb{G}$ be the automorphism group of a unitary scalar product, or of a scalar product $\langle \cdot, \cdot \rangle_M$ with $M^2 = \alpha I$ such that M is real when $\langle \cdot, \cdot \rangle_M$ is complex bilinear. Then every $A \in \mathbb{G}$ has reciprocally paired singular values.*

Note that this result has previously been known for some particular groups, using different arguments in each case. For real pseudo-orthogonal matrices, see [22]; for symplectic matrices, see [14], [39], [49].

For scalar products that are neither unitary nor have $M^2 = \alpha I$, the conclusion of Corollary 8.3 may or may not hold. For example, MATLAB experiments show that reciprocal pairing of singular values is absent when $M = \text{diag}(1, 2, 3)$. With $M = \text{diag}(1, 1, 2)$, though, numerical results indicate that singular value pairing is present—indeed, this pairing is a special case of Example 8.2. Characterizing the set of *all* scalar products whose automorphism groups have reciprocally paired singular values remains an open question.

Although it is not yet known precisely which automorphism groups have structured polar factors, or which automorphism groups have reciprocally paired singular values, Theorem 8.1 provides a necessary condition for the existence of a structured polar decomposition: Any automorphism that does not have the singular value reciprocal pairing property cannot have structured polar factors.

8.2. Singular values of matrices in $\mathbb{L}$ or $\mathbb{J}$. In general there is much less structure in the singular values of matrices in $\mathbb{L}$ or $\mathbb{J}$ than there is for automorphisms; real symmetric matrices and real Hamiltonian matrices are examples of Jordan and Lie algebras whose elements can have arbitrary singular values. On the other hand, there are also Lie and Jordan algebras in which all nonzero singular values of all matrices have even multiplicity.³ Both of these phenomena are explained by the

³Note that all of the examples of $\mathbb{L}$ and $\mathbb{J}$ in Table 2.1 exhibit one of these two behaviors, either no singular value structure at all or even multiplicity of all (nonzero) singular values.

following result.

THEOREM 8.4. *Let $\mathbb{S}$ be either the Lie algebra $\mathbb{L}$ or the Jordan algebra $\mathbb{J}$ of a scalar product $\langle \cdot, \cdot \rangle_{\mathbb{M}}$ on $\mathbb{K}^n$ that is both orthosymmetric and unitary. Suppose $A \in \mathbb{S}$, so that $A^* = \kappa A$ where $\kappa = \pm 1$.*

- (a) *For bilinear forms, $M^T = \varepsilon M$ where $\varepsilon = \pm 1$. If $\varepsilon \neq \kappa$, then the (nonzero) singular values of A all have even multiplicity, while if $\varepsilon = \kappa$, then the singular values of matrices in $\mathbb{S}$ are unstructured; i.e., for an arbitrary list of n nonnegative real numbers (repetitions allowed) there is some $A \in \mathbb{S}$ with these singular values.*
- (b) *For sesquilinear forms, the singular values of matrices in either $\mathbb{L}$ or $\mathbb{J}$ are unstructured, in the same sense used in part (a).*

Proof. (a) We begin by demonstrating the result for $M = I$, i.e., for

$$(8.1) \quad \text{Skew}(\mathbb{K}) := \{A \in \mathbb{K}^{n \times n} : A^T = -A\} \quad \text{and} \quad \text{Sym}(\mathbb{K}) = \{A \in \mathbb{K}^{n \times n} : A^T = A\}.$$

For skew-symmetric A , $A^*A = -\bar{A}A$ is the product of two skew-symmetric matrices, and thus all its nonzero eigenvalues have even multiplicity [13]. Hence the nonzero singular values of A all have even multiplicity. By contrast, the set $\text{Sym}(\mathbb{K})$ contains the real symmetric positive semidefinite matrices, which clearly may have arbitrary singular values.

Next we invoke the orthosymmetry of $\langle \cdot, \cdot \rangle_{\mathbb{M}}$ to see that for any $A \in \mathbb{S}$, MA is either symmetric or skew-symmetric. First, recall from property (g) in Definition 2.4 that for any bilinear orthosymmetric scalar product $\langle \cdot, \cdot \rangle_{\mathbb{M}}$, $M^T = \varepsilon M$ where $\varepsilon = \pm 1$. Then $A \in \mathbb{S} \Rightarrow \kappa A = A^* = M^{-1}A^TM \Rightarrow MA = \kappa A^TM$, and thus $(MA)^T = \kappa(A^TM)^T = \varepsilon\kappa(MA)$. By Definition 2.4 (f) we know that $\mathbb{K}^{n \times n} = \mathbb{L} \oplus \mathbb{J}$, and hence left multiplication by M is a bijection from $\mathbb{K}^{n \times n}$ to $\mathbb{K}^{n \times n}$ that maps $\mathbb{L}$ and $\mathbb{J}$ to $\text{Skew}(\mathbb{K})$ and $\text{Sym}(\mathbb{K})$. To be more precise, we have shown that

$$M \cdot \mathbb{S} = \begin{cases} \text{Skew}(\mathbb{K}) & \text{if } \varepsilon \neq \kappa, \\ \text{Sym}(\mathbb{K}) & \text{if } \varepsilon = \kappa \end{cases}$$

or, equivalently,

$$(8.2) \quad \mathbb{S} = \begin{cases} M^{-1} \cdot \text{Skew}(\mathbb{K}) & \text{if } \varepsilon \neq \kappa, \\ M^{-1} \cdot \text{Sym}(\mathbb{K}) & \text{if } \varepsilon = \kappa. \end{cases}$$

Finally, $\langle \cdot, \cdot \rangle_{\mathbb{M}}$ being a unitary scalar product means that $M^{-1} = \alpha U$ for some unitary U and $\alpha > 0$, by Definition 2.3, property (d). From (8.2) we can therefore conclude that the singular value structure of $\mathbb{S}$ is the same as that of $\text{Skew}(\mathbb{K})$ or $\text{Sym}(\mathbb{K})$, depending only on whether ε and κ are equal or not.

(b) For sesquilinear forms, orthosymmetry of $\langle \cdot, \cdot \rangle_{\mathbb{M}}$ means that $M^* = \gamma M$ for some $|\gamma| = 1$ (see Definition 2.4(g)). Choosing β so that $\beta^2 = \gamma$, an argument similar to the one in part (a) shows that for any $A \in \mathbb{S}$, βMA is either Hermitian or skew-Hermitian. But the set of all skew-Hermitian matrices is $i \cdot \text{Herm}(\mathbb{C})$, where $\text{Herm}(\mathbb{C})$ denotes the set of Hermitian matrices. Combining this with $\mathbb{C}^{n \times n} = \mathbb{L} \oplus \mathbb{J}$, we have the following analogue of (8.2):

$$(8.3) \quad \mathbb{S} = \begin{cases} i\bar{\beta}M^{-1} \cdot \text{Herm}(\mathbb{C}) & \text{if } \kappa = -1, \\ \bar{\beta}M^{-1} \cdot \text{Herm}(\mathbb{C}) & \text{if } \kappa = +1. \end{cases}$$

Since $\langle \cdot, \cdot \rangle_{\mathbb{M}}$ is also unitary we have $M^{-1} = \alpha U$ for some unitary U and $\alpha > 0$. Thus in all cases we have $\mathbb{S} = \tilde{U} \cdot \text{Herm}(\mathbb{C})$ for some unitary $\tilde{U}$, so the singular

value structure of every $\mathbb{S}$ is the same as that of $\text{Herm}(\mathbb{C})$. But the singular values of matrices in $\text{Herm}(\mathbb{C})$ are unstructured, since $\text{Herm}(\mathbb{C})$ contains all real symmetric positive semidefinite matrices, which may have arbitrary singular values. $\square$

8.3. Structured SVD of automorphisms. In this section we survey (without proof) some recent results on the structured SVD question:

For which scalar products does every matrix in the corresponding automorphism group $\mathbb{G}$ have a “structured” SVD, that is, an SVD $U\Sigma V^*$ such that $U, \Sigma, V \in \mathbb{G}$?

By contrast with the other structured factorizations considered in this paper, there is as yet no unified treatment of the structured SVD question for any large class of scalar products. However, for particular scalar products there have recently been some positive results, using arguments specific to the situation in each case. The most important of these results are for the three varieties of symplectic matrices seen in Table 2.1.

THEOREM 8.5 (symplectic SVD). *Let $\mathbb{S}$ be either the real symplectic, complex symplectic, or conjugate symplectic group of $2n \times 2n$ matrices. Then every $A \in \mathbb{S}$ has an SVD, $A = U\Sigma V^*$, where the factors U, Σ , and V are also in $\mathbb{S}$. Furthermore, Σ can always be chosen to be of the form $\Sigma = \text{diag}(D, D^{-1})$, where D is an $n \times n$ diagonal matrix with $d_{11} \geq d_{22} \geq \cdots \geq d_{nn} \geq 1$. This Σ is unique.*

This theorem can be found in [49] for real symplectic and conjugate symplectic matrices, in [14] for complex symplectic matrices, and with a different proof in [39] for all three types of symplectic matrices. Structure-preserving Jacobi algorithms for computing a symplectic SVD of a real or complex symplectic matrix are developed in [10]. Theorem 8.5 also has an analytic version, as shown in [39].

THEOREM 8.6 (analytic symplectic SVD). *Let $\mathbb{S}$ be as in Theorem 8.5, and suppose $A(t) \in \mathbb{S}$ for $t \in [a, b]$ is an analytically varying symplectic matrix. Then there is an analytic SVD, $A(t) = U(t)\Sigma(t)V(t)^*$, such that $U(t), \Sigma(t), V(t) \in \mathbb{S}$ for all $t \in [a, b]$.*

One class of automorphism groups $\mathbb{G}$ for which completely structured SVDs are not possible are the pseudo-orthogonal groups, also known as the Lorentz groups. The only positive diagonal matrix in these groups is the identity matrix; thus no nonorthogonal Lorentz matrix can ever have an SVD in which all three factors are in $\mathbb{G}$. However, there is a closely related structured decomposition of Lorentz matrices from which the singular values and vectors can be easily read off.

THEOREM 8.7 (hyperbolic CS decomposition). *Let $\mathbb{G}$ be one of the pseudo-orthogonal groups $O(p, q, \mathbb{R})$; here without loss of generality we will assume that $p \leq q$. Then any $A \in \mathbb{G}$ has a “hyperbolic CS decomposition”; that is, A has a factorization $A = U\tilde{\Sigma}V^T$, where $U, V, \tilde{\Sigma} \in \mathbb{G}$, U and V are orthogonal, and $\tilde{\Sigma}$ has the form*

$$\tilde{\Sigma} = \begin{bmatrix} C & -S & 0 \\ -S & C & 0 \\ 0 & 0 & I_{q-p} \end{bmatrix}$$

with $p \times p$ diagonal blocks C and S such that $C^2 - S^2 = I$ and $c_{ii} > s_{ii} \geq 0$ for $i = 1, \dots, p$.

Theorem 8.7 can be found in [22], together with references to earlier work on this factorization. An equivalent result is proved in [11] and also extended to the case of a smoothly varying Lorentz matrix $A(t)$, albeit with the restriction that the numbers $s_{ii}(t)$ remain distinct for all t . Note that this restriction is equivalent to assuming that

the singular values of the block matrix $\begin{bmatrix} C & -S \\ -S & C \end{bmatrix}$ remain distinct for all t . Theorem 8.7 can also be easily extended to the pseudo-unitary groups $U(p, q, \mathbb{C})$.

Characterizing which automorphism groups $\mathbb{G}$ have structured SVDs and which do not, as well as which $\mathbb{G}$ have decompositions that are “almost” structured SVDs, e.g., structured “CS-like” decompositions, is still an open question.

8.4. Structured SVD-like decomposition in Lie and Jordan algebras.

Since $\mathbb{L}$ and $\mathbb{J}$ are not closed under multiplication, we cannot in general expect SVDs of such matrices to have all three factors in $\mathbb{L}$ or $\mathbb{J}$. Instead we aim to find SVD-like decompositions that reflect the $\mathbb{L}/\mathbb{J}$ structure of the given matrix in a manner analogous to the symmetric SVD of complex symmetric matrices [9], also known as the Takagi factorization [25]. Let us begin by recalling this factorization and its skew-symmetric analogue. Note that $\text{Sym}(\mathbb{K})$ and $\text{Skew}(\mathbb{K})$ are as defined in (8.1).

THEOREM 8.8 (symmetric SVD/Takagi factorization). *Let $A \in \text{Sym}(\mathbb{C})$. Then there exist a real nonnegative diagonal matrix $\Sigma = \text{diag}(\sigma_1, \dots, \sigma_n)$ and a unitary matrix U such that $A = U\Sigma U^T$.*

We refer to [25, Cor. 4.4.4] for a proof and to [25, p. 218] and [26, sect. 3.0] for references to earlier work on this factorization. Note that the Takagi factorization is a special SVD $A = U\Sigma V^*$ in which $V = \bar{U}$. The σ_i are the singular values of A , and the columns of $U = [u_1, \dots, u_n]$ are the corresponding left singular vectors; that is, $A\bar{u}_i = \sigma_i u_i$.

THEOREM 8.9 (skew-symmetric Takagi factorization). *Suppose $A \in \text{Skew}(\mathbb{C})$. Then there exist a unitary matrix U and a block diagonal matrix*

$$(8.4) \quad \Sigma = D_1 \oplus \dots \oplus D_k \oplus 0 \oplus \dots \oplus 0$$

such that $A = U\Sigma U^T$. Here $D_j = \begin{bmatrix} 0 & z_j \\ -z_j & 0 \end{bmatrix}$ with nonzero $z_j \in \mathbb{C}$ for $j = 1, \dots, k$. If $A \in \text{Skew}(\mathbb{R})$, then U can be chosen to be real orthogonal and the z_j to be real. Note that the nonzero singular values of A are $|z_j|$, each with multiplicity two.

For a proof of this result see [25, sect. 4.4, problems 25–26].

Although the skew-symmetric Takagi factorization is not literally an SVD, it is an example of something that might reasonably be termed an “SVD-like” decomposition, since the Σ -factor reveals the singular values of A in a simple and straightforward fashion. Similarly, the real case of Theorem 8.8 may be viewed as an SVD-like decomposition. Any real symmetric A may be factored as $A = U\Sigma U^T$ with U real orthogonal and Σ real diagonal, but in general one cannot arrange for all the entries of Σ to be nonnegative. Nevertheless the singular values of A are easily recovered from Σ by taking absolute values. It is in this sense that we seek SVD-like decompositions for matrices in $\mathbb{L}$ and $\mathbb{J}$.

We also want a factorization that reflects the structure of the given matrix from $\mathbb{L}$ or $\mathbb{J}$. Observe that the symmetric/skew-symmetric structure of the matrix A in the Takagi factorizations is not expressed by structure in each individual factor, but rather by the *overall* form of the decomposition, i.e., as a congruence of a condensed symmetric or skew-symmetric matrix. In a scalar product space the analogue of congruence is $\star$ -congruence, and it is not hard to see that $\mathbb{L}$ and $\mathbb{J}$ -structure is preserved by arbitrary $\star$ -congruence whenever the scalar product is orthosymmetric; that is, for $\mathbb{S} = \mathbb{L}$ or $\mathbb{J}$ and any $P \in \mathbb{K}^{n \times n}$, $A \in \mathbb{S} \Rightarrow PAP^* \in \mathbb{S}$. Indeed, this property gives another characterization of orthosymmetric scalar products (see Theorem A.5(f)). The next result gives a “Takagi-like” factorization for any $A \in \mathbb{L}$ or $\mathbb{J}$ using $\star$ -congruence of matrices in $\mathbb{L}$ or $\mathbb{J}$ that are simple enough to reveal the singular values of A . Note that we consider only scalar products that are both orthosymmetric and unitary.

THEOREM 8.10 (structured SVD-like decomposition for $\mathbb{L}$ and $\mathbb{J}$). *Let $\mathbb{S}$ be either the Lie algebra $\mathbb{L}$ or Jordan algebra $\mathbb{J}$ of a scalar product $\langle \cdot, \cdot \rangle_{\mathbb{M}}$ on $\mathbb{K}^n$ that is both orthosymmetric and unitary. Then every $A \in \mathbb{S}$ has an SVD-like decomposition of the form $A = U\Sigma U^*$, where U is unitary and $\Sigma \in \mathbb{S}$ reveals the singular values of A in the following sense: with $\alpha > 0$ such that αM is unitary, the matrix $\alpha M\Sigma$ has the same singular values as A and is either diagonal or block diagonal with diagonal blocks of the form $\begin{bmatrix} 0 & z \\ -z & 0 \end{bmatrix}$. If $\mathbb{K} = \mathbb{R}$, then U and Σ may be chosen to be real.*

Proof. Orthosymmetry of a bilinear $\langle \cdot, \cdot \rangle_{\mathbb{M}}$ means that $M^T = \varepsilon M$ with $\varepsilon = \pm 1$. Also by definition $A \in \mathbb{S} \Rightarrow A^* = \kappa A$ with $\kappa = \pm 1$. Thus if $\tilde{A} := MA$, then by (8.2) we have $\tilde{A} \in \tilde{\mathbb{S}}$, where

$$\tilde{\mathbb{S}} = \begin{cases} \text{Skew}(\mathbb{K}) & \text{if } \varepsilon \neq \kappa, \\ \text{Sym}(\mathbb{K}) & \text{if } \varepsilon = \kappa. \end{cases}$$

From Theorems 8.8 and 8.9 there exist V unitary and $\tilde{\Sigma} \in \tilde{\mathbb{S}}$ such that $\tilde{A} = V\tilde{\Sigma}V^T$; this $\tilde{\Sigma}$ is real diagonal when $\tilde{\mathbb{S}} = \text{Sym}(\mathbb{K})$, and block diagonal as in (8.4) when $\tilde{\mathbb{S}} = \text{Skew}(\mathbb{K})$. In either case, $\tilde{\Sigma}$ reveals the singular values of $\tilde{A}$. From this factorization of $\tilde{A}$ we get

$$A = M^{-1}\tilde{A} = M^{-1}V\tilde{\Sigma}V^T = (M^{-1}VM)M^{-1}\tilde{\Sigma}V^T = (V^T)^*(M^{-1}\tilde{\Sigma})V^T = W^*\Sigma W,$$

where $W = V^T$ is unitary and $\Sigma = M^{-1}\tilde{\Sigma} \in \mathbb{S}$.

A similar argument also leads to a factorization $A = W^*\Sigma W$ when $\langle \cdot, \cdot \rangle_{\mathbb{M}}$ is sesquilinear. In this case, orthosymmetry means that $M^* = \gamma M$ for some $|\gamma| = 1$. Using once again that $A \in \mathbb{S} \Rightarrow A^* = \kappa A$ with $\kappa = \pm 1$, we have by (8.3) that $\tilde{A} := \sqrt{\gamma}MA$ is an element of

$$\tilde{\mathbb{S}} = \begin{cases} \text{Herm}(\mathbb{C}) & \text{if } \kappa = 1, \\ i \cdot \text{Herm}(\mathbb{C}) & \text{if } \kappa = -1. \end{cases}$$

Hence there exist a unitary V and diagonal $\tilde{\Sigma} \in \tilde{\mathbb{S}}$ such that $\tilde{A} = V\tilde{\Sigma}V^*$. Thus

$$A = \gamma^{-1/2}M^{-1}\tilde{A} = \gamma^{-1/2}(M^{-1}VM)M^{-1}\tilde{\Sigma}V^* = (V^*)^*(\gamma^{-1/2}M^{-1}\tilde{\Sigma})V^* = W^*\Sigma W,$$

where $W = V^*$ is unitary and $\Sigma = \gamma^{-1/2}M^{-1}\tilde{\Sigma} \in \mathbb{S}$.

In both the bilinear and sesquilinear cases, the Takagi-like form $A = U\Sigma U^*$ with U unitary is now obtained by letting $U = W^*$ and using the involutory property (e) in Definition 2.4 for orthosymmetric scalar products.

Finally, we invoke the hypothesis that $\langle \cdot, \cdot \rangle_{\mathbb{M}}$ is unitary to see how the factor Σ reveals the singular values of $A = U\Sigma U^*$. Since U is unitary, so is U^* by Definition 2.3(b), and thus A and Σ share the same singular values. Property (d) in Definition 2.3 gives us an $\alpha > 0$ such that αM is unitary, and thus $\alpha M\Sigma$ also has the same singular values as A . But $\alpha M\Sigma$, being a scalar multiple of $\tilde{\Sigma}$, is either diagonal or block diagonal of the form (8.4), and thus reveals the singular values of A . $\square$

9. Concluding remarks. This work adds to the growing literature on matrices in automorphism groups, Lie algebras, and Jordan algebras associated with a scalar product. Results on structured factorizations have been given here with as few assumptions on the scalar products as possible. One of our main contributions is the recognition that for many results the appropriate assumption is that the scalar

product is either unitary or orthosymmetric, and we have given a number of different characterizations of these two classes of scalar products.

Some open questions have been identified. A major area for future work concerns the numerical computation of structured factorizations in a way that exploits—and retains, in the presence of rounding errors—the structure. Some recent work along these lines can be found in the references, but much remains to be done.

Appendix A. Special classes of scalar products.

Many useful properties, such as the involutory property $(A^\star)^\star = A$, do not hold in every scalar product space. In this appendix we consider a number of such properties, including all those listed earlier in section 2.3, and show that they cluster together into two groups of equivalent properties, thereby delineating two natural classes of scalar products. We have termed these classes *orthosymmetric* and *unitary*, and believe that their identification has simplified the presentation in this paper and will also help to clarify existing results in the literature. Note that all the “classical” examples of scalar products listed in Table 2.1 are both orthosymmetric and unitary, as will easily be seen from Theorems A.5 and A.7.

A.1. Preliminaries. To demonstrate the equivalence of various scalar product properties, we need a flexible way to detect when a matrix A is a scalar multiple of the identity. It is well known that when A commutes with all of $\mathbb{K}^{n \times n}$, then $A = \alpha I$ for some $\alpha \in \mathbb{K}$. There are many other sets besides $\mathbb{K}^{n \times n}$, though, that suffice to give the same conclusion.

DEFINITION A.1. *A set of matrices $\mathcal{S} \subseteq \mathbb{K}^{n \times n}$ will be called a CS-set for $\mathbb{K}^{n \times n}$ if the centralizer of $\mathcal{S}$ consists only of the scalar multiples⁴ of I ; that is,*

$$BS = SB \text{ for all } S \in \mathcal{S} \implies B = \alpha I \text{ for some } \alpha \in \mathbb{K}.$$

The following lemma describes a number of useful examples⁵ of CS-sets for $\mathbb{R}^{n \times n}$ and $\mathbb{C}^{n \times n}$. For this lemma, D denotes an arbitrary diagonal matrix in $\mathbb{K}^{n \times n}$ with *distinct* diagonal entries, D_+ denotes a diagonal matrix with distinct positive diagonal entries,

$$N = \begin{bmatrix} 0 & 1 & & \\ & 0 & 1 & \\ & & \ddots & \ddots \\ & & & 1 & 0 \end{bmatrix}$$

is the $n \times n$ nilpotent Jordan block,

$$C = \begin{bmatrix} 0 & 1 & & \\ & 0 & 1 & \\ & & \ddots & \ddots \\ 1 & & & 1 & 0 \end{bmatrix}$$

is the cyclic permutation matrix, $E = \begin{bmatrix} 0 & 1 \\ 1 & 0 \end{bmatrix} \oplus I_{n-2}$, and $F = \begin{bmatrix} 0 & -1 \\ 1 & 0 \end{bmatrix} \oplus I_{n-2}$.

LEMMA A.2. *Suppose $\mathcal{S} \subseteq \mathbb{K}^{n \times n}$. Then the following hold.*

- (a) $\mathcal{S}$ contains a CS-set $\implies \mathcal{S}$ is a CS-set.
- (b) Let $\bar{\mathcal{S}}$ denote $\{\bar{A} : A \in \mathcal{S}\}$. If $\mathcal{S}$ is a CS-set for $\mathbb{C}^{n \times n}$, then so is $\bar{\mathcal{S}}$.

⁴One may think of “CS” as standing for either “Commuting implies Scalar,” or “Centralizer equals the Scalars.”

⁵Another important source of CS-sets for $\mathbb{C}^{n \times n}$ is the classical “Schur’s lemma” [34], [48] from representation theory: Any $\mathcal{S} \subseteq \mathbb{C}^{n \times n}$ for which there is no nontrivial $\mathcal{S}$ -invariant subspace in $\mathbb{C}^n$ is a CS-set for $\mathbb{C}^{n \times n}$. Thus the matrices in any irreducible representation of a finite group will be a CS-set.

- (c) Any vector space basis for $\mathbb{K}^{n \times n}$ or algebra generating set for $\mathbb{K}^{n \times n}$ is a CS-set for $\mathbb{K}^{n \times n}$. More generally, any set whose span (either in the vector space sense or the algebra sense) contains a CS-set is a CS-set.
- (d) Each of the finite sets $\{D, N\}$, $\{D, N + N^T\}$, $\{D_+, 3I + N + N^T\}$, and $\{C, E, F\}$ is a CS-set for $\mathbb{K}^{n \times n}$.
- (e) Any open subset $\mathcal{S} \subseteq \mathbb{K}^{n \times n}$ is a CS-set. (Indeed any open subset of $\mathbb{R}^{n \times n}$ is a CS-set for $\mathbb{C}^{n \times n}$.)
- (f) The sets of all unitary matrices, all Hermitian matrices, all Hermitian positive semidefinite matrices, and all Hermitian positive definite matrices are CS-sets for $\mathbb{C}^{n \times n}$. The sets of all real orthogonal matrices and all real symmetric matrices are CS-sets for $\mathbb{R}^{n \times n}$ and for $\mathbb{C}^{n \times n}$.

Proof.

- (a) This is an immediate consequence of Definition A.1.
- (b) $B\bar{S} = \bar{S}B$ for all $\bar{S} \in \bar{\mathcal{S}} \Rightarrow \bar{B}S = S\bar{B}$ for all $S \in \mathcal{S}$. But $\mathcal{S}$ is a CS-set, so $\bar{B} = \alpha I$ or, equivalently, $B = \bar{\alpha}I$. Thus $\bar{\mathcal{S}}$ is a CS-set.
- (c) If B commutes with either a vector space basis or an algebra generating set for $\mathbb{K}^{n \times n}$, then it commutes with all of $\mathbb{K}^{n \times n}$, and hence $B = \alpha I$.
- (d) Any matrix B that commutes with D must itself be a diagonal matrix, and any diagonal B that commutes with N must have equal diagonal entries, so that $B = \alpha I$. Thus $\mathcal{S} = \{D, N\}$ is a CS-set. Similar arguments show that $\{D, N + N^T\}$ and $\{D_+, 3I + N + N^T\}$ are also CS-sets. To see that $\{C, E, F\}$ is a CS-set, first observe that a matrix B commutes with C if and only if it is a polynomial in C , i.e., if and only if B is a circulant matrix. But any circulant B that commutes with E must be of the form $B = \alpha I + \beta K$, where K is defined by

$$K_{ij} = \begin{cases} 0 & \text{if } i = j, \\ 1 & \text{if } i \neq j. \end{cases}$$

Finally, $B = \alpha I + \beta K$ commuting with F forces $\beta = 0$, and thus $B = \alpha I$, showing that $\{C, E, F\}$ is a CS-set.

- (e) This follows from (a) and (c), since any open subset of $\mathbb{K}^{n \times n}$ contains a vector space basis for $\mathbb{K}^{n \times n}$.
- (f) This follows from (a) and (d) by observing that $\{D, N + N^T\}$ consists of two real symmetric matrices, $\{D_+, 3I + N + N^T\}$ consists of two real symmetric positive definite matrices, and $\{C, E, F\}$ consists of three real orthogonal matrices. $\square$

A second simple result that is needed to show the equivalence of various scalar product properties is the following lemma.

LEMMA A.3. Let $M \in \mathbb{K}^{n \times n}$ be a nonzero matrix. Then

1. $M^T = \alpha M$ for some $\alpha \in \mathbb{K} \Leftrightarrow M^T = \pm M$,
2. $M^* = \alpha M$ for some $\alpha \in \mathbb{K} \Leftrightarrow M^* = \alpha M$ for some $|\alpha| = 1$
 $\Leftrightarrow M = \beta H$ for some Hermitian H and $|\beta| = 1$,
3. $MM^* = \alpha I$ for some $\alpha \in \mathbb{K} \Leftrightarrow M = \beta U$ for some unitary U and $\beta > 0$.

Proof. Since the proofs of the reverse implications ($\Leftarrow$) in 1, 2, and 3 are immediate, we include only the proofs of the forward implications ($\Rightarrow$) in each case.

1. $M^T = \alpha M \Rightarrow M = (M^T)^T = (\alpha M)^T = \alpha M^T = \alpha^2 M \Rightarrow \alpha^2 = 1 \Rightarrow \alpha = \pm 1$.
2. $M^* = \alpha M \Rightarrow M = (M^*)^* = (\alpha M)^* = \bar{\alpha} M^* = |\alpha|^2 M \Rightarrow |\alpha|^2 = 1 \Rightarrow |\alpha| = 1$.
 To see the second implication, let $H = \sqrt{\alpha} M$, where $\sqrt{\alpha}$ is either of the two square roots of α on the unit circle. It is easy to check that H is Hermitian, and that $M = \beta H$ with $\beta = (\sqrt{\alpha})^{-1}$ on the unit circle.

3. MM^* is positive semidefinite, so $\alpha \geq 0$; then $M \neq 0$ implies $\alpha > 0$. It follows that $U = \frac{1}{\sqrt{\alpha}}M$ is unitary, and thus $M = \beta U$ with $\beta = \sqrt{\alpha} > 0$. $\square$

A.2. Orthosymmetric scalar products. In Shaw [48], scalar products that enjoy property (b) in Theorem A.5 are called “orthosymmetric.” We adopt this name in the following definition, which extends Definition 2.4.

DEFINITION A.4 (orthosymmetric scalar product). *A scalar product is said to be orthosymmetric if it satisfies any one (and hence all) of the seven equivalent properties in Theorem A.5.*

THEOREM A.5. *For a scalar product $\langle \cdot, \cdot \rangle_M$ on $\mathbb{K}^n$, the following are equivalent:*

- (a) *Adjoint with respect to $\langle \cdot, \cdot \rangle_M$ is involutory; that is, $(A^\star)^\star = A$ for all $A \in \mathbb{K}^{n \times n}$.*
- (a') *$(A^\star)^\star = A$ for all A in some CS-set for $\mathbb{K}^{n \times n}$.*
- (b) *Vector orthogonality is a symmetric relation; that is,*

$$\langle x, y \rangle_M = 0 \iff \langle y, x \rangle_M = 0 \text{ for all } x, y \in \mathbb{K}^n.$$
- (c) $\mathbb{K}^{n \times n} = \mathbb{L} \oplus \mathbb{J}$.
- (d) *For bilinear forms, $M^T = \pm M$. For sesquilinear forms, $M^* = \alpha M$ with $\alpha \in \mathbb{C}$, $|\alpha| = 1$; equivalently, $M = \beta H$ with $\beta \in \mathbb{C}$, $|\beta| = 1$, and Hermitian H .*
- (e) *There exists some CS-set for $\mathbb{K}^{n \times n}$ with the property that every matrix A in this CS-set can be factored as $A = WS$ with $W \in \mathbb{G}$ and $S \in \mathbb{J}$.*
- (f) $\mathbb{L}$ and $\mathbb{J}$ are preserved by arbitrary $\star$ -congruence; that is, for $\mathbb{S} = \mathbb{L}$ or $\mathbb{J}$ and $P \in \mathbb{K}^{n \times n}$, $B \in \mathbb{S} \Rightarrow PBP^\star \in \mathbb{S}$.

Proof. Using (2.2) we have

$$(A.1) \quad (A^\star)^\star = \begin{cases} (M^{-1}M^T)A(M^{-1}M^T)^{-1} & \text{for bilinear forms,} \\ (M^{-1}M^*)A(M^{-1}M^*)^{-1} & \text{for sesquilinear forms.} \end{cases}$$

Hence,

$$(A.2) \quad (A^\star)^\star = A \iff \begin{cases} (M^{-1}M^T)A = A(M^{-1}M^T) & \text{for bilinear forms,} \\ (M^{-1}M^*)A = A(M^{-1}M^*) & \text{for sesquilinear forms.} \end{cases}$$

(a) $\Leftrightarrow$ (a') $\Leftrightarrow$ (d)

(a) $\Rightarrow$ (a'): Obvious.

(a') $\Rightarrow$ (d): Equation (A.2) holding for all A in some CS-set means that $M^{-1}M^T = \alpha I$ (resp., $M^{-1}M^* = \alpha I$). The desired conclusion now follows from Lemma A.3.

(d) $\Rightarrow$ (a): This follows from a straightforward substitution into (A.1).

(a) $\Leftrightarrow$ (c)

(a) $\Rightarrow$ (c): For any scalar product, $\mathbb{L} \cap \mathbb{J} = \{0\}$; if $B \in \mathbb{L} \cap \mathbb{J}$, then $-B = B^\star = B$, and thus $B = 0$. Now suppose that (a) holds and consider an arbitrary $A \in \mathbb{K}^{n \times n}$. Define $L = \frac{1}{2}(A - A^\star)$ and $S = \frac{1}{2}(A + A^\star)$ so that $A = L + S$. From $(A^\star)^\star = A$, we conclude that $L^\star = -L$ so that $L \in \mathbb{L}$. Similarly one sees that $S \in \mathbb{J}$. The decomposition $A = L + S$ shows that $\mathbb{K}^{n \times n} = \mathbb{L} + \mathbb{J}$ and because $\mathbb{L} \cap \mathbb{J} = \{0\}$, the sum is direct.

(c) $\Rightarrow$ (a): $A = L + S \Rightarrow A^\star = L^\star + S^\star = -L + S \Rightarrow (A^\star)^\star = (-L)^\star + S^\star = L + S = A$.

(b) $\Leftrightarrow$ (d)

(b) $\Rightarrow$ (d): Suppose $\langle \cdot, \cdot \rangle_M$ is a bilinear form. Letting $y = Mw$, we have

$$x^T y = 0 \iff x^T M w = 0 \stackrel{(b)}{\iff} w^T M x = 0 \iff x^T M^T w = 0 \iff x^T (M^T M^{-1}) y = 0.$$

A similar argument for sesquilinear forms shows that $x^*y = 0 \Leftrightarrow x^*(M^*M^{-1})y = 0$. Thus, property (b) implies that

$$\langle x, y \rangle_I = 0 \Leftrightarrow \langle x, y \rangle_B = 0, \text{ where } B = \begin{cases} M^T M^{-1} & \text{for bilinear forms,} \\ M^* M^{-1} & \text{for sesquilinear forms.} \end{cases}$$

Using this relationship we can now probe the entries of B with various pairs x, y such that $\langle x, y \rangle_I = 0$. Let $x = e_i$ and $y = e_j$ with $i \neq j$. Then $B_{ij} = \langle e_i, e_j \rangle_B = 0$, and thus B must be a diagonal matrix. Next, let $x = e_i + e_j$ and $y = e_i - e_j$ with $i \neq j$. Then

$$0 = \langle e_i + e_j, e_i - e_j \rangle_B = B_{ii} + B_{ji} - B_{ij} - B_{jj} = B_{ii} - B_{jj};$$

hence $B_{ii} = B_{jj}$ for all $i \neq j$. Thus $B = \alpha I$ for some nonzero $\alpha \in \mathbb{K}$, and the desired conclusion follows from Lemma A.3.

(d) $\Rightarrow$ (b): This direction is a straightforward verification. For bilinear forms,

$$\langle x, y \rangle_M = 0 \Leftrightarrow x^T M y = 0 \Leftrightarrow (x^T M y)^T = 0 \Leftrightarrow \pm(y^T M x) = 0 \Leftrightarrow \langle y, x \rangle_M = 0,$$

and for sesquilinear forms,

$$\langle x, y \rangle_M = 0 \Leftrightarrow x^* M y = 0 \Leftrightarrow (x^* M y)^* = 0 \Leftrightarrow \alpha(y^* M x) = 0 \Leftrightarrow \langle y, x \rangle_M = 0.$$

(e) $\Leftrightarrow$ (a)

(e) $\Rightarrow$ (a): The proof of this equivalence is just a slightly modified version of the proof of Theorem 6.4. For all A in our CS-set we have

$$(A^*)^* = (S^* W^*)^* = (S W^{-1})^* = W^{-*} S^* = (W^{-1})^{-1} S = W S = A,$$

and thus (a') holds. That (a') implies (a) was shown earlier.

(a) $\Rightarrow$ (e): The continuity of the eigenvalues of $A^* A$ implies that there is an open neighborhood $\mathcal{U}$ of the identity in which $A^* A$ has no eigenvalues on $\mathbb{R}^-$. Thus by Theorem 6.2 every A in the CS-set $\mathcal{U}$ can be factored as $A = W S$ with $W \in \mathbb{G}$ and $S \in \mathbb{J}$.

(a) $\Leftrightarrow$ (f)

(a) $\Rightarrow$ (f): Let $B \in \mathbb{S}$ so that $B^* = \pm B$. Then $(P B P^*)^* = (P^*)^* B^* P^* = \pm P B P^*$, and thus $P B P^* \in \mathbb{S}$.

(f) $\Rightarrow$ (a): Consider $\mathbb{S} = \mathbb{J}$ and $B = I \in \mathbb{J}$. Then (f) implies that $P P^* \in \mathbb{J}$ for any $P \in \mathbb{K}^{n \times n}$, and thus $P P^* = (P P^*)^* = (P^*)^* P^*$. Since P^* is nonsingular for any nonsingular P , we have $P = (P^*)^*$ for every nonsingular P . Thus by Lemma A.2(e) we have property (a'), which was previously shown to be equivalent to (a). $\square$

A.3. Unitary scalar products. Finally, we prove the equivalence of a second set of scalar product space properties. We adopt the name “unitary” for the scalar products satisfying these properties because of (b) and (e) in Theorem A.7.

DEFINITION A.6 (unitary scalar product). *A scalar product is said to be unitary if it satisfies any one (and hence all) of the six equivalent properties in Theorem A.7.*

THEOREM A.7. *For a scalar product $\langle \cdot, \cdot \rangle_M$ on $\mathbb{K}^n$, the following are equivalent:*

(a) $(A^*)^* = (A^*)^*$ for all $A \in \mathbb{K}^{n \times n}$.

(a') $(A^*)^* = (A^*)^*$ for all A in some CS-set for $\mathbb{K}^{n \times n}$.

(b) Adjoint preserves unitarity: U unitary $\Rightarrow U^*$ is unitary.

(c) Adjoint preserves Hermitian structure: H Hermitian $\Rightarrow H^*$ is Hermitian.

(d) *Adjoint preserves Hermitian positive (semi)definite structure: H Hermitian positive (semi)definite $\Rightarrow H^\star$ is Hermitian positive (semi)definite.*

(e) *$M = \beta U$ for some unitary U and $\beta > 0$.*

Proof. From (2.2) it follows that

$$(A^\star)^\star = \begin{cases} M^{-1}\bar{A}M & \text{for bilinear forms,} \\ M^{-1}AM & \text{for sesquilinear forms} \end{cases}$$

and

$$(A^\star)^* = \begin{cases} M^*\bar{A}M^{-*} & \text{for bilinear forms,} \\ M^*AM^{-*} & \text{for sesquilinear forms.} \end{cases}$$

Thus for any individual matrix $A \in \mathbb{K}^{n \times n}$ we have

$$(A.3) \quad (A^\star)^\star = (A^\star)^* \iff \begin{cases} \bar{A}(MM^*) = (MM^*)\bar{A} & \text{for bilinear forms,} \\ A(MM^*) = (MM^*)A & \text{for sesquilinear forms.} \end{cases}$$

(a) $\Leftrightarrow$ (a')

(a) $\Rightarrow$ (a'): This implication is trivial.

(a') $\Rightarrow$ (a): Suppose $(A^\star)^\star = (A^\star)^*$ holds for all A in some CS-set for $\mathbb{K}^{n \times n}$. Then from (A.3) we conclude that $MM^* = \alpha I$ and hence that the two sides of (A.3) hold for all $A \in \mathbb{K}^{n \times n}$.

(a) $\Leftrightarrow$ (b)

(a) $\Rightarrow$ (b): $U^* = U^{-1} \Rightarrow (U^*)^\star = (U^{-1})^\star \xrightarrow{(a)} (U^\star)^* = (U^\star)^{-1} \Rightarrow U^\star$ is unitary.

(b) $\Rightarrow$ (a): Suppose U , and hence also $U^\star$, is unitary. Then we have $(U^\star)^* = (U^\star)^{-1} = (U^{-1})^\star = (U^*)^\star$, showing that $(A^\star)^\star = (A^\star)^*$ for all unitary A . But from Lemma A.2(f), the set of all unitaries is a CS-set for $\mathbb{K}^{n \times n}$; thus (a') holds, and hence also (a).

(a) $\Leftrightarrow$ (c)

(a) $\Rightarrow$ (c): $H^* = H \Rightarrow (H^*)^\star = H^\star \xrightarrow{(a)} (H^\star)^* = H^\star \Rightarrow H^\star$ is Hermitian.

(c) $\Rightarrow$ (a): Suppose H , and therefore also $H^\star$, is Hermitian. Then we have $(H^\star)^* = H^\star = (H^*)^\star$, and thus $(A^\star)^\star = (A^\star)^*$ for all Hermitian A . But from Lemma A.2(f), the set of all Hermitian matrices is a CS-set for $\mathbb{K}^{n \times n}$; thus (a') holds, and hence also (a).

(a) $\Leftrightarrow$ (d)

(a) $\Rightarrow$ (d): Because (a) $\Rightarrow$ (c), we just need to show that positive (semi)definiteness is preserved by adjoint. But for H Hermitian, $H^\star$ and H are similar by definition of the adjoint and hence the eigenvalues of $H^\star$ and H are the same.

(d) $\Rightarrow$ (a): This argument is the same as that for (c) $\Rightarrow$ (a), using the fact that the set of all Hermitian positive (semi)definite matrices is a CS-set for $\mathbb{K}^{n \times n}$.

(a) $\Leftrightarrow$ (e)

(a) $\Rightarrow$ (e): Suppose $(A^\star)^\star = (A^\star)^*$ holds for all $A \in \mathbb{K}^{n \times n}$. Then we can conclude from (A.3) that $MM^* = \alpha I$ for some $\alpha \in \mathbb{K}$, and thus from Lemma A.3 that $M = \beta U$ for some unitary U and $\beta > 0$.

(e) $\Rightarrow$ (a): $M = \beta U \Rightarrow MM^* = (\beta U)(\bar{\beta}U^*) = \beta^2 I$. Then by (A.3) we have $(A^\star)^\star = (A^\star)^*$ for all A . $\square$

REFERENCES

- [1] G. AMMAR, C. MEHL, AND V. MEHRMANN, *Schur-like forms for matrix Lie groups, Lie algebras, and Jordan algebras*, Linear Algebra Appl., 287 (1999), pp. 11–39.
- [2] P. BENNER, V. MEHRMANN, AND H. XU, *A numerically stable, structure preserving method for computing the eigenvalues of real Hamiltonian or symplectic pencils*, Numer. Math., 78 (1998), pp. 329–358.
- [3] R. BHATIA, *Matrix factorizations and their perturbations*, Linear Algebra Appl., 197/198 (1994), pp. 245–276.
- [4] Å. BJÖRCK AND S. HAMMARLING, *A Schur method for the square root of a matrix*, Linear Algebra Appl., 52/53 (1983), pp. 127–140.
- [5] Y. BOLSHAKOV AND B. REICHSTEIN, *Unitary equivalence in an indefinite scalar product: An analogue of singular-value decomposition*, Linear Algebra Appl., 222 (1995), pp. 155–226.
- [6] Y. BOLSHAKOV, C. V. M. VAN DER MEE, A. C. M. RAN, B. REICHSTEIN, AND L. RODMAN, *Extension of isometries in finite-dimensional indefinite scalar product spaces and polar decompositions*, SIAM J. Matrix Anal. Appl., 18 (1997), pp. 752–774.
- [7] Y. BOLSHAKOV, C. V. M. VAN DER MEE, A. C. M. RAN, B. REICHSTEIN, AND L. RODMAN, *Polar decompositions in finite dimensional indefinite scalar product spaces: General theory*, Linear Algebra Appl., 261 (1997), pp. 91–141.
- [8] A. BUNSE-GERSTNER, R. BYERS, AND V. MEHRMANN, *A chart of numerical methods for structured eigenvalue problems*, SIAM J. Matrix Anal. Appl., 13 (1992), pp. 419–453.
- [9] A. BUNSE-GERSTNER AND W. B. GRAGG, *Singular value decompositions of complex symmetric matrices*, J. Comput. Appl. Math., 21 (1988), pp. 41–54.
- [10] A. CHAIYAKARN, *Structure Preserving Algorithms for Computing the Symplectic Singular Value Decomposition*, Ph.D. thesis, Western Michigan University, Kalamazoo, MI, 2005.
- [11] L. DIECI, C. ELIA, AND L. LOPEZ, *Smooth SVD on the Lorentz group with application to computation of Lyapunov exponents*, J. Comput. Appl. Math., 164–165 (2004), pp. 255–264.
- [12] D. Ž. DJOKOVIĆ, J. PATERA, P. WINTERNITZ, AND H. ZASSENHAUS, *Normal forms of elements of classical real and complex Lie and Jordan algebras*, J. Math. Phys., 24 (1983), pp. 1363–1374.
- [13] D. ĐOKOVIĆ, *On the product of two alternating matrices*, Amer. Math. Monthly, 98 (1991), pp. 935–936.
- [14] H. FASSBENDER AND K. IKRAMOV, *Several observations on symplectic, Hamiltonian, and skew-Hamiltonian matrices*, Linear Algebra Appl., 400 (2005), pp. 15–29.
- [15] H. FASSBENDER, D. S. MACKEY, AND N. MACKEY, *Hamilton and Jacobi come full circle: Jacobi algorithms for structured Hamiltonian problems*, Linear Algebra Appl., 332–334 (2001), pp. 37–80.
- [16] H. FASSBENDER, D. S. MACKEY, N. MACKEY, AND H. XU, *Hamiltonian square roots of skew-Hamiltonian matrices*, Linear Algebra Appl., 287 (1999), pp. 125–159.
- [17] H. FASSBENDER, D. S. MACKEY, N. MACKEY, AND H. XU, *Real and Complex Hamiltonian Square Roots of Skew-Hamiltonian Matrices*, Tech. report 92, Department of Mathematics & Statistics, Western Michigan University, Kalamazoo, MI, 1999.
- [18] I. GOHBERG, P. LANCASTER, AND L. RODMAN, *Matrices and Indefinite Scalar Products*, Oper. Theory Adv. Appl. 8, Birkhäuser-Verlag, Basel, Switzerland, 1983.
- [19] G. H. GOLUB AND C. F. VAN LOAN, *Matrix Computations*, 3rd ed., The Johns Hopkins University Press, Baltimore, MD, 1996.
- [20] N. J. HIGHAM, *The matrix sign decomposition and its relation to the polar decomposition*, Linear Algebra Appl., 212/213 (1994), pp. 3–20.
- [21] N. J. HIGHAM, *Stable iterations for the matrix square root*, Numer. Algorithms, 15 (1997), pp. 227–242.
- [22] N. J. HIGHAM, *J-orthogonal matrices: Properties and generation*, SIAM Rev., 45 (2003), pp. 504–519.
- [23] N. J. HIGHAM, D. S. MACKEY, N. MACKEY, AND F. TISSEUR, *Computing the polar decomposition and the matrix sign decomposition in matrix groups*, SIAM J. Matrix Anal. Appl., 25 (2004), pp. 1178–1192.
- [24] N. J. HIGHAM, D. S. MACKEY, N. MACKEY, AND F. TISSEUR, *Functions preserving matrix groups and iterations for the matrix square root*, SIAM J. Matrix Anal. Appl., 26 (2005), pp. 849–877.
- [25] R. A. HORN AND C. R. JOHNSON, *Matrix Analysis*, Cambridge University Press, Cambridge, UK, 1985.
- [26] R. A. HORN AND C. R. JOHNSON, *Topics in Matrix Analysis*, Cambridge University Press, Cambridge, UK, 1991.

- [27] R. A. HORN AND D. I. MERINO, *Contragredient equivalence: A canonical form and some applications*, Linear Algebra Appl., 214 (1995), pp. 43–92.
- [28] R. A. HORN, D. I. MERINO, AND D. SERRE, *The S-Orthogonal Groups*, manuscript, 2004.
- [29] K. D. IKRAMOV, *Hamiltonian square roots of skew-Hamiltonian matrices revisited*, Linear Algebra Appl., 325 (2001), pp. 101–107.
- [30] A. ISERLES AND A. ZANNA, *Efficient computation of the matrix exponential by generalized polar decompositions*, SIAM J. Numer. Anal., 42 (2005), pp. 2218–2256.
- [31] I. KAPLANSKY, *Algebraic polar decomposition*, SIAM J. Matrix Anal. Appl., 11 (1990), pp. 213–217.
- [32] C. KENNEY AND A. J. LAUB, *Rational iterative methods for the matrix sign function*, SIAM J. Matrix Anal. Appl., 12 (1991), pp. 273–291.
- [33] U. KINTZEL, *Procrustes problems in finite dimensional indefinite scalar product spaces*, Linear Algebra Appl., 402 (2005), pp. 1–28.
- [34] S. LANG, *Linear Algebra*, 3rd ed., Springer-Verlag, New York, 1987.
- [35] J. D. LAWSON, *Polar and Ol'shanskii decompositions*, J. Reine Angew. Math., 448 (1994), pp. 191–219.
- [36] W.-W. LIN, V. MEHRMANN, AND H. XU, *Canonical forms for Hamiltonian and symplectic matrices and pencils*, Linear Algebra Appl., 302/303 (1999), pp. 469–533.
- [37] D. S. MACKEY AND N. MACKEY, *On the Determinant of Symplectic Matrices*, Numerical Analysis Report 422, Manchester Centre for Computational Mathematics, Manchester, UK, 2003.
- [38] D. S. MACKEY, N. MACKEY, AND D. M. DUNLAVY, *Structure preserving algorithms for perplectic eigenproblems*, Electron. J. Linear Algebra, 13 (2005), pp. 10–39.
- [39] D. S. MACKEY, N. MACKEY, AND V. MEHRMANN, *Analytic SVD of Symplectic Matrices*, Tech. report, Technische Universität Berlin, Berlin, Germany, 2004, in preparation.
- [40] C. MEHL, *Condensed forms for skew-Hamiltonian/Hamiltonian pencils*, SIAM J. Matrix Anal. Appl., 21 (1999), pp. 454–476.
- [41] C. MEHL, V. MEHRMANN, AND H. XU, *Canonical forms for doubly structured matrices and pencils*, Electron. J. Linear Algebra, 7 (2000), pp. 112–151.
- [42] C. MEHL, A. C. M. RAN, AND L. RODMAN, *Polar decompositions of normal operators in indefinite inner product spaces*, in Operator Theory in Krein Spaces and Nonlinear Eigenvalue Problems, Oper. Theory Adv. Appl. 162, Birkhäuser-Verlag, Basel, Switzerland, 2006, pp. 277–292.
- [43] V. MEHRMANN AND H. XU, *Structured Jordan canonical forms for structured matrices that are Hermitian, skew-Hermitian, or unitary with respect to indefinite inner products*, Electron. J. Linear Algebra, 5 (1999), pp. 67–103.
- [44] V. L. MEHRMANN, *The Autonomous Linear Quadratic Control Problem—Theory and Numerical Solution*, Lecture Notes in Control and Inform. Sci. 163, Springer-Verlag, Berlin, 1991.
- [45] K. R. MEYER AND G. R. HALL, *Introduction to Hamiltonian Dynamical Systems and the N-Body Problem*, Appl. Math. Sci. 90, Springer-Verlag, New York, 1991.
- [46] H. Z. MUNTKE-KAAS, G. R. W. QUISPÉL, AND A. ZANNA, *Generalized polar decomposition on Lie groups with involutive automorphisms*, Found. Comput. Math., 1 (2001), pp. 297–324.
- [47] J. D. ROBERTS, *Linear model reduction and solution of the algebraic Riccati equation by use of the sign function*, Internat. J. Control, 32 (1980), pp. 677–687.
- [48] R. SHAW, *Linear Algebra and Group Representations. Vol. I*, Academic Press Inc. [Harcourt Brace Jovanovich], London, New York, 1982.
- [49] H. XU, *An SVD-like matrix decomposition and its applications*, Linear Algebra Appl., 368 (2003), pp. 1–24.
- [50] T. YAMANOUCHI, *Quantum mechanics*, in Mathematics Applied to Physics, E. Roubine, ed., Springer-Verlag, New York, 1970, pp. 562–601.
- [51] A. ZANNA AND H. Z. MUNTKE-KAAS, *Generalized polar decompositions for the approximation of the matrix exponential*, SIAM J. Matrix Anal. Appl., 23 (2002), pp. 840–862.

EXTREMAL EIGENVALUES OF REAL SYMMETRIC MATRICES WITH ENTRIES IN AN INTERVAL*

XINGZHI ZHAN†

Abstract. We determine the exact range of the smallest and largest eigenvalues of real symmetric matrices of a given order whose entries are in a given interval. The maximizing and minimizing matrices are specified. We also consider the maximal spread of such matrices.

Key words. eigenvalue, symmetric matrix, spread

AMS subject classifications. 15A18, 15A42, 15A57

DOI. 10.1137/050627812

1. Introduction. Let $S_n[a, b]$ denote the set of $n \times n$ real symmetric matrices whose entries are in the interval $[a, b]$. For an $n \times n$ real symmetric matrix A , we always denote the eigenvalues of A in decreasing order by $\lambda_1(A) \geq \cdots \geq \lambda_n(A)$. We will study the smallest eigenvalue $\lambda_n(A)$ and the largest eigenvalue $\lambda_1(A)$ when A varies in $S_n[a, b]$. Constantine [2] proved that if $A \in S_n[0, b]$, then

$$\lambda_n(A) \geq \begin{cases} -nb/2 & \text{if } n \text{ is even,} \\ -\sqrt{n^2-1}b/2 & \text{if } n \text{ is odd.} \end{cases}$$

So the matrices treated there are nonnegative. The proof techniques of [2] are graph-theoretic. In [7] Roth gave another proof of this result by analysis of eigenvectors.

In this paper we will determine the smallest and largest values of both $\lambda_n(A)$ and $\lambda_1(A)$ when $A \in S_n[a, b]$ for generic $a < b$, thus generalizing Constantine's result.

The spread of an $n \times n$ real symmetric matrix A is defined as $s(A) = \lambda_1(A) - \lambda_n(A)$. This quantity has applications in combinatorial optimization problems [3]. Some lower bounds on the spread of Hermitian matrices are known; see [5] and the references therein. We will determine the maximal value of $s(A)$ for $A \in S_n[-a, a]$.

We always regard real vectors in $\mathbb{R}^n$ as $n \times 1$ matrices. A basic fact (see [1] or [4]) for an $n \times n$ real symmetric matrix A is

$$\lambda_n(A) = \min\{x^T A x : \|x\| = 1, x \in \mathbb{R}^n\}, \quad \lambda_1(A) = \max\{x^T A x : \|x\| = 1, x \in \mathbb{R}^n\}.$$

2. Extremal eigenvalues. We first consider the lower bound for the smallest eigenvalue. Denote by $J_{r,s}$ the $r \times s$ matrix with all entries equal to 1, and write J_r for $J_{r,r}$.

THEOREM 1. *Let $A \in S_n[a, b]$ with $n \geq 2$ and $a < b$.*

(i) *If $|a| < b$, then*

$$\lambda_n(A) \geq \begin{cases} n(a-b)/2 & \text{if } n \text{ is even,} \\ \left(na - \sqrt{a^2 + (n^2-1)b^2}\right)/2 & \text{if } n \text{ is odd.} \end{cases}$$

*Received by the editors March 27, 2005; accepted for publication (in revised form) by R. Bhatia July 6, 2005; published electronically January 27, 2006. This work was supported by NSFC grant 10571060.

<http://www.siam.org/journals/simax/27-3/62781.html>

†Department of Mathematics, East China Normal University, Shanghai 200062, China (zhan@math.ecnu.edu.cn).

If n is even, equality holds if and only if A is permutation similar to

$$\begin{bmatrix} a & b \\ b & a \end{bmatrix} \otimes J_{\frac{n}{2}}.$$

If n is odd, equality holds if and only if A is permutation similar to

$$\begin{bmatrix} aJ_{\frac{n-1}{2}} & bJ_{\frac{n-1}{2}, \frac{n+1}{2}} \\ bJ_{\frac{n+1}{2}, \frac{n-1}{2}} & aJ_{\frac{n+1}{2}} \end{bmatrix}.$$

(ii) If $|a| \geq b$, then $\lambda_n(A) \geq na$. If $|a| > b$, equality holds if and only if $A = aJ_n$. If $|a| = b$, equality holds if and only if A is permutation similar to

$$\begin{bmatrix} aJ_k & bJ_{k, n-k} \\ bJ_{n-k, k} & aJ_{n-k} \end{bmatrix}$$

for some k with $1 \leq k \leq n$.

Proof. For any fixed $A \in S_n[a, b]$, let $x = (x_1, \dots, x_n)^T$ be a unit (i.e., $\|x\| = 1$) eigenvector corresponding to $\lambda_n(A)$. By simultaneous permutations of the rows and columns of A if necessary, we may suppose that $x_i \geq 0$ for $i = 1, \dots, k$ and $x_j < 0$ for $j = k+1, \dots, n$, $1 \leq k \leq n$. We need not consider the case $k = 0$, as in that case we use $-x$ instead of x . Let $e \in \mathbb{R}^n$ be the vector with all entries equal to 1. Denote by $A \circ B$ the Hadamard product of A and B , i.e., the entrywise product. Then we may write $x^T Ax$ in a more visible form:

$$(1) \quad \lambda_n(A) = x^T Ax = e^T [A \circ (xx^T)] e.$$

Note that the matrix xx^T is divided into four blocks: The entries in $(xx^T)[1, \dots, k]$ and in $(xx^T)[k+1, \dots, n]$ are nonnegative, while the entries in $(xx^T)[1, \dots, k | k+1, \dots, n]$ and in $(xx^T)[k+1, \dots, n | 1, \dots, k]$ are nonpositive. Thus from (1) we see clearly that if we define

$$(2) \quad \tilde{A} = J(k; a, b) \equiv \begin{bmatrix} aJ_k & bJ_{k, n-k} \\ bJ_{n-k, k} & aJ_{n-k} \end{bmatrix},$$

then

$$\lambda_n(\tilde{A}) = \min\{y^T \tilde{A} y : \|y\| = 1, y \in \mathbb{R}^n\} \leq x^T \tilde{A} x \leq x^T A x = \lambda_n(A).$$

Therefore the smallest value of $\lambda_n(A)$ for $A \in S_n[a, b]$ can be attained at some matrix of the form in (2). Since the rank of $J(k; a, b)$ is at most 2, it has at most two nonzero eigenvalues. By considering the trace and the Frobenius norm we deduce that

$$(3) \quad \lambda_n(J(k; a, b)) = \left(na - \sqrt{(n-2k)^2 a^2 + 4k(n-k)b^2} \right) / 2.$$

(i) Since $|a| < b$, if n is even, the right side of (3) attains its minimum at $k = n/2$ and

$$\lambda_n(A) \geq \lambda_n(J(n/2; a, b)) = n(a-b)/2$$

for any $A \in S_n[a, b]$. If n is odd, the right side of (3) attains its minimum at $k = (n-1)/2$ and $k = (n+1)/2$. Hence

$$\lambda_n(A) \geq \lambda_n(J((n-1)/2; a, b)) = \left(na - \sqrt{a^2 + (n^2-1)b^2} \right) / 2.$$

Now we prove the equality conditions. First suppose n is even. Let $A = (a_{ij}) \in S_n[a, b]$ such that $\lambda_n(A) = n(a - b)/2$ and let $x = (x_1, \dots, x_n)^T$ be a corresponding unit eigenvector. Suppose x has exactly t nonzero components with $t < n$. By the above bounds, if t is even, then

$$\lambda_n(A) = x^T A x \geq t(a - b)/2 > n(a - b)/2,$$

and if t is odd, then

$$\lambda_n(A) = x^T A x \geq \left(ta - \sqrt{(a^2 - b^2) + t^2 b^2} \right) / 2 > t(a - b)/2 > n(a - b)/2,$$

both contradicting the assumption that $\lambda_n(A) = n(a - b)/2$. Therefore all the components of x are nonzero. Suppose x has k positive components and $n - k$ negative components. From (3) we know that if $k \neq n/2$,

$$\lambda_n(A) \geq \lambda_n(J(k; a, b)) > \lambda_n(J(n/2; a, b)) = n(a - b)/2,$$

a contradiction. Thus we must have $k = n/2$. By simultaneous row and column permutations of A if necessary, we may suppose that $x_i > 0$ for $i = 1, \dots, n/2$ and $x_j < 0$ for $j = (n/2) + 1, \dots, n$. Then

$$\lambda_n(A) = x^T A x \geq x^T J(n/2; a, b) x \geq \lambda_n(A)$$

forces $A = J(n/2; a, b)$, since otherwise the first inequality above will be strict, which is impossible. Therefore the original A is permutation similar to $J(n/2; a, b)$.

The equality condition for the case when n is odd can be similarly proved. Just note that

$$t(a - b) > na - \sqrt{a^2 + (n^2 - 1)b^2}$$

for $1 \leq t \leq n - 1$, the lower bound $(na - \sqrt{a^2 + (n^2 - 1)b^2})/2$ is strictly decreasing in n , and

$$\begin{bmatrix} aJ_{\frac{n+1}{2}} & bJ_{\frac{n+1}{2}, \frac{n-1}{2}} \\ bJ_{\frac{n-1}{2}, \frac{n+1}{2}} & aJ_{\frac{n-1}{2}} \end{bmatrix} \quad \text{and} \quad \begin{bmatrix} aJ_{\frac{n-1}{2}} & bJ_{\frac{n-1}{2}, \frac{n+1}{2}} \\ bJ_{\frac{n+1}{2}, \frac{n-1}{2}} & aJ_{\frac{n+1}{2}} \end{bmatrix}$$

are permutation similar.

(ii) $|a| \geq b$ and $a < b$ imply $a < 0$. If $|a| > b$, the minimum of the right side of (3) is attained at $k = n$, and if $|a| = b$, the right side of (3) has equal values for all k with $1 \leq k \leq n$. In any case the minimum is na . This proves $\lambda_n(A) \geq na$. The proof of the equality conditions is similar to that of case (i) and we omit the details. $\square$

Since $A \in S_n[a, b]$ implies $A - aJ_n \in S_n[0, b - a]$, it is natural to ask whether Theorem 1 can be deduced from Constantine's result by using the perturbation inequality: $\lambda_n(G + H) \geq \lambda_n(G) + \lambda_n(H)$ for $n \times n$ real symmetric matrices G, H [1]. In general the answer is no. Let us examine the case $0 < a < b$. If n is odd, the perturbation inequality and Constantine's result give

$$\begin{aligned} \lambda_n(A) &= \lambda_n[(A - aJ_n) + aJ_n] \\ &\geq \lambda_n(A - aJ_n) + \lambda_n(aJ_n) \\ &= \lambda_n(A - aJ_n) \\ &\geq \sqrt{n^2 - 1}(a - b)/2. \end{aligned}$$

It is easy to verify that this lower bound $\sqrt{n^2-1}(a-b)/2$ is strictly less than the sharp bound $(na - \sqrt{a^2 + (n^2-1)b^2})/2$ in Theorem 1. On the other hand, if n is even, the lower bound $n(a-b)/2$ can indeed be deduced from Constantine's result.

For a real $n \times n$ symmetric matrix A , $\lambda_1(A) = -\lambda_n(-A)$. Also $a \leq a_{ij} \leq b$ is equivalent to $-b \leq -a_{ij} \leq -a$. Thus the following corollary on upper bounds for the largest eigenvalue follows from Theorem 1.

COROLLARY 2. *Let $A \in S_n[a, b]$ with $n \geq 2$ and $a < b$.*

(i) *If $a < -|b|$, then*

$$\lambda_1(A) \leq \begin{cases} n(b-a)/2 & \text{if } n \text{ is even,} \\ (nb + \sqrt{b^2 + (n^2-1)a^2})/2 & \text{if } n \text{ is odd.} \end{cases}$$

If n is even, equality holds if and only if A is permutation similar to

$$\begin{bmatrix} b & a \\ a & b \end{bmatrix} \otimes J_{\frac{n}{2}}.$$

If n is odd, equality holds if and only if A is permutation similar to

$$\begin{bmatrix} bJ_{\frac{n-1}{2}} & aJ_{\frac{n-1}{2}, \frac{n+1}{2}} \\ aJ_{\frac{n+1}{2}, \frac{n-1}{2}} & bJ_{\frac{n+1}{2}} \end{bmatrix}.$$

(ii) *If $a \geq -|b|$, then $\lambda_1(A) \leq nb$. If $a > -|b|$, equality holds if and only if $A = bJ_n$. If $a = -|b|$, equality holds if and only if A is permutation similar to*

$$\begin{bmatrix} bJ_k & aJ_{k, n-k} \\ aJ_{n-k, k} & bJ_{n-k} \end{bmatrix}$$

for some k with $1 \leq k \leq n$.

Now we turn to the study of upper bounds on the smallest eigenvalue and lower bounds on the largest eigenvalue. For real matrices A, B , we write $A \leq B$ to mean that $B - A$ is entrywise nonnegative. We need the following two lemmas.

LEMMA 3 (see [1] or [4]). *Let H be a real symmetric matrix of order n and G be a principal submatrix of order k of H . Then*

$$\lambda_j(H) \geq \lambda_j(G) \geq \lambda_{j+n-k}(H)$$

for $j = 1, \dots, k$.

LEMMA 4 (see [6, p. 38]). *Let A, B be nonnegative matrices of the same order satisfying $A \leq B$. Then $\rho(A) \leq \rho(B)$, where $\rho(\cdot)$ is the Perron root (spectral radius). If, in addition, $A \neq B$ and B is irreducible, then $\rho(A) < \rho(B)$.*

Since $\lambda_n(A) = -\lambda_1(-A)$ and $\lambda_1(A) = -\lambda_n(-A)$ for $n \times n$ real symmetric matrices A , for our problem there are essentially two different cases: $0 < a < b$ and $a \leq 0 < b$. Denote by J and I the $n \times n$ all-one matrix and the $n \times n$ identity matrix, respectively.

THEOREM 5. *Let $A \in S_n[a, b]$ with $n \geq 2$ and $a < b$.*

(i) *Let $0 < a < b$. Then*

$$(4) \quad \lambda_n(A) \leq b - a.$$

Equality in (4) holds if and only if $A = aJ + (b-a)I$.

$$(5) \quad \lambda_1(A) \geq na.$$

Equality in (5) holds if and only if $A = aJ$.

(ii) Let $a \leq 0 < b$. Then

$$(6) \quad \lambda_n(A) \leq b.$$

Equality in (6) holds if and only if $A = bI$.

$$(7) \quad \lambda_1(A) \geq a.$$

Equality in (7) holds if and only if $A = aI$.

Proof. (i) Let $A = (a_{ij})$. For $i < j$, by Lemma 3 we have

$$(8) \quad \begin{aligned} \lambda_n(A) &\leq \lambda_2 \left(\begin{bmatrix} a_{ii} & a_{ij} \\ a_{ji} & a_{jj} \end{bmatrix} \right) = \frac{a_{ii} + a_{jj} - \sqrt{(a_{ii} - a_{jj})^2 + 4a_{ij}^2}}{2} \\ &\leq \frac{a_{ii} + a_{jj} - 2a_{ij}}{2} \leq b - a. \end{aligned}$$

Thus $\lambda_n(A) \leq b - a$. If $\lambda_n(A) = b - a$, all the inequalities in (8) must be equality. This forces $a_{ii} = a_{jj} = b$ and $a_{ij} = a_{ji} = a$. As this should be true for all $i < j$, $A = aJ + (b - a)I$.

$A \in S_n[a, b]$ implies $A \geq aJ \geq 0$. By Lemma 4 and the Perron–Frobenius theory [6],

$$\lambda_1(A) = \rho(A) \geq \rho(aJ) = \lambda_1(aJ) = na.$$

If $A \in S_n[a, b]$ and $A \neq aJ$, then since A is irreducible (A is in fact entrywise positive), again by Lemma 4, $\lambda_1(A) > \lambda_1(aJ) = na$. Thus $\lambda_1(A) = na$ if and only if $A = aJ$.

(ii)

$$\lambda_n(A) \leq \frac{\text{tr} A}{n} \leq \frac{nb}{n} = b.$$

If $\lambda_n(A) = b$, then $\text{tr} A = nb$ and consequently $a_{ii} = b$, $i = 1, \dots, n$. For any $i < j$, by Lemma 3 we have

$$b = \lambda_n(A) \leq \lambda_2 \left(\begin{bmatrix} b & a_{ij} \\ a_{ji} & b \end{bmatrix} \right) = b - |a_{ij}|.$$

Thus $a_{ij} = 0$ for all $i < j$, i.e., $A = bI$.

$$\lambda_1(A) \geq \frac{\text{tr} A}{n} \geq \frac{na}{n} = a.$$

If $\lambda_1(A) = a$, then $\text{tr} A = na$ and hence $a_{ii} = a$, $i = 1, \dots, n$. For any $i < j$, by Lemma 3 we have

$$a = \lambda_1(A) \geq \lambda_1 \left(\begin{bmatrix} a & a_{ij} \\ a_{ji} & a \end{bmatrix} \right) = a + |a_{ij}|.$$

So $a_{ij} = 0$ for all $i < j$, i.e., $A = aI$. This completes the proof. $\square$

3. The maximal spread. Denote by $s(A)$ the spread of A . We treat only the case when the interval is symmetric about the origin. Of course we may use the upper bound on λ_1 in Corollary 2 and the lower bound on λ_n in Theorem 1 to give an upper bound on the spread $\lambda_1 - \lambda_n$, but that bound is not sharp. This is because the upper bound on λ_1 and the lower bound on λ_n cannot be simultaneously attained at one common matrix.

A $\{\pm 1\}$ -matrix is a matrix whose entries are either 1 or -1 . Two matrices A, B of the same order are said to be *D-similar* if there is a diagonal matrix D with diagonal entries equal to 1 or -1 such that $DAD = B$. We will need the following lemma.

LEMMA 6. *Let A be an $n \times n$ symmetric $\{\pm 1\}$ -matrix with all diagonal entries equal to 1. Then either A is D-similar to J_n or A has a principal submatrix of order 3 which is similar to*

$$B_1 = \begin{bmatrix} 1 & -1 & 1 \\ -1 & 1 & 1 \\ 1 & 1 & 1 \end{bmatrix}.$$

Proof. The cases $n = 1, 2$ are obvious. Use induction on n for $n \geq 3$. First let $n = 3$. If $A = J_3$, there is nothing to prove. Otherwise A has an off-diagonal entry equal to -1 . Then there are the following possibilities of A :

$$B_1, \begin{bmatrix} 1 & 1 & -1 \\ 1 & 1 & 1 \\ -1 & 1 & 1 \end{bmatrix}, \begin{bmatrix} 1 & 1 & 1 \\ 1 & 1 & -1 \\ 1 & -1 & 1 \end{bmatrix}, \begin{bmatrix} 1 & -1 & -1 \\ -1 & 1 & 1 \\ -1 & 1 & 1 \end{bmatrix},$$

$$\begin{bmatrix} 1 & 1 & -1 \\ 1 & 1 & -1 \\ -1 & -1 & 1 \end{bmatrix}, \begin{bmatrix} 1 & -1 & 1 \\ -1 & 1 & -1 \\ 1 & -1 & 1 \end{bmatrix}, \begin{bmatrix} 1 & -1 & -1 \\ -1 & 1 & -1 \\ -1 & -1 & 1 \end{bmatrix}.$$

It is easy to check that each of these matrices satisfies the conclusion. Now consider $n \geq 4$ and suppose the lemma holds for matrices of order $n - 1$. Let

$$A = \begin{bmatrix} 1 & u^T \\ u & A_1 \end{bmatrix}.$$

If A_1 has a principal submatrix of order 3 which is similar to B_1 , then so does A . Otherwise, by the assumption, A_1 is D-similar to J_{n-1} . So A is D-similar to

$$H = \begin{bmatrix} 1 & v^T \\ v & J_{n-1} \end{bmatrix} = (h_{ij}).$$

If each entry of v is 1 or each entry of v is -1 , then H and hence A are D-similar to J_n . Otherwise we have $h_{1,p} = -1$, $h_{1,q} = 1$ or $h_{1,p} = 1$, $h_{1,q} = -1$ for some $1 < p < q$. In the first case $H[1, p, q] = B_1$. In the second case $H[1, p, q]$ is D-similar to B_1 . Therefore in both cases A has a principal submatrix which is similar to B_1 . $\square$

Two matrices A, B of the same order are said to be *sign-permutation similar* if there exist a permutation matrix P and a diagonal matrix D with diagonal entries equal to 1 or -1 such that $DP^TAPD = B$. It is clear that sign-permutation similarity is an equivalence relation.

THEOREM 7. *Let $A \in S_n[-a, a]$ with $n \geq 2$ and $a > 0$. Then*

$$s(A) \leq \begin{cases} \sqrt{2na} & \text{if } n \text{ is even,} \\ \sqrt{2n^2 - 1a} & \text{if } n \text{ is odd.} \end{cases}$$

If n is even, equality holds if and only if A is sign-permutation similar to

$$a \begin{bmatrix} 1 & 1 \\ 1 & -1 \end{bmatrix} \otimes J_{\frac{n}{2}}.$$

If n is odd, equality holds if and only if A is sign-permutation similar to

$$\pm a \begin{bmatrix} J_{\frac{n+1}{2}} & J_{\frac{n+1}{2}, \frac{n-1}{2}} \\ J_{\frac{n-1}{2}, \frac{n+1}{2}} & -J_{\frac{n-1}{2}} \end{bmatrix}.$$

Proof. By considering $a^{-1}A$ instead of A , it suffices to prove the theorem for the case $a = 1$. Suppose $A \in S_n[-1, 1]$. Throughout this proof we write λ_j for $\lambda_j(A)$. For $x \in \mathbb{R}^n$, we always write its components as $x_1, \dots, x_n$. Given $A \in S_n[-1, 1]$, let $x, y \in \mathbb{R}^n$ be unit eigenvectors such that $\lambda_1 = x^T A x$, $\lambda_n = y^T A y$. Then

$$(9) \quad s(A) = x^T A x - y^T A y = e^T [A \circ (xx^T - yy^T)] e.$$

Note that the (i, j) entry of $xx^T - yy^T$ is $x_i x_j - y_i y_j$. We define a new matrix $\tilde{A} = (\tilde{a}_{ij})$ as $\tilde{a}_{ij} = 1$ if $x_i x_j - y_i y_j \geq 0$ and $\tilde{a}_{ij} = -1$ if $x_i x_j - y_i y_j < 0$. Then from (9) we have

$$\begin{aligned} s(A) &\leq x^T \tilde{A} x - y^T \tilde{A} y \\ &\leq \max\{w^T \tilde{A} w : \|w\| = 1, w \in \mathbb{R}^n\} - \min\{z^T \tilde{A} z : \|z\| = 1, z \in \mathbb{R}^n\} \\ &= s(\tilde{A}). \end{aligned}$$

Therefore the maximal spread can always be attained at some $\{\pm 1\}$ -matrix. If $n = 2$, the conclusions of the theorem are easily checked to be true. Next we assume $n \geq 3$. Now suppose A is a $\{\pm 1\}$ -matrix of order n .

The following three matrices will play a role in our proof:

$$B_1 = \begin{bmatrix} 1 & -1 & 1 \\ -1 & 1 & 1 \\ 1 & 1 & 1 \end{bmatrix}, \quad B_2 = \begin{bmatrix} 1 & 1 & 1 \\ 1 & 1 & -1 \\ 1 & -1 & -1 \end{bmatrix}, \quad B_3 = \begin{bmatrix} 1 & 1 & 1 \\ 1 & -1 & 1 \\ 1 & 1 & -1 \end{bmatrix}.$$

Their eigenvalues are

$$\lambda(B_1) = \{2, 2, -1\}, \quad \lambda(B_2) = \{2, 1, -2\}, \quad \lambda(B_3) = \{2, -1, -2\}.$$

If A has a principal submatrix of order 3 which is similar to B_1 , then by Lemma 3 $\lambda_2 \geq 2$. If A has a principal submatrix of order 3 which is similar to B_2 , then by Lemma 3 $\lambda_2 \geq 1$. If A has a principal submatrix of order 3 which is similar to B_3 , then by Lemma 3 $\lambda_{n-1} \leq -1$. In all three cases we have

$$\lambda_1^2 + \lambda_n^2 = \|A\|_F^2 - \sum_{j=2}^{n-1} \lambda_j^2 \leq n^2 - 1.$$

Hence

$$(10) \quad s(A) = \lambda_1 - \lambda_n \leq \sqrt{2(\lambda_1^2 + \lambda_n^2)} \leq \sqrt{2(n^2 - 1)} < \min\{\sqrt{2}n, \sqrt{2n^2 - 1}\}.$$

Thus if one of the above cases occurs, the spread is less than our claimed upper bound.

If all the diagonal entries of A are 1, then by Lemma 6 either A is D-similar to J_n or A has a principal submatrix of order 3 which is similar to B_1 . Since $s(J_n) = n$, in both cases $s(A)$ is not the maximal value. If all the diagonal entries of A are -1 , then since $s(-A) = s(A)$, this case is the same as what we just discussed.

Next consider those $\{\pm 1\}$ -matrices A whose diagonal contains both 1 and -1 . By simultaneous row and column permutations if necessary, we may suppose

$$A = \begin{bmatrix} A_r & A_{r,s} \\ A_{r,s}^T & A_s \end{bmatrix},$$

where A_r is of order r ($1 \leq r \leq n-1$) and A_r 's diagonal entries are all 1, and A_s is of order $s = n-r$ and A_s 's diagonal entries are all -1 . Since $s(-A) = s(A)$, we need consider only the case $r \geq s$. Then $r \geq 2$. By Lemma 6, either A_r is D-similar to J_r or A_r has a principal submatrix of order 3 which is similar to B_1 . In the first case A is D-similar to a matrix whose r th leading principal submatrix is J_r , while in the second case $s(A)$ is not the maximal value. Thus we may suppose $A_r = J_r$. Now two cases can occur. (i) $A_{r,s}$ has a column which contains both 1 and -1 . Then A has a principal submatrix which is similar to B_2 , and $s(A)$ is not the maximal value. (ii) Each column of $A_{r,s}$ contains only 1 or only -1 . Then A is D-similar to

$$G = \begin{bmatrix} J_r & J_{r,s} \\ J_{s,r} & \tilde{A}_s \end{bmatrix},$$

where all the diagonal entries of $\tilde{A}_s$ remain -1 . If $\tilde{A}_s$ has an off-diagonal entry equal to 1, then G has B_3 as a principal submatrix and $s(G)$ is not the maximal value. Thus we further consider the case $\tilde{A}_s = -J_s$.

Now there remains the case

$$(11) \quad A = \begin{bmatrix} J_r & J_{r,s} \\ J_{s,r} & -J_s \end{bmatrix},$$

where $r \geq s$. It is easy to see that

$$(12) \quad s \left(\begin{bmatrix} J_r & J_{r,s} \\ J_{s,r} & -J_s \end{bmatrix} \right) = \sqrt{2n^2 - (2r-n)^2}.$$

Therefore by (12) the maximal spread of the matrices in (11) is $\sqrt{2}n$ attained uniquely at $r = n/2$ if n is even, and the maximal spread is $\sqrt{2n^2 - 1}$ attained uniquely at $r = (n+1)/2$ if n is odd. For the odd case note that we have assumed $r \geq s = n-r$, and hence $r = (n-1)/2$ does not occur. At this stage we have found the maximal spread of $A \in S_n[-1, 1]$.

Next we determine those matrices which attain the maximal spread. Suppose $A \in S_n[-1, 1]$ attains the maximal spread and x, y are the unit eigenvectors corresponding to λ_1 and λ_n , respectively. Assume that $xx^T - yy^T$ has a zero entry, say, $x_i x_j - y_i y_j = 0$ for some i, j . From (9) it is clear that we can change the corresponding entry a_{ij} of A arbitrarily without affecting the value of $s(A)$. So we may suppose $a_{ij} = 0$. Then

$$n^2 - 1 \geq \|A\|_F^2 \geq \lambda_1^2 + \lambda_n^2 \geq \frac{(\lambda_1 - \lambda_n)^2}{2} = \begin{cases} n^2 & \text{if } n \text{ is even,} \\ n^2 - \frac{1}{2} & \text{if } n \text{ is odd,} \end{cases}$$

which is a contradiction. Thus every entry of $xx^T - yy^T$ is nonzero. By (9) we deduce that A must be a $\{\pm 1\}$ -matrix. On the other hand, the above analysis leading

to the maximal spread shows that when n is even, $s(A) = \sqrt{2}n$ if and only if A is sign-permutation similar to

$$\begin{bmatrix} 1 & 1 \\ 1 & -1 \end{bmatrix} \otimes J_{\frac{n}{2}},$$

and when n is odd, $s(A) = \sqrt{2n^2 - 1}$ if and only if A is sign-permutation similar to

$$\pm \begin{bmatrix} J_{\frac{n+1}{2}} & J_{\frac{n+1}{2}, \frac{n-1}{2}} \\ J_{\frac{n-1}{2}, \frac{n+1}{2}} & -J_{\frac{n-1}{2}} \end{bmatrix}.$$

The permutation similarity comes from our operation to put positive diagonal entries together and let them appear first. The possible minus sign comes from the fact that $s(-A) = s(A)$, which we used to simplify our analysis. Note also that in the case when n is even,

$$\begin{bmatrix} 1 & 1 \\ 1 & -1 \end{bmatrix} \otimes J_{\frac{n}{2}} \quad \text{and} \quad - \begin{bmatrix} 1 & 1 \\ 1 & -1 \end{bmatrix} \otimes J_{\frac{n}{2}}$$

are sign-permutation similar, and the minus sign need not appear in our assertion for the equality case. This completes the proof. $\square$

Let $e_t \in \mathbb{R}^t$ denote the all-one vector. By Theorem 7 and (9) we get the following interesting corollary.

COROLLARY 8.

$$\max \left\{ \sum_{i,j=1}^n |x_i x_j - y_i y_j| : \|x\| = \|y\| = 1, x, y \in \mathbb{R}^n \right\} = \begin{cases} \sqrt{2}n & \text{if } n \text{ is even,} \\ \sqrt{2n^2 - 1} & \text{if } n \text{ is odd.} \end{cases}$$

The maximum is attained at x, y if and only if $x = DPx_0$, $y = DPy_0$ for some diagonal matrix D with diagonal entries equal to 1 or -1 and some permutation matrix P where

$$x_0 = \begin{pmatrix} ae_{n/2} \\ be_{n/2} \end{pmatrix}, \quad y_0 = \begin{pmatrix} -be_{n/2} \\ ae_{n/2} \end{pmatrix}, \quad a = (1 + \sqrt{2})b, \quad b = \frac{1}{\sqrt{(2 + \sqrt{2})n}}$$

if n is even and

$$x_0 = \begin{pmatrix} ae_{(n+1)/2} \\ be_{(n-1)/2} \end{pmatrix}, \quad a = \sqrt{\frac{1}{n+1} \left(1 + \frac{n}{\sqrt{2n^2 - 1}} \right)}, \quad b = \sqrt{\frac{1}{n-1} \left(1 - \frac{n}{\sqrt{2n^2 - 1}} \right)}$$

$$y_0 = \begin{pmatrix} ce_{(n+1)/2} \\ de_{(n-1)/2} \end{pmatrix}, \quad c = \sqrt{\frac{1}{n+1} \left(1 - \frac{n}{\sqrt{2n^2 - 1}} \right)}, \quad d = -\sqrt{\frac{1}{n-1} \left(1 + \frac{n}{\sqrt{2n^2 - 1}} \right)}$$

if n is odd.

We remark that the above x_0 and y_0 are the unit eigenvectors corresponding to the largest and smallest eigenvalues of the maximizing matrix in Theorem 7.

We have the following two obvious problems which are not solved here.

Problem 1. For a given integer j with $2 \leq j \leq n-1$, determine

$$\max\{\lambda_j(A) : A \in S_n[a, b]\},$$

$$\min\{\lambda_j(A) : A \in S_n[a, b]\}$$

and determine which matrices attain the maximum and which matrices attain the minimum.

Problem 2. For generic $a < b$, determine

$$\max\{s(A) : A \in S_n[a, b]\},$$

where $s(A)$ denotes the spread of A , and determine which matrices attain the maximum.

REFERENCES

- [1] R. BHATIA, *Matrix Analysis*, Springer-Verlag, New York, 1997.
- [2] G. CONSTANTINE, *Lower bounds on the spectra of symmetric matrices with nonnegative entries*, Linear Algebra Appl., 65 (1985), pp. 171–178.
- [3] G. FINKE, R. E. BURKARD, AND F. RENDL, *Quadratic assignment problems*, in Surveys of Combinatorial Optimization, North-Holland, Amsterdam, 1987, pp. 61–82.
- [4] R. A. HORN AND C. R. JOHNSON, *Matrix Analysis*, Cambridge University Press, Cambridge, UK, 1985.
- [5] E. JIANG AND X. ZHAN, *Lower bounds for the spread of a Hermitian matrix*, Linear Algebra Appl., 256 (1997), pp. 153–163.
- [6] H. MINC, *Nonnegative Matrices*, John Wiley & Sons, New York, 1988.
- [7] R. ROTH, *On the eigenvectors belonging to the minimum eigenvalue of an essentially nonnegative symmetric matrix with bipartite graph*, Linear Algebra Appl., 118 (1989), pp. 1–10.

CORE PROBLEMS IN LINEAR ALGEBRAIC SYSTEMS*

CHRISTOPHER C. PAIGE[†] AND ZDENĚK STRAKOŠ[‡]

Abstract. For any linear system $Ax \approx b$ we define a set of core problems and show that the orthogonal upper bidiagonalization of $[b, A]$ gives such a core problem. In particular we show that these core problems have desirable properties such as minimal dimensions. When a total least squares problem is solved by first finding a core problem, we show the resulting theory is consistent with earlier generalizations, but much simpler and clearer. The approach is important for other related solutions and leads, for example, to an elegant solution to the data least squares problem. The ideas could be useful for solving ill-posed problems.

Key words. scaled total least squares, least squares, data least squares, orthogonal regression, core problem, orthogonal reduction, minimum 2-norm solutions, bidiagonalization, singular value decomposition, ill-posed problems

AMS subject classifications. 15A06, 15A18, 15A21, 65F20, 65F25, 65G50

DOI. 10.1137/040616991

1. Introduction. We will use uppercase Roman letters to denote matrices, lowercase Roman to denote vectors and indices, and lowercase Greek to denote scalars. The i th column of the unit matrix I is e_i , $\|\cdot\|$ denotes the 2-norm, $\|\cdot\|_F$ denotes the Frobenius norm, and $\mathcal{R}(M)$ denotes the range (column space) of a matrix M .

Consider estimating $\tilde{x}$ from the (possibly compatible) real linear system

$$(1.1) \quad \tilde{A}\tilde{x} \approx \tilde{b}, \quad \tilde{A} \text{ a nonzero } n \text{ by } k \text{ matrix,} \quad \tilde{b} \text{ a nonzero } n\text{-vector.}$$

Suppose (1.1) can be transformed to $Ax \equiv (P^T \tilde{A} Q)(Q^T \tilde{x}) \approx P^T \tilde{b} \equiv b$, where

$$(1.2) \quad P^T \begin{bmatrix} \tilde{b} \\ \tilde{A} Q \end{bmatrix} = \begin{bmatrix} b \\ A \end{bmatrix} = \left[\begin{array}{c|c} b_1 & A_{11} \\ \hline 0 & 0 \end{array} \middle| \begin{array}{c} 0 \\ A_{22} \end{array} \right]; \quad P^{-1} = P^T, \quad Q^{-1} = Q^T.$$

We say this is a nontrivial decomposition if A_{22} has at least one row and one column, even if $A_{22} = 0$. In this nontrivial case the singular value decompositions (SVDs) of $[b, A]$ and A can each be split into two independent SVDs, with the SVD of A_{22} being common to both. More importantly for this exposition, the approximation problem $\tilde{A}\tilde{x} \approx \tilde{b}$ can then be transformed to two *independent* approximation problems as follows:

$$(1.3) \quad A_{11}x_1 \approx b_1, \quad A_{22}x_2 \approx 0, \quad \tilde{x} \equiv Qx, \quad x \equiv \begin{bmatrix} x_1 \\ x_2 \end{bmatrix}.$$

It can be seen from (1.2) that the solution to each of the approximation problems in (1.3) does not affect, and can be found independently of, the other. The problem $A_{22}x_2 \approx 0$ says only that x_2 lies approximately in the nullspace of A_{22} . Thus unless

*Received by the editors October 4, 2004; accepted for publication (in revised form) by P. C. Hansen July 7, 2005; published electronically January 27, 2006.

<http://www.siam.org/journals/simax/27-3/61699.html>

[†]School of Computer Science, McGill University, Montreal, Quebec, H3A 2A7, Canada (paige@cs.mcgill.ca). The research of this author was supported by NSERC of Canada grant OGP0009236.

[‡]Institute of Computer Science, Academy of Sciences of the Czech Republic, Pod Vodárenskou věží 2, 182 07 Praha 8, Czech Republic (strakos@cs.cas.cz). The research of this author was supported by the National Program of Research “Information Society” under project 1ET400300415.

there is some reason not to (see, for example, Remark 1.1), we can take $x_2 = 0$, and only $A_{11}x_1 \approx b_1$ need be solved. For orthogonally transformed problems we suggest the following definition.

DEFINITION 1.1. *We say $A_{11}x_1 \approx b_1$ is a core problem in $\tilde{A}\tilde{x} \approx \tilde{b}$ if $[b_1, A_{11}]$ is minimally dimensioned (or A_{22} is maximally dimensioned) subject to (1.2).*

The ideas presented here may be applicable to other than orthogonal transformations of (1.1), but we will concentrate on orthogonal transformations because of their practicality (their relationship with scaled total least squares (scaled TLS)) and the elegance of the results (their relationship with the SVD). Even when we restrict ourselves to orthogonal transformations, the concept of core problems is meaningful outside optimal approximation methods, but these give the best motivation for introducing core problems. For (1.1), the unifying example of optimal approximation problems whose optima are invariant under orthogonal transformations is then scaled TLS: for given $\gamma > 0$:

$$(1.4) \quad \text{scaled TLS distance} \equiv \min_{\tilde{g}, \tilde{E}, \tilde{x}} \|\tilde{g}, \tilde{E}\|_F \quad \text{subject to} \quad (\tilde{A} + \tilde{E})\tilde{x}\gamma = \tilde{b}\gamma + \tilde{g}.$$

This is a reformulation of the unification in [22]; see [20, 21]. When $\gamma = 1$ scaled TLS becomes total least squares (TLS, see in particular [10], [7, section 6], [11], [9, pp. 324–326], [25, 23]). TLS is also known in the statistical literature as orthogonal regression. In the limit scaled TLS corresponds to ordinary least squares (LS) when $\gamma \rightarrow 0$, and to data least squares (DLS, see [13]) when $\gamma \rightarrow \infty$; see [11, 22, 20, 21].

It is not theoretically necessary, but for the remainder of this paper we will assume

$$(1.5) \quad \tilde{b} \notin \mathcal{R}(\tilde{A}) \quad (\text{that is, } \tilde{A}^T \tilde{b} \neq 0).$$

This eliminates the annoying trivial case where A_{11} in (1.2) has no columns.

It is often important, even essential, to find a *core* problem $A_{11}x_1 \approx b_1$. It was shown in [20, section 7] that the existence of a nontrivial decomposition of the form (1.2) can prevent the TLS, scaled TLS, and DLS formulations for solving (1.1) from having meaningful solutions when applied directly to $[\tilde{b}, \tilde{A}]$. Here we give a simple example to make this obvious.

For our analysis it is sufficient and convenient to consider the TLS problem ((1.4) with $\gamma = 1$) applied to $\tilde{A}\tilde{x} \approx \tilde{b}$ with nontrivial decomposition (1.2). If $A_{11}x_1 \approx b_1$ is a core problem, then its TLS solution exists and is unique (see Remark 2.1), and its TLS distance is $\sigma_{\min}([b_1, A_{11}])$, where $\sigma_{\min}(\cdot)$ denotes the minimum singular value; see, for example, [12, section 12.3]. Using temporary notation, suppose $\sigma_k \equiv \sigma_{\min}(A_{22})$ and

$$\sigma_k < \sigma_{\min}([b_1, A_{11}]), \quad A_{22}v_k = u_k\sigma_k, \quad u_k^T A_{22} = \sigma_k v_k^T, \quad \|u_k\| = \|v_k\| = 1.$$

For any real vector z define $r_1 \equiv b_1 - A_{11}z$, then for any real scalar $\theta > 0$

$$\left[\begin{array}{c|c} A_{11} & r_1\theta^{-1}v_k^T \\ \hline 0 & A_{22} - u_k\sigma_kv_k^T \end{array} \right] \left[\begin{array}{c} z \\ v_k\theta \end{array} \right] = \left[\begin{array}{c} b_1 \\ 0 \end{array} \right],$$

so the square of the Frobenius norm of the corresponding correction to A is $\|r_1\|^2\theta^{-2} + \sigma_k^2 \rightarrow \sigma_k^2$ as $\theta \rightarrow \infty$.

Thus by applying (1.4) directly to $[\tilde{b}, \tilde{A}]$ we can get a nonoptimal “TLS distance” less than $\sigma_{\min}([b_1, A_{11}])$, the TLS distance for the meaningful core problem $A_{11}x_1 \approx b_1$. The above “solution vector” need have nothing to do with the TLS solution

vector for $A_{11}x_1 \approx b_1$, but is *essentially determined by v_k from the noncore part of the problem $A_{22}x_2 \approx 0$* . This does not reflect any useful information contained in the data. Moreover, in this case the optimal solution to (1.4) does not even exist. Instead, in general we recommend finding a core problem via (1.2); then

$$(1.6) \quad \text{solve the core problem } A_{11}x_1 \approx b_1 \text{ and set } x_2 = 0 \text{ in (1.3).}$$

Section 2 introduces an SVD based decomposition of the form (1.2) with minimally dimensioned $[b_1, A_{11}]$. It also suggests a possible application to ill-posed problems with uncertain data. Section 3 shows how an orthogonal bidiagonalization will give an optimally partitioned decomposition of the form (1.2) directly. This bidiagonalization will also show whether the original problem (1.1) is compatible or not, and is an ideal first step in obtaining the TLS, scaled TLS, or DLS solutions to (1.1); see [20, 21]. For completeness this is briefly reviewed in section 4, with an emphasize on an elegant solution of the following DLS formulation applied to a core problem:

$$(1.7) \quad \text{DLS distance} \equiv \min_{E_{11}, x_1} \|E_{11}\|_F \quad \text{subject to} \quad (A_{11} + E_{11})x_1 = b_1.$$

Section 5 summarizes our ideas and compares them with other approaches. In all cases the extension to the complex case is straightforward; see, for example, [20]. Throughout the paper we will use the following easily proven result.

LEMMA 1.2. *For (1.1), and (1.3) satisfying (1.2), $A_{11}x_1 \approx b_1$ is a compatible (incompatible) system if and only if $\tilde{A}\tilde{x} \approx \tilde{b}$ is a compatible (incompatible) system.*

Remark 1.1. In some applications such as solving noisy ill-posed problems, one may be tempted to use a nonzero x_2 in order to enforce some regularization constraints on the solution $\tilde{x}$. Such constraints are typically formulated in terms of the generalized norm $\|L\tilde{x}\|$, which for a given matrix L can combine x_1 and x_2 ; see [16]. In [6, section 5] the problem involving $\tilde{A}$, L , and $\tilde{b}$ is transformed into a standard form TLS problem with the matrix A_{sf} and the right-hand side b_{sf} . Then standard algorithms for finding the regularized TLS solution of the transformed problem are applied, followed by the transformation of the regularized solution $\tilde{x}_{sf}$ back to the general setting. Within such a framework the core problem theory and computations can be applied to solving the transformed problem $A_{sf}x_{sf} \approx b_{sf}$.

Here we focus on the basic theory of core problems in (1.1). We are aware of the possible implementation difficulties, and of promising applications to regularization of ill-posed problems; see Remarks 2.2 and 3.2 below. These issues are, however, not in the scope of this paper, and we leave them for further investigation. Therefore, within the scope of our paper, the obvious choice for the solution of $A_{22}x_2 \approx 0$ is $x_2 = 0$.

2. Understanding core problems in $\tilde{A}\tilde{x} \approx \tilde{b}$. In Definition 1.1 we said $A_{11}x_1 \approx b_1$ is a core problem in $\tilde{A}\tilde{x} \approx \tilde{b}$ if an orthogonal decomposition of the form of (1.2) exists where $[b_1, A_{11}]$, and so A_{11} , has minimal dimensions. An understanding of such minimal dimensions can be gained by the following construction, which shows how to concentrate the relevant information into A_{11} and b_1 while moving the irrelevant and redundant information into A_{22} . Let $\tilde{A}$ have rank r and SVD

$$\tilde{A} = U \left[\begin{array}{c|c} S & 0 \\ \hline 0 & 0 \end{array} \right] V^T,$$

where $S = \text{diag}(\sigma_1, \dots, \sigma_r)$ with $\sigma_1 \geq \dots \geq \sigma_r > 0$, the $n \times n$ matrix $U^{-1} = U^T$, and the $k \times k$ matrix $V^{-1} = V^T$. Then

$$(2.1) \quad U^T [\tilde{b} \parallel \tilde{A}V] = \left[\begin{array}{c|c|c} \tilde{c} & S & 0 \\ d & 0 & 0 \end{array} \right].$$

Although singular values are unique, in any SVD representation their ordering, and sometimes some singular vectors, are not. In order to obtain the core problem we will seek to transform $U^T[\tilde{b}, \tilde{A}V]$ further, while maintaining the SVD of $\tilde{A}$. Consider the partitioning $U \equiv [U_1, U_2]$, $V \equiv [V_1, V_2]$, where U_1 and V_1 have r columns. The vectors $\tilde{c}$ and d might already have some zero elements, and our aim is to introduce as many more as possible. For d , choose orthogonal P_{22} so that $P_{22}^T d = e_1 \delta$, $\delta \equiv \|d\|$, and replace U_2 by $U_2 P_{22}$. In this way, d is transformed into a vector having at most one nonzero entry. Now consider $\tilde{c} \equiv [\tilde{\gamma}_1, \dots, \tilde{\gamma}_r]^T$. Suppose that some of the singular values of A have multiplicities greater than one, for example $\sigma_i = \sigma_{i+1} = \dots = \sigma_j$, $j > i$. Choose orthogonal $P_{(i,j)}$ so that $P_{(i,j)}^T [\tilde{\gamma}_i, \dots, \tilde{\gamma}_j]^T = [\gamma_{i,j}, 0, \dots, 0]^T$. Then transform U_1 and V_1 by application of $P_{(i,j)}$ from the right to the block of columns numbered $i, \dots, j$. This transformation will leave S unchanged and therefore preserves the SVD. Do this for each block of multiple singular values, and so obtain P_{11} where $P_{11}^T \tilde{c}$ has at most one nonzero element corresponding to each block of equal singular values of S , and replace U_1 by $U_1 P_{11}$, V_1 by $V_1 P_{11}$. Next permute the columns of $U_1 P_{11}$ and $V_1 P_{11}$ identically, to move the zero elements of $P_{11}^T \tilde{c}$ to the bottom of this vector, leaving c at the top with nonzero elements while keeping S diagonal. Finally, if $\delta > 0$, move its row so δ is immediately below c by a further permutation from the left to give, with obvious new notation and indexing,

$$(2.2) \quad U^T [\tilde{b} \parallel \tilde{A}V] = \left[\begin{array}{c|c|c} b_1 & A_{11} & 0 \\ 0 & 0 & A_{22} \end{array} \right] = \left[\begin{array}{c|c|c} c & S_1 & 0 \\ \delta & 0 & 0 \\ 0 & 0 & S_2 \end{array} \right],$$

where, assuming $\tilde{b} \notin \mathcal{R}(\tilde{A})$,

$$(2.3) \quad \begin{aligned} & c \equiv [\gamma_1, \dots, \gamma_p]^T; \quad \gamma_i \neq 0, \quad i = 1, \dots, p; \quad U^{-1} = U^T, \quad V^{-1} = V^T; \\ & S_1 \equiv \text{diag}(\sigma_1, \dots, \sigma_p), \quad \sigma_1 > \dots > \sigma_p > 0; \quad \text{the row with} \\ & \text{the scalar } \delta \text{ is nonexistent if and only if } \tilde{A}\tilde{x} \approx \tilde{b} \text{ is compatible.} \end{aligned}$$

The final partitioning corresponds to that in (1.2). Here S_2 has the remaining $r - p$ singular values of $\tilde{A}$, and the comment in (2.3) follows from Lemma 1.2. We emphasize that the diagonal elements of S_1 are different from each other and that all entries in c are nonzero. In this way, the redundant information (multiplicities of singular values) and irrelevant data are removed to S_2 .

We now show that $A_{11}x_1 \approx b_1$ obtained by the transformation process described above has the desired minimality property. For the SVD of $\tilde{A}$ write $U \equiv [u_1, \dots, u_n]$. If $u_i, u_{i+1}, \dots, u_j$, $i \leq j$, are *all* the left singular vectors corresponding to a given singular value σ , we say $\mathcal{R}([u_i, u_{i+1}, \dots, u_j])$ is the *left singular subspace* of $\tilde{A}$ corresponding to σ . But the left and right singular subspaces corresponding to a given (possibly multiple) nonzero singular value σ of $\tilde{A}$ are unique; see, for example, [17, Thm. 3.1.1', p. 147]. Thus if $\tilde{b}$ is orthogonal (or not orthogonal) to the left singular subspace of $\tilde{A}$ corresponding to a given singular value $\sigma > 0$, this will be obvious in *all* SVD representations of $\tilde{A}$. In particular for the SVD of $\tilde{A}$ in (2.2)–(2.3), for $i = 1, \dots, p$

we see that $u_i \gamma_i = u_i(u_i^T \tilde{b})$ is the projection of $\tilde{b}$ onto the left singular subspace of $\tilde{A}$ corresponding to $\sigma_i > 0$ (since the construction has ensured that the other singular vectors of any multiple σ_i are orthogonal to $\tilde{b}$). This construction of the decomposition (1.2) in the special form (2.2)–(2.3) leads to Lemma 2.1. The subsequent minimality theorem proves the minimal dimensions of the resulting core problem $A_{11}x_1 \approx b_1$.

LEMMA 2.1. *$[\tilde{b}, \tilde{A}]$ has a decomposition of the form in (2.2)–(2.3) if and only if $\tilde{b}$ has nonzero projections on exactly p left singular subspaces of $\tilde{A}$ corresponding to distinct nonzero singular values. The projections correspond to those in $[b_1, A_{11}]$.*

THEOREM 2.2. *Suppose $\tilde{b} \notin \mathcal{R}(\tilde{A})$ has nonzero projections on exactly p left singular subspaces of $\tilde{A}$ corresponding to distinct nonzero singular values. Then among all decompositions of the form (1.2), the minimally dimensioned A_{11} is $p \times p$ if $\tilde{A}\tilde{x} \approx \tilde{b}$ is compatible, and $(p+1) \times p$ if $\tilde{A}\tilde{x} \approx \tilde{b}$ is incompatible.*

Proof. From Lemmas 1.2 and 2.1, $[\tilde{b}, \tilde{A}]$ has a decomposition of the form in (2.2)–(2.3), where A_{11} is $p \times p$ if $\tilde{A}\tilde{x} \approx \tilde{b}$ is compatible and $(p+1) \times p$ if $\tilde{A}\tilde{x} \approx \tilde{b}$ is incompatible. Suppose that there exists another decomposition of the form

$$\bar{P}^T \left[\begin{array}{c|c} \tilde{b} & \tilde{A}\bar{Q} \end{array} \right] = \left[\begin{array}{c|c|c} \bar{b}_1 & \bar{A}_{11} & 0 \\ \hline 0 & 0 & \bar{A}_{22} \end{array} \right],$$

where $\bar{P}$ and $\bar{Q}$ are orthogonal matrices and $\bar{A}_{11}$ has q columns. Here we can assume that $\bar{A}_{11}$ has full column rank q , and $[\bar{b}_1, \bar{A}_{11}]$ has full row rank $\bar{q}$, otherwise $\bar{Q}$ and $\bar{P}$ could be chosen to give $\bar{A}_{22}$ more columns or rows. From Lemma 1.2 we see that $\bar{q} = q$ if $\tilde{A}\tilde{x} \approx \tilde{b}$ is compatible and $\bar{q} = q+1$ if it is not.

Suppose $q < p$; then $\bar{A}_{11}$ must have fewer columns and rows than A_{11} . Obtain an SVD of $\tilde{A}$ by obtaining the individual SVDs of $\bar{A}_{11}$ (and transforming $\bar{b}_1$ accordingly) and $\bar{A}_{22}$, leading to the form of (2.2)–(2.3) with p replaced by q . But this would mean $\tilde{b}$ has nonzero projections on at most $q < p$ left singular subspaces of $\tilde{A}$ (see Lemma 2.1), which by assumption is false; so $q \geq p$, and (2.3) provides a minimally dimensioned, or *core*, problem within $\tilde{A}\tilde{x} \approx \tilde{b}$. $\square$

Remark 2.1. It follows from Definition 1.1, Lemma 2.1, and Theorem 2.2 that $[b_1, A_{11}]$ in (2.2)–(2.3) represents a core problem. From the form of this it can be shown that the TLS, scaled TLS, and DLS formulations have unique and meaningful solutions for any core problem $A_{11}x_1 \approx b_1$; see [20, (1.10) et seq.] and [21, (9) et al.]. This theory, and the method of solution of such problems, is discussed further in sections 3 (following Theorem 3.2), 4, and 5. In fact (1.10) in [20], and (9) in [21], is just (5.11) here.

The SVD is costly to compute, and the computation is necessarily iterative; see, for example, [12, sections 5.4.3–5, pp. 251–254]. In order to find a core problem, we do not need to follow the costly procedure described above. In section 3 we show how to find a core problem directly and cheaply. This will also give us the ideal first step towards computing c , δ , and S_1 in (2.2)–(2.3), should we want them.

Remark 2.2. Because of data and rounding errors, few practical problems will decompose computationally as in (1.2). However, if we have a good idea of the accuracy of our data and computer arithmetic, this analysis will allow us to go from (2.1) to (2.2)–(2.3) within this accuracy (cf. [11, section 5]), and this could be particularly useful for ill-posed problems. Suppose $\tilde{b}$ is only accurate to within $\beta\|\tilde{b}\|$, and $\tilde{A}$ to within $\alpha\|\tilde{A}\|$. Then here is the outline of an approach to get from (2.1) to the form in (2.2)–(2.3):

- Any elements of $|\tilde{c}|$ less than say $\beta\|\tilde{b}\|$ can be set to zero.
- Any diagonal elements of S less than say $\alpha\|\tilde{A}\|$ can be set to zero.

- Any block of diagonal elements of S which are equal to within say $2\alpha\|\tilde{A}\|$ can be set equal to their midvalue.
- The resulting new (2.1) can be transformed as above to (2.2) with (2.3).

3. Computing a core problem within $\tilde{A}\tilde{x} \approx \tilde{b}$. We can compute a decomposition of the form (1.2) directly by choosing orthogonal matrices P and Q to reduce $[\tilde{b}, \tilde{A}]$ to a real upper-bidiagonal matrix; see, for example, [12, section 5.4.3, p. 251] (and also [20, section 8]). Partitions $P = [P_1, P_2]$ and $Q = [Q_1, Q_2]$ are obtained by stopping at the first zero element, giving (1.2) where A_{22} has not been bidiagonalized, while upper-bidiagonal $[b_1, A_{11}] = P_1^T [\tilde{b}, \tilde{A}Q_1]$ has nonzero bidiagonal elements and is either

$$(3.1) \quad [b_1 | A_{11}] = \left[\begin{array}{c|cccc} \beta_1 & \alpha_1 & & & \\ & \beta_2 & \alpha_2 & & \\ & & \cdot & \cdot & \\ & & & \beta_p & \alpha_p \end{array} \right], \quad \beta_i \alpha_i \neq 0, \quad i = 1, \dots, p$$

if $\beta_{p+1} = 0$ or $p = n$; or

$$(3.2) \quad [b_1 | A_{11}] = \left[\begin{array}{c|cccc} \beta_1 & \alpha_1 & & & \\ & \beta_2 & \alpha_2 & & \\ & & \cdot & \cdot & \\ & & & \beta_p & \alpha_p \\ & & & & \beta_{p+1} \end{array} \right], \quad \beta_i \alpha_i \neq 0, \quad i = 1, \dots, p; \quad \beta_{p+1} \neq 0$$

if $\alpha_{p+1} = 0$ or $p = k$. In the first case $A_{11}x_1 = b_1$ in (1.3) is a compatible system since A_{11} is $p \times p$ and nonsingular. In the second case $A_{11}x_1 \approx b_1$ is an incompatible system since $[b_1, A_{11}]$ has rank $p + 1$. Note that under the assumption $\tilde{b} \notin \mathcal{R}(\tilde{A})$ we have $\alpha_1 \neq 0$ (see (1.5)).

Remark 3.1. Whether (3.1) or (3.2) results, this bidiagonalization has two important alternative interpretations, and these help us to understand its effectiveness. With the above partitioning of P and Q we see that

$$P^T \left[\begin{array}{c|c} \tilde{b} & \tilde{A}Q_1 \end{array} \right] = \left[\begin{array}{c|c} b_1 & A_{11} \\ 0 & 0 \end{array} \right],$$

so that the bidiagonalization gives the QR factorization of $[\tilde{b}, \tilde{A}Q_1]$, ensuring that $[b_1, A_{11}]$ has full row rank. Next we see that

$$\left[\begin{array}{c} P_1^T \tilde{A} \end{array} \right] Q = \left[\begin{array}{c|c} A_{11} & 0 \end{array} \right],$$

so that the bidiagonalization gives the LQ factorization of $P_1^T \tilde{A}$, ensuring that A_{11} has full column rank.

The proof that (3.1) and (3.2) correspond to core problems will be given in Theorems 3.2 and 3.3. But first we need a lemma.

LEMMA 3.1. *Let $J = [b_1, A_{11}]$ be bidiagonal as in (3.1) or (3.2). Then all its left and right singular vectors (for its p or $p + 1$ nonzero singular values) have nonzero first and last elements. The nonzero null-vector of (3.1) has no zero elements.*

Proof. With $u \equiv (\mu_1, \dots, \mu_{p+1})^T$, $v \equiv (\nu_1, \dots, \nu_{p+1})^T$, we see that

$$(3.3) \quad u^T J = \sigma v^T \Rightarrow \mu_1 \beta_1 = \sigma \nu_1; \quad \mu_i \alpha_i + \mu_{i+1} \beta_{i+1} = \sigma \nu_{i+1}, \quad i = 1, \dots, p;$$

$$(3.4) \quad Jv = u\sigma \Rightarrow \beta_i \nu_i + \alpha_i \nu_{i+1} = \mu_i \sigma, \quad i = 1, \dots, p; \quad \beta_{p+1} \nu_{p+1} = \mu_{p+1} \sigma;$$

where μ_{p+1} and β_{p+1} are nonexistent in (3.1). For $\sigma > 0$, (3.3) shows that if either μ_1 or ν_1 is zero, then so is the other, and then (3.4) and (3.3) show all the remaining elements are zero. Similar arguments give the rest of the proof. $\square$

THEOREM 3.2. *For $[b_1, A_{11}]$, $b_1 \neq 0$, $\alpha_1 \neq 0$, in (3.1) or (3.2) with SVD $A_{11} = \sum_{i=1}^p u_i \sigma_i v_i^T$, the p singular values σ_i of A_{11} are distinct and nonzero, and they strictly separate the $p+1$ distinct and nonzero singular values of $[b_1, A_{11}]$ in (3.2) (or the p distinct and nonzero singular values of $[b_1, A_{11}]$ together with 0 in (3.1)). In both cases*

$$(3.5) \quad \text{rank}(A_{11}) = p; \quad b_1^T u_i \neq 0, \quad i = 1, \dots, p.$$

Proof. From their obvious ranks, A_{11} has exactly p nonzero singular values, and $[b_1, A_{11}]$ has exactly p nonzero singular values in (3.1) and exactly $p+1$ in (3.2). But $T \equiv [b_1, A_{11}]^T [b_1, A_{11}]$ is $(p+1) \times (p+1)$ symmetric tridiagonal with nonzero next to diagonal elements, and $A_{11}^T A_{11}$ remains when the first row and column are deleted. Thus the eigenvalues of $A_{11}^T A_{11}$ strictly separate those of T ; see, for example, [26, Ch. 5, sect. 37, p. 300]. This proves the first part of the theorem. For the second part of (3.5) $b_1^T u_i = \beta_1 e_1^T u_i \neq 0$ from Lemma 3.1. $\square$

The condition (3.5) also directly ensures that the TLS, scaled TLS, and DLS formulations have unique and meaningful solutions for $A_{11}x_1 \approx b_1$ in the incompatible case (3.2); see [20, (1.10)], [21, (9)], and (5.11) with the discussion in section 5 ((3.5) implies (5.11) for $[b_1, A_{11}]$).

In Theorem 3.3 we will show that the orthogonal bidiagonalization leading to (3.1) or (3.2) gives a core problem $A_{11}x_1 \approx b_1$ in $\tilde{A}\tilde{x} \approx \tilde{b}$. It will help if we first briefly restate the relevant parts of Theorems 2.2 and 3.2.

THEOREM 2.2. *Suppose $\tilde{b} \notin \mathcal{R}(\tilde{A})$ has nonzero projections on exactly p left singular subspaces of $\tilde{A}$ corresponding to distinct nonzero singular values. Then among all decompositions of the form (1.2), the minimally dimensioned A_{11} is $p \times p$ if $\tilde{A}\tilde{x} \approx \tilde{b}$ is compatible, and $(p+1) \times p$ if $\tilde{A}\tilde{x} \approx \tilde{b}$ is incompatible.*

THEOREM 3.2. *For $[b_1, A_{11}]$, $b_1 \neq 0$, $\alpha_1 \neq 0$, in (3.1) or (3.2) with SVD $A_{11} = \sum_{i=1}^p u_i \sigma_i v_i^T$, we have that $\text{rank}(A_{11}) = p$ and $b_1^T u_i \neq 0$ for $i = 1, \dots, p$. Also the p singular values of A_{11} are distinct and nonzero. (The rest is omitted here.)*

THEOREM 3.3. *If $\tilde{b} \notin \mathcal{R}(\tilde{A})$ and $n \times (k+1)$ $[\tilde{b}, \tilde{A}]$ has an orthogonal decomposition of the form (1.2) with $[b_1, A_{11}]$ as in (3.1) or (3.2), then*

- (a) A_{11} has no zero or multiple singular values, and thus any zero singular values or repeats that $\tilde{A}$ has must appear in A_{22} ;
- (b) $[b_1, A_{11}]$ (and thus A_{11}) has minimal dimensions, and A_{22} maximal dimensions, over all orthogonal transformations of the form shown in (1.2);
- (c) orthogonal $\hat{U}_{11}$ and $\hat{V}_{11}$ in the transformation $\hat{U}_{11}^T [b_1, A_{11}] \hat{V}_{11}$ can be designed to produce the form of $[b_1, A_{11}]$ in (2.2)–(2.3).

Proof. (a) This follows immediately from Theorem 3.2. (b) Theorem 3.2 shows that $\tilde{b}$ has nonzero projections on exactly p left singular subspaces of $\tilde{A}$ corresponding to distinct nonzero singular values. Thus, following Theorem 2.2, $[b_1, A_{11}]$ is minimally dimensioned, so that A_{22} is maximally dimensioned. (c) Using the SVDs $A_{11} = U_{11}S_1V_{11}^T$, $A_{22} = U_{22}S_2V_{22}^T$, $U_{11}^T U_{11} = V_{11}^T V_{11} = I_p$, etc.,

$$\begin{aligned} & \left[\begin{array}{c|c|c} b_1 & A_{11} & 0 \\ \hline 0 & 0 & A_{22} \end{array} \right] = \left[\begin{array}{c|c|c} b_1 & U_{11}S_1V_{11}^T & 0 \\ \hline 0 & 0 & U_{22}S_2V_{22}^T \end{array} \right] \\ & = \left[\begin{array}{c|c|c} U_{11} & r_1 & 0 \\ \hline 0 & 0 & U_{22} \end{array} \right] \left[\begin{array}{c|c|c} c & S_1 & 0 \\ \hline \delta & 0 & 0 \\ \hline 0 & 0 & S_2 \end{array} \right] \left[\begin{array}{c|c|c} 1 & 0 & 0 \\ \hline 0 & V_{11}^T & 0 \\ \hline 0 & 0 & V_{22}^T \end{array} \right], \end{aligned}$$

where $c \equiv U_{11}^T b_1$, $w \equiv b_1 - U_{11}c$, and if $w \neq 0$, $\delta \equiv \|w\|$, $r_1 \equiv w/\delta$. If $\tilde{A}\tilde{x} = \tilde{b}$ is compatible, then U_{11} is square, $w = 0$, and in this case δ and its row, and r_1 and its column, are nonexistent. In the incompatible case $w \neq 0$, $\|r_1\| = 1$, $U_{11}^T r_1 = 0$, and denoting $\hat{U}_{11} \equiv [U_{11}, r_1]$, $\hat{V}_{11} \equiv V_{11}$ gives the structure in (2.2), while Theorem 3.2 shows (2.3) holds for this structure. $\square$

Theorems 2.2 and 3.3 are new. The result (b) of Theorem 3.3 was mentioned in [20, 21], but not proven, because we had not then obtained a sufficiently readable proof of what seemed a fairly obvious result. Theorem 2.2 allowed this, but we hope that others can provide an even simpler proof. The essence of Theorem 3.2 was given in [21, Thm. 1], which is an extended version of [20, Thm. 8.1].

Remark 3.2. In practical computations which involve rounding errors or noise in the data, one must consider threshold criteria to decide which elements of the bidiagonal matrix are small in magnitude and should be set to zero. The criteria will be problem-dependent, but unlike the case in Remark 2.2, their choice is less obvious and needs further investigation.

It will be interesting to relate the core problem formulation to the work on truncated TLS [4, 5]; see also [15, section 6.6] and [24]. In particular, the core problem formulation can be considered as a theoretical basis for the Lanczos truncated TLS proposed in [6, section 4.1] as well as for the partial least squares (PLS) method of Wold, et al. [27] which is equivalent to the Lanczos bidiagonalization-based truncated least squares; see [3]. Various related regularization aspects are described, for example, in [2, 18, 14, 8]; see also [28].

4. Solving the LS, scaled TLS, and DLS problems using bidiagonalization and the core problem. Consider the upper bidiagonalization

$$[b, A] = P^T [\tilde{b}, \tilde{A}Q] \quad \text{of the form in (1.2),}$$

with the core problem part $[b_1, A_{11}] = P_1^T [\tilde{b}, \tilde{A}Q_1]$ given by (3.1) or (3.2). In (3.1) $A_{11}x_1 = b_1$ is a compatible system, therefore $\tilde{A}\tilde{x} = \tilde{b}$ is a compatible system (see Lemma 1.2). Then the LS residual, the scaled TLS distance (for any positive finite γ), and the DLS distance are zero, and the solutions are obvious. We will now consider only the incompatible case (3.2) and take $x_2 = 0$, $\tilde{x} = Q_1x_1$.

The LS solution of $A_{11}x_1 \approx b_1$ with $[b_1, A_{11}]$ in (3.2) is obtained by orthogonal reduction of the matrix $[A_{11}, b_1]$ to upper triangular form (note the reversal of A_{11} and b_1), followed by solution of a triangular system to give x_1 .

For any given finite positive scaling γ , the scaled TLS solution (see (1.4)) of $A_{11}x_1 \approx b_1$ with $[b_1, A_{11}]$ in (3.2) is uniquely determined by the unique SVD component $\bar{u}\bar{\sigma}\bar{v}^T$ of $[\gamma b_1, A_{11}]$ corresponding to $\bar{\sigma} \equiv \sigma_{\min}([\gamma b_1, A_{11}])$, its minimal singular value. From Lemma 3.1, $\nu = e_1^T \bar{v} \neq 0$, $\bar{v}^T \equiv [\nu, \bar{w}^T]$ and

$$x_1 = -\bar{w}/\nu \quad \text{with the scaled TLS correction} \quad -\bar{u}\bar{\sigma}\bar{v}^T \quad \text{to} \quad [\gamma b_1, A_{11}].$$

In the original variables the scaled TLS solution of $\tilde{A}\tilde{x} \approx \tilde{b}$ is $\tilde{x} = Q_1x_1$. The scaled TLS correction to $[\gamma\tilde{b}, \tilde{A}]$ is obtained from the following exercise:

$$\begin{aligned} [\tilde{g}, \tilde{E}] &= P \begin{bmatrix} -\bar{u}\bar{\sigma}\bar{v}^T & 0 \\ 0 & 0 \end{bmatrix} \begin{bmatrix} 1 & 0 \\ 0 & Q^T \end{bmatrix} = [P_1|P_2] \left[\begin{array}{c|c} -\bar{u}\bar{\sigma}(\nu, \bar{w}^T) & 0 \\ \hline 0 & 0 \end{array} \right] \begin{bmatrix} 1 & 0 \\ 0 & Q_1^T \\ 0 & Q_2^T \end{bmatrix} \\ &= -(P_1\bar{u})\bar{\sigma}(\nu, \bar{w}^T Q_1^T) = (P_1\bar{u})\bar{\sigma}\nu[-1, \tilde{x}^T]. \end{aligned}$$

The DLS problem was proposed and used in [13]. Suppose that the core part $[b_1, A_{11}]$ of the transformed $[\tilde{b}, \tilde{A}]$ has the form in (3.2). We will show how to solve the DLS problem (1.7) for this core data; see [20, 21]. Write

$$[b_1|A_{11}] \equiv \left[\begin{array}{c|c} \beta_1 & \alpha_1 e_1^T \\ \hline 0 & A_2 \end{array} \right] \equiv P_1^T [\tilde{b}|\tilde{A}Q_1], \quad E_{11} \equiv \left[\begin{array}{c} e^T \\ \hline E_2 \end{array} \right] \equiv P_1^T \tilde{E}Q_1, \quad \tilde{x} \equiv Q_1 x_1,$$

where A_2 is square with all its singular values distinct and nonzero. Let σ , u , and v be the minimum singular value $\sigma \equiv \sigma_{\min}(A_2)$ and its left and right vectors for A_2 . For the above reduced data the DLS problem (1.7) becomes

$$\min_{e, E_2, x_1} \{ \|e\|^2 + \|E_2\|_F^2 \} \quad \text{subject to} \quad \left[\begin{array}{c|c} \beta_1 & \alpha_1 e_1^T + e^T \\ \hline 0 & A_2 + E_2 \end{array} \right] \begin{bmatrix} -1 \\ x_1 \end{bmatrix} = 0.$$

Since β_1 is nonzero, $x_1 \neq 0$, and the minimum $\|E_2\|_F$ in $(A_2 + E_2)x_1 = 0$ is σ , given by $E_2 = -u\sigma v^T$ and $x_1 = v\xi$ for some $\xi \neq 0$. To make this x_1 satisfy the full constraints we need $\beta_1 = \alpha_1 e_1^T v\xi + e^T v\xi$. But $e_1^T v \neq 0$ from Lemma 3.1, so $e = 0$ gives the overall minimum with $\xi = \beta_1/(\alpha_1 e_1^T v)$, and

$$x_D \equiv x_1 = v\beta_1/(\alpha_1 e_1^T v), \quad \sigma_D \equiv \sigma = \sigma_{\min}(A_2)$$

are the DLS solution and distance in (1.7) for the reduced data $[b_1, A_{11}]$. In the original variables $\tilde{x} = Q_1 x_1 = Q_1 v\xi$ and

$$\tilde{E} = P \begin{bmatrix} E_{11} & 0 \\ 0 & 0 \end{bmatrix} Q^T = P_1 E_{11} Q_1^T = P_1 \begin{bmatrix} 0 \\ -u\sigma v^T \end{bmatrix} Q_1^T = P_1 \begin{bmatrix} 0 \\ -u(\sigma/\xi)\tilde{x}^T \end{bmatrix}.$$

From these we see that the solutions of the scaled TLS and DLS problems are reduced to computing the smallest singular value and its right singular vector for the nonsingular bidiagonal matrices $[b_1, A_{11}]$ and A_2 , respectively, and these are relatively easy to find; see, for example, [12, section 8.6.2, pp. 452–456].

5. Review and comparisons with earlier work. The main topics we have dealt with so far are (in order of presentation) as follows:

- T1. Core problems $A_{11}x_1 \approx b_1$ in $\tilde{A}\tilde{x} \approx \tilde{b}$; see Definition 1.1 and Theorem 2.2.
- T2. Some criteria for solving $\tilde{A}\tilde{x} \approx \tilde{b}$; see (1.4), (1.7).
- T3. A special form of the SVD of $\tilde{A}$ related to $\tilde{b}$, which gives the core problem; see (2.2)–(2.3).
- T4. The orthogonal upper bidiagonalization of $[\tilde{b}, \tilde{A}]$ as the way of computing the core problem efficiently; see (3.1)–(3.2).
- T5. Implementing topic T2 for topic T1; see section 4 (also [1, 11, 12, 25, 20, 21]).

Our main purpose has been to introduce T3, show its relationship to T4, and prove that the core problem in T1 could be found immediately from either T3 or T4. Using T4 to obtain T1 is a direct computation, and is also an ideal first step in either computing T3 (it gives an orthogonal transformation of $\tilde{A}$ to lower bidiagonal form) or in solving (1.4) or (1.7). Using T3 is an iterative process (but very fast after the bidiagonalization in T4) and gives an excellent guide as to what elements can be considered zero or equal in different cases; see Remark 2.2. An analogous guide which would apply directly to T4 needs further work; see Remark 3.2.

For applications, our suggested approach to finding a scaled TLS solution to (1.1) is to determine the core problem (3.1)–(3.2) and continue according to (1.6). Then (1.4) with finite positive γ for $A_{11}x_1 \approx b_1$ can be solved in the manner reviewed briefly

in section 4 and described in more detail from here to (5.5) (based on the algorithm of Golub and Van Loan in [11]; see [12, section 12.3]), while the DLS problem can be solved as in section 4. These will always give meaningful solutions (see Remark 2.1), and because of minimal dimensions of $[b_1, A_{11}]$, they will also minimize the cost.

On reading this, Sabine van Huffel suggested that we also compare the core problem approach here to the approaches in [25] for solving the TLS problem (with a single right-hand side $\tilde{b}$, since the extension of the theory here to multiple right-hand side problems remains a subject for further investigation). This will require some basic analysis. We need only consider the incompatible case and $\tilde{b} \notin \mathcal{R}(\tilde{A})$.

One reason for developing our formulation (1.4) was that its solution is clearly equivalent to the TLS solution of $\tilde{A}(\tilde{x}\gamma) \approx \tilde{b}\gamma$. Thus by just considering

$$(5.1) \quad \text{TLS distance} \equiv \min_{g, E, x} \|[g, E]\|_F \quad \text{subject to} \quad (A + E)x = b + g,$$

the results we obtain will also cover the extension of the results in [25] to $\{(1.4) \text{ with any finite positive } \gamma\}$. We will also assume that

$$(5.2) \quad [b, A] \equiv P^T[\tilde{b}, \tilde{A}Q] \quad \text{has the form in (1.2) with } [b_1, A_{11}] \text{ given by (3.2),}$$

and in this case we will show that the minimum 2-norm solution approaches that were given in [25] theoretically give the same answers to (5.1) as (1.6) does. Since the optimum in (5.1) is independent of such orthogonal transformations, this will mean that the answers for any $[\tilde{b}, \tilde{A}]$ will theoretically be the same here and for the minimum 2-norm solution approaches in (the extended versions of) [25].

The approaches to solving (5.1)–(5.2) make use of the SVD of $n \times (k+1)$ $[b, A]$, which we write as:

$$(5.3) \quad [b, A] = \tilde{U}\tilde{S}\tilde{V}^T = \sum_{i=1}^{k+1} \tilde{u}_i \tilde{\sigma}_i \tilde{v}_i^T, \quad \tilde{\sigma}_1 \geq \cdots \geq \tilde{\sigma}_{k+1} \geq 0.$$

Here we have assumed $n > k$, which can be attained by adding zero rows to $[b, A]$ if necessary. Note that the SVD used in [25] (see, for example, [25, equation (1.22), p. 22]) does not take account of the possible structure in (1.2), while here we only use the SVD of $[b_1, A_{11}]$; the SVD of A_{22} need not be computed. Thus in our case the SVD of $[b, A]$ in (5.3) is the direct sum of the SVDs of $[b_1, A_{11}]$ and A_{22} . We can still use the ordering of vectors and singular values in (5.3), but for equal singular values we will always assume the first in order comes from the $[b_1, A_{11}]$ block. Clearly for our version of (5.1)–(5.3), any SVD component $\tilde{u}_i \tilde{\sigma}_i \tilde{v}_i^T$ coming from the $[b_1, A_{11}]$ block can be nonzero only in its leading principle $(p+1) \times (p+1)$ block. Similarly for our version, any SVD component coming from the A_{22} block can be nonzero only in the block corresponding to A_{22} .

The SVD used in [25] is not precisely defined, but since we want to compare the solution $\begin{bmatrix} x_1 \\ 0 \end{bmatrix}$ here based on $A_{11}x_1 \approx b_1$ to the solution x in [25] based on $Ax \approx b$ in (1.2), it is easiest to assume [25] uses the same SVD as here and comment on the effects of using a more general SVD.

First we derive our unique solution (1.6) to (1.1) with core problem $A_{11}x_1 \approx b_1$. As in section 4, let $[b_1, A_{11}]$ have unique “minimum” SVD component $\bar{u}\bar{\sigma}\bar{v}^T$, where

$$(5.4) \quad \bar{\sigma} \equiv \sigma_{\min}([b_1, A_{11}]) = \tilde{\sigma}_m \quad \text{say, in (5.3).}$$

$\bar{\sigma}$ will be unique in the SVD of $[b_1, A_{11}]$ (see Theorem 3.2), but we choose $\tilde{\sigma}_m$ to be the first of any equals in the SVD (5.3). From Lemma 3.1, $\nu \equiv e_1^T \bar{v} \neq 0$. Define

$$[g, E] \equiv -\tilde{u}_m \tilde{\sigma}_m \tilde{v}_m^T = - \begin{bmatrix} \bar{u} \bar{\sigma} \bar{v}^T & 0 \\ 0 & 0 \end{bmatrix}, \quad \tilde{u}_m = \begin{bmatrix} \bar{u} \\ 0 \end{bmatrix}, \quad \tilde{v}_m = \begin{bmatrix} \bar{v} \\ 0 \end{bmatrix}.$$

Denote

$$\tilde{v}_m \equiv \begin{bmatrix} \nu \\ w \end{bmatrix}, \quad w \equiv \begin{bmatrix} \bar{w} \\ 0 \end{bmatrix}, \quad \bar{v} = \begin{bmatrix} \nu \\ \bar{w} \end{bmatrix}.$$

The solution to the TLS problem (5.1) for $A_{11}x_1 \approx b_1$ is then $x_1 = -\bar{w}/\nu$, with correction $-\bar{u}\bar{\sigma}\bar{v}^T$ to $[b_1, A_{11}]$. With (1.6) this means our solution to $Ax \approx b$ is

$$(5.5) \quad x = -w/\nu, \text{ where } \tilde{v}_m \equiv \begin{bmatrix} \nu \\ w \end{bmatrix}, \text{ with correction } [g, E] \equiv -\tilde{u}_m \tilde{\sigma}_m \tilde{v}_m^T \text{ to } [b, A].$$

We will show that the minimum 2-norm solutions in [25] are identical to this. To help in this we will use the following lemma.

LEMMA 5.1. *The singular values of A interlace those of $[b, A]$ ([12, Cor.8.6.3]).*

There are exactly three cases, denoted 1, 2(a), and 2(b) below.

Case 1. If the most well-known criterion for a unique solution to (5.1) holds, i.e.

$$(5.6) \quad \tilde{\sigma}_{k+1} \equiv \sigma_{\min}([b, A]) < \sigma_{\min}(A),$$

(see [11, Thm.4.1], [12, Thm.12.3.1], [25, Thm.2.6, p.35]), then in (5.4), $\bar{\sigma} = \tilde{\sigma}_m = \tilde{\sigma}_{k+1}$, which from Lemma 5.1 is the unique minimum singular value of $[b, A]$, and the solution in [25, Thm. 2.6, p. 35] to (5.1) is unique and given by (5.5), and so is identical to the solution here.

Case 2. Next suppose for a general SVD in (5.3) that for some $j \leq k$

$$(5.7) \quad \tilde{\sigma}_j > \tilde{\sigma}_{j+1} = \cdots = \tilde{\sigma}_{k+1}, \quad V' \equiv [\tilde{v}_{j+1}, \dots, \tilde{v}_{k+1}], \quad U' \equiv [\tilde{u}_{j+1}, \dots, \tilde{u}_{k+1}].$$

If $j < k$, then from Lemma 5.1, (5.6) does not hold. If $e_1^T V' = 0$, then it can be seen from Lemma 3.1 that $\tilde{\sigma}_{j+1} = \cdots = \tilde{\sigma}_{k+1}$ must correspond wholly to A_{22} , and (5.6) does not hold. These lead to the special cases in [25].

Case 2(a). If $j < k$ and $e_1^T V' \neq 0$, the TLS solution is not unique. Golub and Van Loan [11, pp. 885–886], and later Van Huffel and Vandewalle, effectively design orthogonal Q' so that $e_1^T V' Q' = \nu e_1^T$ and set $\tilde{u} \equiv U' Q' e_1$ and $\tilde{v} \equiv V' Q' e_1 = (\nu, w^T)^T$. Then it was proven in [25, Thm. 3.7, p. 58] that $[g, E] \equiv -\tilde{u} \tilde{\sigma}_{j+1} \tilde{v}^T$ is an optimal correction to $[b, A]$ in (5.1) with minimum 2-norm solution $x = -w/\nu$ and distance $\tilde{\sigma}_{j+1}$. For our structured form of SVD in (5.3), we see from Theorem 3.2 that the $p+1$ singular values of $[b_1, A_{11}]$ in (3.2) are distinct and nonzero, so at most one can come from among $\tilde{\sigma}_{j+1} = \cdots = \tilde{\sigma}_{k+1}$ above, and the other $k-j$ (or $k-j+1$) in this set of equal values must come from A_{22} in (1.2). Now if $\tilde{v}_{j+1}, \dots, \tilde{v}_{k+1}$ all came from A_{22} , then $e_1^T V' = 0$, a contradiction. So exactly one such vector (with our chosen ordering for (5.3) it is $\tilde{v}_{j+1}$) comes from $[b_1, A_{11}]$, and by Lemma 3.1 it has nonzero first element. What has happened is that the splitting (1.2) caused by our bidiagonalization ensures that V' already satisfies $e_1^T V' = \pm \nu e_1^T$. Thus $m = j+1$ in (5.4), and our solution (5.5) is again obtained in [25, Thm. 3.7, p. 58].

Case 2(b). The remaining case is where $e_1^T V' = 0$ in (5.7) (which may happen when $j = k$ or $j < k$). This is called the “nongeneric” case in [25, section 3.4] and

corresponds to the unpleasant example that we gave near the end of section 1. In linear regression the columns of such V' correspond to nonpredictive multicollinearities; they are of no value in predicting the response b [25, p. 71]. To handle this, Van Huffel and Vandewalle [25, Def. 3.2, p. 68] define a new problem, a more constrained version of (5.1). The strategy is to eliminate those directions in A , corresponding to the smallest singular value or to the several smallest singular values of $[b, A]$, that are not at all correlated with the observation vector b . For any general SVD (5.3), let q be the maximum value of i such that $e_1^T \tilde{v}_i \neq 0$. Note that we might have $q < j$ in (5.7). The “nongeneric” problem of [25, Def. 3.2, p. 68] is then just (5.1) with the added restriction that $[g, E][\tilde{v}_{q+1}, \dots, \tilde{v}_{k+1}] = 0$, and any solution x for this is called a “nongeneric TLS solution.” In [25, Thm. 3.12, p. 72] it is shown how to obtain such a solution, and the comments following that indicate how to compute the minimum 2-norm solution.

For our structured form of SVD in (5.3), define q as above, set $\tilde{V}_1 = [\tilde{v}_1, \dots, \tilde{v}_q]$, and with conformable partitioning in (5.3) write

$$(5.8) \quad \tilde{U} = [\tilde{U}_1, \tilde{U}_2], \quad \tilde{S} = \text{diag}(\tilde{S}_1, \tilde{S}_2), \quad \tilde{V} = [\tilde{V}_1, \tilde{V}_2] \quad \text{so that } [g, E] \tilde{V}_2 = 0$$

is the added constraint. Lemma 3.1 with $e_1^T \tilde{V}_2 = 0$ shows that the singular vectors $\tilde{v}_{q+1}, \dots, \tilde{v}_{k+1}$ come from A_{22} ; $e_1^T \tilde{v}_q \neq 0$ shows that $\tilde{v}_q$ comes from $[b_1, A_{11}]$, which with the ordering in (5.3) implies $m = q$ in (5.4); and (5.7) shows $\tilde{\sigma}_q \equiv \sigma_{\min}([b_1, A_{11}]) > \sigma_{\min}(A_{22})$, so (5.6) does not hold. Clearly our solution (5.5) satisfies the constraints in (5.1) and (5.8), but we still need to show it is optimal and minimal in length.

We can eliminate the explicit constraint $[g, E] \tilde{V}_2 = 0$ by taking

$$[g, E] \equiv \tilde{U} H \tilde{V}_1^T \equiv (\tilde{U}_1 H_1 + \tilde{U}_2 H_2) \tilde{V}_1^T$$

and now minimize $\|H\|_F$. By defining $y^T \equiv (y_1^T, y_2^T) \equiv (-1, x^T)[\tilde{V}_1, \tilde{V}_2]$ and transforming the constraints in (5.1) to

$$0 = \tilde{U}^T \{[b, A] + [g, E]\} \tilde{V} \tilde{V}^T \begin{bmatrix} -1 \\ x \end{bmatrix} = \begin{bmatrix} \tilde{S}_1 + H_1 & 0 \\ H_2 & \tilde{S}_2 \end{bmatrix} \begin{bmatrix} y_1 \\ y_2 \end{bmatrix},$$

we can reformulate the “nongeneric problem” in [25, Def. 3.2, p. 68] to

$$(5.9) \quad \min_{H_1, H_2, y_1, y_2, x} (\|H_1\|_F^2 + \|H_2\|_F^2) \\ \text{subject to} \quad \begin{bmatrix} \tilde{S}_1 + H_1 & 0 \\ H_2 & \tilde{S}_2 \end{bmatrix} \begin{bmatrix} y_1 \\ y_2 \end{bmatrix} = 0, \quad \begin{bmatrix} -1 \\ x \end{bmatrix} = \tilde{V}_1 y_1 + \tilde{V}_2 y_2.$$

Now whatever H_1 and y_1 are, we can take $H_2 = 0$ and $y_2 = 0$ (since $e_1^T \tilde{V}_2 = 0$, only y_1 contributes to the -1 in (5.9), and thus the last constraint gives no restriction on y_2). Therefore, the problem simplifies to

$$(5.10) \quad \min_{H_1, y_1, x} \|H_1\|_F^2 \quad \text{subject to} \quad (\tilde{S}_1 + H_1)y_1 = 0, \quad \begin{bmatrix} -1 \\ x \end{bmatrix} = \tilde{V}_1 y_1.$$

Since $\tilde{S}_1 = \text{diag}(\tilde{\sigma}_1, \dots, \tilde{\sigma}_q)$ with $\tilde{\sigma}_1 \geq \dots \geq \tilde{\sigma}_q = \sigma_{\min}([b_1, A_{11}]) > 0$ (see Theorem 3.2), a solution is given by $H_1 = -e_q \tilde{\sigma}_q e_q^T$, $y_1 = e_q / (-\nu)$, giving our solution (5.5) with $m = q$. But this is the minimum norm solution, since $\tilde{v}_q$ is the only right singular vector with nonzero first element that corresponds to $\tilde{\sigma}_q$, using our version

of (5.3). Thus our approach (1.6) immediately gives the minimum 2-norm solution *satisfying (5.1) together with the added restriction in (5.8)*, so again the minimum 2-norm solution in [25] is the same as the solution here. This concludes our discussion of equivalent solutions.

Now let $\mathcal{U}_{\min}$ be the left singular vector subspace of A corresponding to $\sigma_{\min}(A)$. We argued in [20, section 7] that (5.6) was not ideal and that a satisfactory condition for ensuring unique solutions to, and for building the theory of, the TLS, DLS, and scaled TLS formulations for solving (1.1) is the following γ -independent criterion:

$$(5.11) \quad \text{the } n \times k \text{ matrix } A \text{ has rank } k \text{ and } b \notin \mathcal{U}_{\min}.$$

We showed in [20, Thm. 3.1] that this implies $\sigma_{\min}([b\gamma, A]) < \sigma_{\min}(A)$ for *all* $\gamma \geq 0$, which is important for scaled TLS, and of course implies (5.6).

A crucial property of any core problem $A_{11}x_1 \approx b_1$ is that A_{11} and b_1 *always* satisfy (5.11), see (3.5), or (2.2)–(2.3), and so its scaled TLS solution exists, is unique, and can be computed from the SVD of $[b_1\gamma, A_{11}]$. This can be computed efficiently from either of the forms of $[b_1, A_{11}]$ in (3.2) or (2.2). The bidiagonalization leads to $[b_1, A_{11}]$ satisfying even more than just (5.11), since it removes all redundancies and irrelevant parts of the problem corresponding to all singular values, and in theory it does this implicitly before any singular value computation. Minimizing the dimensions of $[b_1, A_{11}]$ also maximizes the computational efficiency.

The solution of the original problem (compatible or not) obtained via (1.6) is then the minimum 2-norm solution of (1.1) such that the core problem is solved, and any corrections correspond to corrections only in $[b_1, A_{11}]$ in (1.2).

6. Conclusion. If $\tilde{A}$ has full column rank, these results show us that we can only find a decomposition of the form (1.2) with nontrivial A_{22} if either $\tilde{b}$ is orthogonal to a left singular vector subspace of $\tilde{A}$ or $\tilde{A}$ has at least one repeated singular value or both. For any $[\tilde{b}, \tilde{A}]$, the bidiagonalization (3.1) or (3.2) (a direct computation, that is, noniterative) will provide that decomposition with minimally dimensioned A_{11} in (1.2). This bidiagonalization will also show whether the original problem (1.1) is compatible or not, and is an ideal first step in solving the TLS, scaled TLS, or DLS formulations for finding the optimal solution to (1.1); see [20, 21]. In some cases (for example, using LSQR in [19]) it is also an excellent first step for ordinary linear LS problems. We showed how this bidiagonalization and the core problem formulation can be used to solve the LS, scaled TLS, and DLS problems.

The TLS solutions obtained via (1.6) were shown to be theoretically identical to the minimum 2-norm solutions of all formulations of TLS in [25]. The one simple and efficient approach given here can be applied to all such problems with a single right-hand side, while different, and more complicated, approaches were needed to solve different classes of problems in [25].

We feel that in addition to its more general contributions, this study simplifies and extends the body of work which was essentially started in [10, 7, 11], but which has been so extensively developed in [25] and elsewhere; see, for example, the collections containing [22, 21].

These results revise our understanding of both the theory and computations in all forms of linear LS problems with a single right-hand side. So far we have presented essentially theoretical and algorithmic ideas, while many implementation details still need to be worked out; see, for example, Remarks 2.2 and 3.2.

Acknowledgments. We thank Sabine van Huffel for her valuable suggestion that we show in all cases the mathematical equivalence of the TLS solutions in [25]

with the corresponding solutions obtained by the theoretically more direct approach here. It was difficult to find a rigorous way to do this that was still reasonably readable, but we agree it was important to complete this link. We feel the end result of section 5 rounds out the work nicely, even though the proofs could be more elegant.

For improvements to the revised version we thank Per Christian Hansen both for pointing out a useful reference and offering valuable suggestions which led to Remarks 1.1 and 3.2; Gene Golub for supplying three historical references; and an anonymous referee for carefully reading the manuscript and providing comments which improved the presentation nicely.

REFERENCES

- [1] A. BJÖRCK, *Numerical Methods for Least Squares Problems*, SIAM, Philadelphia, 1996.
- [2] A. BJÖRCK, E. GRIMME, AND P. VAN DOOREN, *An implicit shift bidiagonalization algorithm for ill-posed problems*, BIT, 34 (1994), pp. 510–534.
- [3] L. ELDÉN, *Partial least-squares vs. Lanczos bidiagonalization-I: Analysis of a projection method for multiple regression*, Comput. Statist. Data Anal., 46 (2004), pp. 11–31.
- [4] R. D. FIERRO AND J. R. BUNCH, *Collinearity and total least squares*, SIAM J. Matrix Anal. Appl., 15 (1994), pp. 1167–1181.
- [5] R. D. FIERRO AND J. R. BUNCH, *Perturbation theory for orthogonal projection methods with applications to least squares and total least squares*, Linear Algebra Appl., 234 (1996), pp. 71–96.
- [6] R. D. FIERRO, G. H. GOLUB, P. C. HANSEN, AND D. P. O’LEARY, *Regularization by truncated total least squares*, SIAM J. Sci. Comput., 18 (1997), pp. 1223–1241.
- [7] G. H. GOLUB, *Some modified matrix eigenvalue problems*, SIAM Rev., 15 (1973), pp. 318–334.
- [8] G. H. GOLUB, P. C. HANSEN, AND D. P. O’LEARY, *Tikhonov regularization and total least squares*, SIAM J. Matrix. Anal. Appl., 21 (1999), pp. 185–194.
- [9] G. H. GOLUB, A. HOFFMAN, AND G. W. STEWART, *A generalization of the Eckart-Young-Mirsky matrix approximation theorem*, Linear Algebra Appl., 88/89 (1987), pp. 317–327.
- [10] G. H. GOLUB AND C. REINSCH, *Singular value decomposition and least squares solutions*, Numer. Math., 14 (1970), pp. 403–420.
- [11] G. H. GOLUB AND C. F. VAN LOAN, *An analysis of the total least squares problem*, SIAM J. Numer. Anal., 17 (1980), pp. 883–893.
- [12] G. H. GOLUB AND C. F. VAN LOAN, *Matrix Computations*, 3rd ed., The Johns Hopkins University Press, Baltimore, MD, 1996.
- [13] R. D. D. GROAT AND E. M. DOWLING, *The data least squares problem and channel equalization*, IEEE Trans. Signal Process., 42 (1993), pp. 407–411.
- [14] M. HANKE, *On Lanczos based methods for the regularization of discrete ill-posed problems*, BIT, 41 (2001), pp. 1008–1018.
- [15] P. C. HANSEN, *Rank-Deficient and Discrete Ill-Posed problems: Numerical Aspects of Linear Inversion*, SIAM Monogr. Math. Model. Comput. 4, SIAM, Philadelphia, 1997.
- [16] P. C. HANSEN, T. SEKII, AND H. SHIBAHASHI, *The modified truncated SVD method for regularization in general form*, SIAM J. Sci. Statist. Comput., 13 (1992), pp. 1142–1150.
- [17] R. A. HORN AND C. R. JOHNSON, *Topics in Matrix Analysis*, Cambridge University Press, Cambridge, UK, 1991.
- [18] D. P. O’LEARY AND J. A. SIMMONS, *A bidiagonalization-regularization procedure for large-scale discretizations of ill-posed problems*, SIAM J. Sci. Statist. Comput., 2 (1981), pp. 474–489.
- [19] C. C. PAIGE AND M. A. SAUNDERS, *LSQR: An algorithm for sparse linear equations and sparse least squares*, ACM Trans. Math. Software, 8 (1982), pp. 43–71.
- [20] C. C. PAIGE AND Z. STRAKOŠ, *Scaled total least squares fundamentals*, Numer. Math., 91 (2002), pp. 117–146.
- [21] C. C. PAIGE AND Z. STRAKOŠ, *Unifying least squares, total least squares and data least squares*, in Total Least Squares and Errors-in-Variables Modeling, S. van Huffel and P. Lemmerling, eds., Kluwer Academic Publishers, Dordrecht, The Netherlands, 2002, pp. 25–34.
- [22] B. D. RAO, *Unified treatment of LS, TLS, and truncated SVD methods using a weighted TLS framework*, in Recent Advances in Total Least Squares Techniques and Errors-in-Variables Modelling, S. Van Huffel, ed., SIAM, Philadelphia, 1997, pp. 11–20.
- [23] A. VAN DER SLUIS, *Stability of the solutions of linear least squares problems*, Numer. Math., 23 (1975), pp. 241–254.

- [24] A. VAN DER SLUIS AND G. W. VELTKAMP, *Restoring rank and consistency by orthogonal projection*, Linear Algebra Appl., 28 (1979), pp. 257–278.
- [25] S. VAN HUFFEL AND J. VANDEWALLE, *The Total Least Squares Problem: Computational Aspects and Analysis*, Frontiers Appl. Math. 9, SIAM, Philadelphia, 1991.
- [26] J. H. WILKINSON, *The Algebraic Eigenvalue Problem*, Clarendon Press, Oxford, UK, 1965.
- [27] S. WOLD, A. RUHE, H. WOLD, AND W. J. DUNN, III, *The collinearity problem in linear regression. The partial least squares (PLS) approach to generalized inverses*, SIAM J. Sci. Statist. Comput., 5 (1984), pp. 735–743.
- [28] A. I. ZHDANOV AND O. A. KATSYUBA, *Strong consistency of estimates made by the method of orthogonal projections*, Internat. J. Systems Sci., 21 (1990), pp. 1463–1471.

ON THE FACTORIZATION OF HYPERBOLIC AND UNITARY TRANSFORMATIONS INTO ROTATIONS*

MICHAEL STEWART[†] AND PAUL VAN DOOREN[‡]

Abstract. This paper presents a Σ -unitary analogue to the CS decomposition of a partitioned unitary matrix. The hyperbolic rotations revealed by the decomposition are shown to be optimal in that, among a broader class of decompositions of Σ -unitary matrices into elementary hyperbolic rotations, they are the smallest possible in a sum-of-squares sense. A similar optimality property is shown to hold for the sines in the CS decomposition of a unitary matrix.

Key words. CS decomposition, hyperbolic rotations, plane rotations, hyperbolic triangularization

AMS subject classifications. 15A23, 15A45

DOI. 10.1137/050635328

1. Background. In block Cholesky downdating problems and in block implementations [3, 4] of the generalized Schur algorithm for the Cholesky factorization of a block Toeplitz matrix, it is necessary to compute a Σ -unitary matrix

$$(1) \quad H = \begin{pmatrix} H_{11} & H_{12} \\ H_{21} & H_{22} \end{pmatrix}$$

such that

$$(2) \quad H \begin{pmatrix} A \\ B \end{pmatrix} = \begin{pmatrix} \hat{A} \\ 0 \end{pmatrix},$$

where A and $\hat{A}$ are $p \times m$ and B is $q \times m$. The matrix $\hat{A}$ is not assumed to be upper triangular. The transformation H is required to satisfy the Σ -orthogonality relation

$$(3) \quad H^H \Sigma H = \Sigma$$

with respect to the signature matrix

$$\Sigma = \begin{pmatrix} I_p & 0 \\ 0 & -I_q \end{pmatrix}.$$

Any transformation, H , satisfying (3) is referred to as Σ -unitary.

In block Cholesky downdating and in the block generalized Schur algorithm, $A^H A - B^H B$ is always positive definite. We will also use the notation $A^H A - B^H B > 0$ or $A^H A > B^H B$ for this assumption. Positive definiteness is sufficient to guarantee the existence of a Σ -unitary transformation satisfying (2). In particular if $A^\dagger = (A^H A)^{-1} A^H$, then we can choose

$$H = \begin{pmatrix} (I - (BA^\dagger)^H (BA^\dagger))^{-1/2} & 0 \\ 0 & (I - (BA^\dagger)(BA^\dagger)^H)^{-1/2} \end{pmatrix} \begin{pmatrix} I & -(BA^\dagger)^H \\ BA^\dagger & -I \end{pmatrix}.$$

*Received by the editors September 18, 2002; accepted for publication (in revised form) by H. A. van der Vorst July 27, 2005; published electronically January 27, 2006.

<http://www.siam.org/journals/simax/27-3/63532.html>

[†]Department of Mathematics and Statistics, Georgia State University, Atlanta, GA 30303 (mastewart@gsu.edu).

[‡]CESAME, Université Catholique de Louvain, Louvain-la-Neuve, Belgium (vdooren@csam.ucl.ac.be).

Assuming the inverses and square roots exist, it is easy to algebraically verify that $H^H \Sigma H = \Sigma$ and that (2) holds with

$$\hat{A} = (I - (BA^\dagger)^H(BA^\dagger))^{1/2}A.$$

To see that the matrix square roots and inverses exist, note that for full rank A , $A^H A - B^H B > 0$ is equivalent to $A^H(I - (BA^\dagger)^H(BA^\dagger))A > 0$. This implies that $I - (BA^\dagger)^H(BA^\dagger) > 0$. To see why, let

$$x = Ax_0 + x_1,$$

where $x \neq 0$ is an arbitrary nonzero vector and $x_1^H A = 0$. Then

$$x^H (I - (BA^\dagger)^H(BA^\dagger)) x = x_0^H A^H (I - (BA^\dagger)^H(BA^\dagger)) A x_0 + x_1^H x_1 > 0.$$

It follows that $\|BA^\dagger\|_2 < 1$, which is sufficient to show that both matrix inverse square roots exist.

The inverse and conjugate transpose of a Σ -unitary matrix H are Σ -unitary. The inverse always exists and is given by $H^{-1} = \Sigma H^H \Sigma$. The product of Σ -unitary matrices can be shown to be Σ -unitary. It follows that Σ -unitary transformations form a multiplicative group. It is natural to decompose such matrices into a product of simpler Σ -unitary transformations. Typical choices are block diagonal unitary matrices of the form

$$\begin{pmatrix} U_A & 0 \\ 0 & U_B \end{pmatrix}$$

for unitary U_A and U_B , and *hyperbolic rotations*

$$\begin{pmatrix} I & & & & \\ & c_h & & \bar{s}_h & \\ & & I & & \\ & s_h & & c_h & \\ & & & & I \end{pmatrix},$$

where $|c_h|^2 - |s_h|^2 = 1$ and where the latter transformation acts on a single row of A together with a single row of B . In the case in which c_h is real and positive it is common to express c_h and s_h in terms of a single parameter by writing

$$c_h = \frac{1}{\sqrt{1 - |\rho|^2}}, \quad s_h = \frac{\rho}{\sqrt{1 - |\rho|^2}}.$$

Common algorithms for applying Σ -unitary matrices compute and apply H as a product of such elementary transformations, and the choice of c_h being real positive is not really restrictive. It can be shown that any Σ -unitary transformation can be represented as a product of block diagonal unitary matrices and hyperbolic rotations.

A common approach for computing H satisfying (2) as such a product follows a simple triangularization procedure. Introducing subscripts to keep track of different stages of the process, we suppose that we are given matrices A_1 and B_1 such that $A_1^H A_1 - B_1^H B_1 > 0$. We start by computing unitary U_1 and U_2 such that

$$\begin{pmatrix} U_1^H & 0 \\ 0 & U_2^H \end{pmatrix} \begin{pmatrix} A_1 \\ B_1 \end{pmatrix} = \begin{pmatrix} \hat{A}_1 \\ \hat{B}_1 \end{pmatrix},$$

where $\hat{A}_1$ and $\hat{B}_1$ are either upper triangular or at least have a first column of the form

$$\hat{A}_1 e_1 = \begin{pmatrix} a_{11} \\ 0 \end{pmatrix} \quad \text{and} \quad \hat{B}_1 e_1 = \begin{pmatrix} b_{11} \\ 0 \end{pmatrix}$$

for scalar a_{11} and b_{11} and where e_1 is a first standard basis vector. A procedure for introducing the zero elements in $\hat{A}_1$ and $\hat{B}_1$ is simply the well-known method of QR factorization via Givens rotations or Householder transformations.

For the next step of the procedure we note that

$$\frac{1}{\sqrt{1-|\rho|^2}} \begin{pmatrix} 1 & \bar{\rho} \\ \rho & 1 \end{pmatrix} \begin{pmatrix} a \\ b \end{pmatrix} = \begin{pmatrix} \text{sign}(a)\sqrt{|a|^2-|b|^2} \\ 0 \end{pmatrix}$$

for $\rho = -b/a$. Thus zeros can be introduced into $\hat{A}_1$ and $\hat{B}_1$ using hyperbolic rotations. If

$$\begin{pmatrix} \hat{A}_1 \\ \hat{B}_1 \end{pmatrix} = \begin{pmatrix} a_{11} & a_{12}^H \\ 0 & A_{22} \\ b_{11} & b_{12}^H \\ 0 & B_{22} \end{pmatrix},$$

then a hyperbolic rotation can be applied to transform $\hat{A}_1$ and $\hat{B}_1$ to A_2 and B_2 of the form

$$\begin{pmatrix} A_2 \\ B_2 \end{pmatrix} = \begin{pmatrix} \text{sign}(a_{11})\sqrt{|a_{11}|^2-|b_{11}|^2} & \tilde{a}_{12}^H \\ 0 & A_{22} \\ 0 & \tilde{b}_{12}^H \\ 0 & B_{22} \end{pmatrix}.$$

This process can be applied recursively to

$$\begin{pmatrix} A_{22} \\ b_{12}^H \\ B_{22} \end{pmatrix}$$

using transformations that are Σ -unitary with respect to $\Sigma = I_{p-1} \oplus -I_q$ to successively zero the columns of B . This particular process computes an $\hat{A}$ that is upper triangular although none of the results of this paper require that H be computed so that this is the case.

From this algorithm it follows that the condition $A^H A - B^H B > 0$ ensures that we can compute H satisfying (2) as a product

$$(4) \quad H = \begin{pmatrix} U_A & 0 \\ 0 & U_B \end{pmatrix} H^{(p)} U^{(p)} H^{(p-1)} U^{(p-1)} \dots H^{(1)} U^{(1)},$$

where the matrices $U^{(k)}$ and $H^{(k)}$ have the following form:

$$(5) \quad U^{(k)} = \left(\begin{array}{cc|c} I_{k-1} & 0 & 0 \\ 0 & U_1^{(k)} & 0 \\ \hline 0 & 0 & U_2^{(k)} \end{array} \right)$$

and

$$(6) \quad H^{(k)} = \left(\begin{array}{ccc|c} I_{k-1} & 0 & 0 & 0 \\ 0 & h_{11}^{(k)} & 0 & (h_{13}^{(k)})^H \\ 0 & 0 & I_{p-k} & 0 \\ 0 & h_{31}^{(k)} & 0 & H_{33}^{(k)} \end{array} \right).$$

The vertical lines in the partitions of $H^{(k)}$ and $U^{(k)}$ separate the matrices between columns p and $p+1$. The horizontal lines separate the matrices between rows p and $p+1$. This notation is applied consistently throughout this paper. Although (6) presents each $H^{(k)}$ as a relatively general Σ -unitary transformation, in most cases of computational interest each $H^{(k)}$ will be either a hyperbolic rotation or a hyperbolic Householder transformation. The unitary matrices U_A and U_B are not necessary for introducing zeros into the factorization (2). But if they are included it is not difficult to show, without any reference to A or B , that any Σ -unitary matrix admits a factorization of the form (4). Neither (4), (6), nor (2) suffice to uniquely define the factors $H^{(k)}$ and $U^{(k)}$. This is not a problem, however, since the results of this paper will apply to any factorization of a Σ -unitary matrix of the form (4).

Algorithmically the significance of this factorization is that it corresponds to a procedure for computing the rows of $\hat{A}$ one at a time. In later sections of this paper we will explore the mathematical aspects of this problem by introducing a hyperbolic version of the CS decomposition. This decomposition can be viewed as a special case of a factorization of H of the form (4). We will then show that out of all such factorizations the hyperbolic CS decomposition has optimality properties that parallel those of the direct rotation. In particular we will show that if H is of the form (4) and the $H^{(k)}$ are hyperbolic rotations with parameters $\rho^{(k)}$, then the parameters $\hat{\rho}^{(k)}$ revealed by the hyperbolic CS decomposition give a lower bound

$$\sum_{k=1}^p |\rho^{(k)}|^2 \geq \sum_k |\hat{\rho}^{(k)}|^2.$$

If both the $\rho^{(k)}$ and the $\hat{\rho}^{(k)}$ are in decreasing order, then

$$|\hat{\rho}^{(p)}| \leq |\rho^{(k)}| \leq |\hat{\rho}^{(1)}|.$$

Thus among all factorizations of a Σ -unitary matrix H into hyperbolic rotations the $\hat{\rho}^{(k)}$ are smallest in a 2-norm sense but the largest in an ∞ -norm sense. The parameters given by the hyperbolic CS decomposition give upper and lower bounds on hyperbolic rotations that can be used in the computational application of a general Σ -unitary H or to solve problem (2).

Regarding the relevance of these results, we note that problems that are naturally solved by application of a Σ -unitary transformation via hyperbolic rotations arise routinely in signal and image processing. Notable examples include the factorization of structured matrices using the generalized Schur algorithm [4] and the Cholesky downdating problem [3]. The results of this paper show that the size of parameters $\rho^{(k)}$ and hence $\|H^{(k)}\|$ are to a large extent determined by the problem. A different choice of a unitary transformation or a poor choice of ordering in applying hyperbolic rotations will never dramatically increase the norms of the transformations used. This has potential significance for both error analysis of algorithms and sensitivity analysis of problems involving downdating and structured matrix factorization.

It is also worth noting that most problems involving unitary transformations or orthogonal constraints have a natural, but often more difficult, analogue involving Σ -unitary transformations or Σ -unitary constraints. The algorithmic use of orthogonality constraints in a variety of ways is covered in [2]. The CS decomposition and its optimality have significant interpretations for such problems. A hyperbolic CS decomposition and its optimality properties should find natural application to problems involving Σ -orthogonality constraints.

An outline of this paper is as follows. In section 2 we introduce a decomposition of a partitioned Σ -unitary transformation that is analogous to the CS decomposition of a partitioned unitary matrix. In section 3 we prove the optimality properties of the hyperbolic CS decomposition. In section 4 we will briefly discuss analogous ideas for unitary matrices and contrast the results with the optimality of the direct rotation.

2. Decomposition of a partitioned Σ -unitary matrix. This section describes a hyperbolic CS decomposition. The decomposition is similar to the CS decomposition of a partitioned unitary matrix [7] except that it applies instead to a partitioned Σ -unitary matrix. The decomposition also appears in [5].

THEOREM 2.1 (hyperbolic CS decomposition). *Let H be Σ -unitary with $\Sigma = I_p \oplus -I_q$. If $q \geq p$, then H can be decomposed as*

$$(7) \quad H = \left(\begin{array}{c|c} U_A & 0 \\ \hline 0 & U_B \end{array} \right) \left(\begin{array}{c|c} D_A & (D_A^2 - I)^{1/2} \\ \hline (D_A^2 - I)^{1/2} & D_A \end{array} \begin{array}{c} 0 \\ 0 \end{array} \right) \left(\begin{array}{c|c} V_A^H & 0 \\ \hline 0 & V_B^H \end{array} \right),$$

where U_A , U_B , V_A , and V_B are unitary and D_A is diagonal. The matrices U_A , V_A , and D_A are $p \times p$.

If $q \leq p$, then

$$H = \left(\begin{array}{c|c} U_A & 0 \\ \hline 0 & U_B \end{array} \right) \left(\begin{array}{c|c} D_A & 0 \\ \hline 0 & I_{p-q} \end{array} \begin{array}{c} (D_A^2 - I)^{1/2} \\ 0 \end{array} \right) \left(\begin{array}{c|c} V_A^H & 0 \\ \hline 0 & V_B^H \end{array} \right).$$

The diagonal elements in D_A are real, positive, and greater than or equal to 1.

Proof. Suppose that $q \geq p$ and partition H as in (1). Let the singular value decomposition of the $p \times p$ block H_{11} be

$$H_{11} = U_A D_A V_A^H.$$

The relation $H^H \Sigma H = \Sigma$ implies that

$$D_A^2 - V_A^H H_{21}^H H_{21} V_A = I_p$$

so that $V_A^H H_{21}^H H_{21} V_A$ must be diagonal. Clearly the elements of D_A must be greater than one in magnitude and there must exist a unitary U_B such that

$$H_{21} = U_B \begin{pmatrix} (D_A^2 - I)^{1/2} \\ 0 \end{pmatrix} V_A^H.$$

This is just the singular value decomposition of H_{21} . Since $H \Sigma H^H = \Sigma$

$$D_A^2 - U_A^H H_{12} H_{12}^H U_A = I_p,$$

and clearly there is a unitary V_B such that

$$H_{12} = U_A \begin{pmatrix} (D_A^2 - I)^{1/2} & 0 \end{pmatrix} V_B^H.$$

The singular value decomposition of H_{12} , together with $H_{12}^H H_{12} - H_{22}^H H_{22} = -I_q$, implies

$$\begin{pmatrix} (D_A^2 - I) & 0 \\ 0 & 0 \end{pmatrix} - V_B^H H_{22}^H H_{22} V_B = -I_q.$$

Similarly $H_{21} H_{21}^H - H_{22} H_{22}^H = -I_q$ implies

$$\begin{pmatrix} (D_A^2 - I) & 0 \\ 0 & 0 \end{pmatrix} - U_B^H H_{22} H_{22}^H U_B = -I_q.$$

If we define $D_B = U_B^H H_{22} V_B$, then

$$D_B^H D_B = \begin{pmatrix} D_A^2 & 0 \\ 0 & I_{q-p} \end{pmatrix}, \quad D_B D_B^H = \begin{pmatrix} D_A^2 & 0 \\ 0 & I_{q-p} \end{pmatrix},$$

which imply that D_B is normal and hence diagonalizable by a unitary similarity. These relations also imply that

$$(8) \quad D_B \begin{pmatrix} D_A^2 & 0 \\ 0 & I_{q-p} \end{pmatrix} - \begin{pmatrix} D_A^2 & 0 \\ 0 & I_{q-p} \end{pmatrix} D_B = 0.$$

By writing out the matrix equation (8) element by element we get

$$\left(\left[\begin{pmatrix} D_A^2 & 0 \\ 0 & I_{q-p} \end{pmatrix} \right]_{jj} - \left[\begin{pmatrix} D_A^2 & 0 \\ 0 & I_{q-p} \end{pmatrix} \right]_{ii} \right) [D_B]_{ij} = 0.$$

From this, D_B can easily be seen to be block diagonal with the possibility of nontrivial (i.e., greater than 1×1) diagonal blocks only where the matrix $D_A^2 \oplus I_{q-p}$ has repeated diagonal elements. Since D_B is normal a unitary similarity can be applied to fully diagonalize D_B and, since the transformation can be chosen to act only on the blocks where there are repeated singular values, this transformation of U_B and V_B will result in matrices that are still composed of singular vectors for the other blocks H_{12} and H_{21} . Thus we can assume that U_B and V_B are chosen so that

$$H_{22} = U_B \begin{pmatrix} D_A & 0 \\ 0 & I_{q-p} \end{pmatrix} V_B^H.$$

Putting together the singular value decompositions of H_{11} , H_{12} , H_{21} , and H_{22} we get the required decomposition for $q \geq p$.

For $q \leq p$ we can obtain the result by application of the $q \geq p$ case to a matrix formed by permutation of the blocks of H ,

$$\begin{pmatrix} H_{22}^H & H_{12}^H \\ H_{21}^H & H_{11}^H \end{pmatrix} \begin{pmatrix} I_q & 0 \\ 0 & -I_p \end{pmatrix} \begin{pmatrix} H_{22} & H_{21} \\ H_{12} & H_{11} \end{pmatrix} = \begin{pmatrix} I_q & 0 \\ 0 & -I_p \end{pmatrix}. \quad \square$$

In section 1 we noted that the hyperbolic CS decomposition is a particular case of a decomposition of the form (4). To see this we define

$$D = (I - D_A^{-2})^{1/2}$$

to get the following form for (7):

$$(9) \quad H = \left(\begin{array}{c|c} U_A & 0 \\ \hline 0 & U_B \end{array} \right) \left(\begin{array}{c|c} \frac{(I-D^2)^{-1/2}}{D(I-D^2)^{-1/2}} & \frac{D(I-D^2)^{-1/2}}{(I-D^2)^{-1/2}} \frac{0}{0} \\ \hline 0 & I_{q-p} \end{array} \right) \left(\begin{array}{c|c} V_A^H & 0 \\ \hline 0 & V_B^H \end{array} \right).$$

If we define $\hat{\rho}^{(k)}$ by

$$D = \text{diag}(\hat{\rho}^{(1)}, \hat{\rho}^{(2)}, \dots, \hat{\rho}^{(p)}),$$

then (9) corresponds to a decomposition of H into a set of elementary hyperbolic rotations with real parameters $\hat{\rho}^{(k)}$. In particular, if we define

$$\hat{H}^{(k)} = \begin{pmatrix} I_{k-1} & 0 & 0 & 0 & 0 \\ 0 & \frac{1}{\sqrt{1-(\hat{\rho}^{(k)})^2}} & 0 & \frac{\hat{\rho}^{(k)}}{\sqrt{1-(\hat{\rho}^{(k)})^2}} & 0 \\ 0 & 0 & I_{p-1} & 0 & 0 \\ 0 & \frac{\hat{\rho}^{(k)}}{\sqrt{1-(\hat{\rho}^{(k)})^2}} & 0 & \frac{1}{\sqrt{1-(\hat{\rho}^{(k)})^2}} & 0 \\ 0 & 0 & 0 & 0 & I_{q-k} \end{pmatrix},$$

then

$$H = \left(\begin{array}{c|c} U_A & 0 \\ \hline 0 & U_B \end{array} \right) \hat{H}^{(p)} \hat{H}^{(p-1)} \dots \hat{H}^{(1)} \left(\begin{array}{c|c} V_A^H & 0 \\ \hline 0 & V_B^H \end{array} \right).$$

The right-hand side of this equation is clearly a factorization of the form (4). The decomposition is not difficult to adapt to the case in which $q < p$.

3. Optimality of the hyperbolic CS decomposition. In this section we will show that the hyperbolic CS decomposition is an optimal representation of a Σ -unitary matrix as a product of hyperbolic rotations. For a given H of the form (4), where the $H^{(k)}$ are hyperbolic rotations, the $\hat{\rho}^{(k)}$ in (9) are smallest in a sum-of-squares sense among all possible factorizations of the form (4). However, it will also be shown that among all factorizations of H into hyperbolic rotations, the hyperbolic CS decomposition produces the $\rho^{(k)}$ of largest magnitude.

To get the precise result we will first narrow the class of factorizations we must consider. Consider a transformation of the form (4). We can rewrite this as

$$H = \left(\begin{array}{c|c} U_A & 0 \\ \hline 0 & U_B \end{array} \right) H^{(p)} \left(\begin{array}{c|c} I_{p-1} & 0 \\ \hline 0 & u_1^{(p)} \end{array} \middle| \begin{array}{c} 0 \\ 0 \end{array} \right) \left(\begin{array}{c|c} I_p & 0 \\ \hline 0 & U_2^{(p)} \end{array} \right) \dots H^{(1)} \left(\begin{array}{c|c} U_1^{(1)} & 0 \\ \hline 0 & I_q \end{array} \right) \left(\begin{array}{c|c} I_p & 0 \\ \hline 0 & U_2^{(1)} \end{array} \right).$$

However, for $j < k$ the unitary transformation

$$\left(\begin{array}{c|c} I_{k-1} & 0 \\ \hline 0 & U_1^{(k)} \end{array} \middle| \begin{array}{c} 0 \\ 0 \end{array} \right)$$

commutes both with $H^{(j)}$ and with $I_p \oplus U_2^{(j)}$. Thus

$$H = \left(\begin{array}{c|c} U_A & 0 \\ \hline 0 & U_B \end{array} \right) H^{(p)} \left(\begin{array}{c|c} I_p & 0 \\ \hline 0 & U_2^{(p)} \end{array} \right) \dots H^{(1)} \left(\begin{array}{c|c} I_p & 0 \\ \hline 0 & U_2^{(1)} \end{array} \right) \left(\begin{array}{c|c} U_1 & 0 \\ \hline 0 & I_q \end{array} \right),$$

where

$$U_1 = U_1^{(p)} U_1^{(p-1)} \dots U_1^{(1)}.$$

Let

$$\tilde{H}^{(k)} = H^{(k)} \left(\begin{array}{c|c} I_p & 0 \\ \hline 0 & U_2^{(k)} \end{array} \right) = \left(\begin{array}{ccc|c} I_{k-1} & 0 & 0 & 0 \\ 0 & h_{11}^{(k)} & 0 & (h_{13}^{(k)})^H U_2^{(k)} \\ 0 & 0 & I_{p-k} & 0 \\ \hline 0 & h_{31}^{(k)} & 0 & H_{33}^{(k)} U_2^{(k)} \end{array} \right).$$

Clearly $\tilde{H}^{(k)}$ is a Σ -unitary transformation of the same general form as (6). Further, the quantity we will be concerned with in this section is $\|h_{31} h_{11}^{-1}\|_2$, which is invariant under the transformation by $U_2^{(k)}$. Thus we may assume without any loss of generality that H has the form

$$(10) \quad H = \left(\begin{array}{c|c} U_A & 0 \\ \hline 0 & U_B \end{array} \right) H^{(1)} H^{(2)} \dots H^{(p)} \left(\begin{array}{c|c} U_1 & 0 \\ \hline 0 & I_q \end{array} \right),$$

where each $H^{(k)}$ is of the form (6). We will state theorems concerning $\|h_{31}^{(k)} / h_{11}^{(k)}\|_2$, which apply to any $H^{(k)}$ in a factorization of the form (4), but when proving those theorems we will immediately assume that H is of the form (10).

Several intermediate results are required before we attempt to prove the main result of this section. First we need a theorem proven in [6]. The theorem is well known in several fields.

THEOREM 3.1 (unitary/ Σ -unitary correspondence). *Let*

$$H = \begin{pmatrix} H_{11} & H_{12} \\ H_{21} & H_{22} \end{pmatrix}$$

be Σ -unitary with $\Sigma = I_p \oplus -I_q$, where H_{11} is $p \times p$. Then H_{11} is nonsingular and the matrix

$$Q = \text{exc}(H) = \begin{pmatrix} H_{11}^{-1} & -H_{11}^{-1} H_{12} \\ H_{21} H_{11}^{-1} & H_{22} - H_{21} H_{11}^{-1} H_{12} \end{pmatrix}$$

is unitary. Conversely, if Q is unitary with a nonsingular leading principal submatrix of order p , then $\text{exc}(Q)$ is Σ -unitary. In the unitary case the leading principal submatrix is not guaranteed to be nonsingular.

The exchange theorem is necessary for the proof of the following lemma which, when applied recursively, will yield the main result of this section. In fact, the lemma may be interpreted as a block form of the main theorem applied to a product of just two matrices.

LEMMA 3.2. *Let G and H be Σ -unitary matrices of the form*

$$G = \left(\begin{array}{cc|c} G_{11} & 0 & G_{13} \\ 0 & I_{p-k_1} & 0 \\ \hline G_{31} & 0 & G_{33} \end{array} \right), \quad H = \left(\begin{array}{cc|c} I_{p-k_2} & 0 & 0 \\ 0 & H_{22} & H_{23} \\ \hline 0 & H_{32} & H_{33} \end{array} \right),$$

where G_{11} and H_{22} are $k_1 \times k_1$ and $k_2 \times k_2$, respectively, with $k_1 + k_2 = p$ and $\Sigma = I_p \oplus -I_q$. Let their product be

$$F = HG = \left(\begin{array}{c|c} F_{11} & F_{12} \\ \hline F_{21} & F_{22} \end{array} \right).$$

Let $\hat{\rho}^{(k)}$ for $k = 1, 2, \dots, p$ be the singular values of $F_{21}F_{11}^{-1}$ ordered from largest to smallest. If $p > q$, then we define $\hat{\rho}^{(k)} = 0$ for $k > q$. Then

$$\|F_{21}F_{11}^{-1}\|_F^2 = \sum_{k=1}^{\min(p,q)} (\hat{\rho}^{(k)})^2 \leq \|H_{32}H_{22}^{-1}\|_F^2 + \|G_{31}G_{11}^{-1}\|_F^2,$$

$$\hat{\rho}^{(p)} \leq \sigma_k(G_{31}G_{11}^{-1}) \leq \hat{\rho}^{(1)}$$

for $k = 1, 2, \dots, k_1$ and

$$\hat{\rho}^{(p)} \leq \sigma_k(H_{32}H_{22}^{-1}) \leq \hat{\rho}^{(1)}$$

for $k = 1, 2, \dots, k_2$.

Proof. The invertibility of G_{11} , H_{22} , and F_{11} follows from the previous theorem. Multiplying out HG we see that

$$F_{11} = \begin{pmatrix} G_{11} & 0 \\ H_{23}G_{31} & H_{22} \end{pmatrix}, \quad F_{21} = (H_{33}G_{31} \quad H_{32}).$$

Thus

$$F_{11}^{-1} = \begin{pmatrix} G_{11}^{-1} & 0 \\ -H_{22}^{-1}H_{23}G_{31}G_{11}^{-1} & H_{22}^{-1} \end{pmatrix}$$

and

$$F_{21}F_{11}^{-1} = ((H_{33} - H_{32}H_{22}^{-1}H_{23})G_{31}G_{11}^{-1} \quad H_{32}H_{22}^{-1})$$

so that

$$\begin{aligned} \|F_{21}F_{11}^{-1}\|_F^2 &= \|(H_{33} - H_{32}H_{22}^{-1}H_{23})G_{31}G_{11}^{-1}\|_F^2 + \|H_{32}H_{22}^{-1}\|_F^2 \\ &\leq \|H_{33} - H_{32}H_{22}^{-1}H_{23}\|_2^2 \|G_{31}G_{11}^{-1}\|_F^2 + \|H_{32}H_{22}^{-1}\|_F^2. \end{aligned}$$

However, since H is Σ -unitary the exchanged matrix $\text{exc}(H)$ is unitary. It is easily seen that $\text{exc}(H)$ contains $H_{33} - H_{32}H_{22}^{-1}H_{23}$ as a block. Consequently

$$\|H_{33} - H_{32}H_{22}^{-1}H_{23}\|_2 \leq \|\text{exc}(H)\|_2 = 1$$

from which the first inequality of the lemma follows.

The second inequality also follows from the expression for $F_{21}F_{11}^{-1}$. Clearly

$$\sigma_k(H_{32}H_{22}^{-1}) \leq \sigma_1(H_{32}H_{22}^{-1}) \leq \sigma_1((H_{33} - H_{32}H_{22}^{-1}H_{23})G_{31}G_{11}^{-1} \quad H_{32}H_{22}^{-1})$$

from which we get the upper bound. If $p \leq q$, then

$$\sigma_k(H_{32}H_{22}^{-1}) \geq \sigma_{k_2}(H_{32}H_{22}^{-1}) \geq \sigma_p((H_{33} - H_{32}H_{22}^{-1}H_{23})G_{31}G_{11}^{-1} \quad H_{32}H_{22}^{-1})$$

so that the lower bound holds. If $p > q$, then the lower bound is trivial, since $\sigma_p(F_{21}F_{11}^{-1}) = 0$.

To establish the third inequality we note that Theorem 2.1 implies that the singular values of $F_{11}^{-1}F_{12}$ are the same as those of $F_{21}F_{11}^{-1}$. The result then follows from

$$(11) \quad F_{11}^{-1}F_{12} = \begin{pmatrix} G_{11}^{-1}G_{13} \\ H_{22}^{-1}H_{23}(G_{33} - G_{31}G_{11}^{-1}G_{13}) \end{pmatrix}$$

through an argument identical to that used in verifying the second inequality. $\square$

Although they will not be used in this paper, more sophisticated results are possible in special cases. For example if $k_2 = 1$, then (11) implies that the singular values of $G_{11}^{-1}G_{13}$ interlace the values $\hat{\rho}^{(k)}$.

We now require that the $H^{(k)}$ in (4) be hyperbolic rotations and we extend the theorem to a product of p Σ -unitary transformations.

THEOREM 3.3. *Let H be Σ -unitary with $\Sigma = I_p \oplus -I_q$, $p \leq q$, and let*

$$H = \left(\begin{array}{c|c} U_A & 0 \\ \hline 0 & U_B \end{array} \right) H^{(p)} U^{(p)} H^{(p-1)} U^{(p-1)} \dots H^{(1)} U^{(1)},$$

where

$$H^{(k)} = \begin{pmatrix} I_{k-1} & 0 & 0 & 0 & 0 \\ 0 & \frac{1}{\sqrt{1-(\rho^{(k)})^2}} & 0 & \frac{\rho^{(k)}}{\sqrt{1-(\rho^{(k)})^2}} & 0 \\ 0 & 0 & I_{p-k+l-1} & 0 & 0 \\ 0 & \frac{\rho^{(k)}}{\sqrt{1-(\rho^{(k)})^2}} & 0 & \frac{1}{\sqrt{1-(\rho^{(k)})^2}} & 0 \\ 0 & 0 & 0 & 0 & I_{q-l} \end{pmatrix}$$

for arbitrary $1 \leq l \leq q$. Also let $U^{(k)}$ be a block diagonal unitary matrix of the form (5). Here U_A and U_B are arbitrary unitary matrices and do not necessarily correspond to the U_A and U_B in the hyperbolic CS decomposition. If $\hat{\rho}^{(k)}$ are the parameters from the hyperbolic CS decomposition of H arranged in decreasing order for $k = 1, 2, \dots, p$, then

$$(12) \quad \sum_k |\rho^{(k)}|^2 \geq \sum_k |\hat{\rho}^{(k)}|^2$$

and

$$|\hat{\rho}^{(p)}| \leq |\rho^{(k)}| \leq |\hat{\rho}^{(1)}|$$

for $k = 1, 2, \dots, p$.

Proof. Note that if we partition $H^{(k)}$ as in (6), then

$$h_{31}^{(k)} / h_{11}^{(k)} = \begin{pmatrix} 0 \\ \rho^{(k)} \\ 0 \end{pmatrix}$$

so that $\|h_{31}^{(k)} / h_{11}^{(k)}\|_2 = |\rho^{(k)}|$. As we argued at the beginning of this section, we can assume that H has the form (10). The $H^{(k)}$ in (10) are not the same as the $H^{(k)}$ in the statement of the theorem. The new $H^{(k)}$ are no longer elementary hyperbolic rotations but are instead general Σ -unitary transformations of the form (6). However, the quantity $\|h_{31}^{(k)} / h_{11}^{(k)}\|_2 = |\rho^{(k)}|$ is not changed. We will assume that H has the form (10) and prove the statements of the theorem for the quantities $\|h_{31}^{(k)} / h_{11}^{(k)}\|_2$ instead of for $|\rho^{(k)}|$.

Note that U_1 , U_A , and U_B do not change the parameters $\hat{\rho}^{(k)}$ so that without loss of generality we can assume that each of these matrices is the identity. Hence to prove (12) we must show that if H is Σ -unitary with $\Sigma = I_p \oplus -I_q$ and is factored as

$$(13) \quad H = H^{(p)} H^{(p-1)} \dots H^{(1)},$$

then

$$\|H_{21}H_{11}^{-1}\|_F^2 \leq \|h_{31}^{(p)}/h_{11}^{(p)}\|_2^2 + \|h_{31}^{(p-1)}/h_{11}^{(p-1)}\|_2^2 + \cdots + \|h_{31}^{(1)}/h_{11}^{(1)}\|_2^2.$$

The proof is by induction on p . The case $p = 1$ is trivial. Assume that if L is Σ -unitary with $\Sigma = I_{p-1} \oplus -I_q$ and

$$(14) \quad L = L^{(p-1)}L^{(p-2)} \cdots L^{(1)},$$

then

$$\|L_{21}L_{11}^{-1}\|_F^2 \leq \|l_{31}^{(p-1)}/l_{11}^{(p-1)}\|_2^2 + \|l_{31}^{(p-2)}/l_{11}^{(p-2)}\|_2^2 + \cdots + \|l_{31}^{(1)}/l_{11}^{(1)}\|_2^2,$$

where the matrices L and $L^{(k)}$ are partitioned in the same manner as H and $H^{(k)}$. Define

$$G = H^{(p-1)}H^{(p-2)} \cdots H^{(1)} = \left(\begin{array}{cc|c} G_{11} & 0 & G_{13} \\ 0 & 1 & 0 \\ \hline G_{31} & 0 & G_{33} \end{array} \right)$$

so that $H = H^{(p)}G$. We may apply Lemma 3.2 to conclude that

$$(15) \quad \|H_{21}H_{11}^{-1}\|_F^2 \leq \|h_{31}^{(p)}/h_{11}^{(p)}\|_2^2 + \|G_{31}G_{11}^{-1}\|_F^2.$$

Define

$$L^{(k)} = \left(\begin{array}{ccc|c} I_{k-1} & 0 & 0 & 0 \\ 0 & h_{11}^{(k)} & 0 & (h_{13}^{(k)})^H \\ 0 & 0 & I_{p-k-1} & 0 \\ \hline 0 & h_{31}^{(k)} & 0 & H_{33}^{(k)} \end{array} \right).$$

Then $L^{(k)}$ is Σ -unitary with respect to $I_{p-1} \oplus -I_q$ and

$$(16) \quad L = L^{(p-1)}L^{(p-2)} \cdots L^{(1)} = \left(\begin{array}{c|c} G_{11} & G_{13} \\ \hline G_{31} & G_{33} \end{array} \right).$$

Thus the induction hypothesis implies that

$$\|G_{31}G_{11}^{-1}\|_F^2 \leq \|h_{31}^{(p-1)}/h_{11}^{(p-1)}\|_2^2 + \|h_{31}^{(p-2)}/h_{11}^{(p-2)}\|_2^2 + \cdots + \|h_{31}^{(1)}/h_{11}^{(1)}\|_2^2.$$

With (15) this completes the induction step to prove (12).

The proof of the upper and lower bounds on $|\rho^{(k)}|$ is similar. We can assume that H is of the form (13) and we must then prove that

$$\hat{\rho}^{(p)} \leq \|h_{31}^{(k)}/h_{11}^{(k)}\|_2 \leq \hat{\rho}^{(1)}.$$

The proof is again inductive on p . The result is clearly true if $p = 1$. We assume that it is true for $p - 1$; i.e., if L is of the form (14) and partitioned as (16), then

$$\sigma_{p-1}(G_{31}G_{11}^{-1}) \leq \|h_{31}^{(k)}/h_{11}^{(k)}\|_2 \leq \sigma_1(G_{31}G_{11}^{-1}).$$

Now consider H . Since $H = H^{(p)}G$ Lemma 3.2 immediately implies that

$$\hat{\rho}^{(p)} \leq \|h_{31}^{(p)}/h_{11}^{(p)}\|_2 \leq \hat{\rho}^{(1)}$$

and that

$$\hat{\rho}^{(p)} \leq \sigma_k(G_{31}G_{11}^{-1}) \leq \hat{\rho}^{(1)}$$

so that the induction hypothesis implies

$$\hat{\rho}^{(p)} \leq \sigma_{p-1}(G_{31}G_{11}^{-1}) \leq \|h_{31}^{(k)}/h_{11}^{(k)}\| \leq \sigma_1(G_{31}G_{11}^{-1}) \leq \hat{\rho}^{(1)}$$

for $k = 1, 2, \dots, p-1$. $\square$

The theorem shows that in a sum-of-squares sense the parameters $\hat{\rho}^{(k)}$ are less than or equal to the $\rho^{(k)}$ associated with any factorization of H of the form (4) into elementary hyperbolic rotations. However, the theorem also shows that $\hat{\rho}^{(1)}$ is larger than any of the $\rho^{(k)}$. Thus, noting that a $\rho^{(k)}$ close to 1 represents a hyperbolic rotation with undesirably large norm, the theorem implies that the hyperbolic CS decomposition is the best possible factorization of H into rotations as measured by the 2-norm of the vector of $\rho^{(k)}$, but the worst as measured by the ∞ -norm. Unfortunately these results are difficult to interpret numerically.

If the $H^{(k)}$ are not hyperbolic rotations, then the theorem holds with the values $\rho^{(k)}$ replaced by $\|h_{31}^{(k)}/h_{11}^{(k)}\|_2$. This might be of computational interest if the $H^{(k)}$ are hyperbolic Householder transformations; see [6].

4. Unitary matrices. The results of the last section can be adapted to factorizations of unitary matrices. We start by stating the CS decomposition theorem for a partitioned unitary matrix [7].

THEOREM 4.1 (CS decomposition). *Let*

$$H = \begin{pmatrix} H_{11} & H_{12} \\ H_{21} & H_{22} \end{pmatrix}$$

be unitary with H_{11} $p \times p$ and H_{22} $q \times q$. If $q \geq p$, then H can be decomposed as

$$(17) \quad H = \left(\begin{array}{c|c} U_A & 0 \\ \hline 0 & U_B \end{array} \right) \left(\begin{array}{c|c|c} C & -\bar{S} & 0 \\ \hline \bar{S} & C & 0 \\ \hline 0 & 0 & I_{q-p} \end{array} \right) \left(\begin{array}{c|c} V_A^H & 0 \\ \hline 0 & V_B^H \end{array} \right),$$

where U_A , U_B , V_A , and V_B are unitary, C and S are diagonal with $|C|^2 + |S|^2 = I$, and C has positive, real diagonal elements.

If $q \leq p$, then

$$H = \left(\begin{array}{c|c} U_A & 0 \\ \hline 0 & U_B \end{array} \right) \left(\begin{array}{c|c|c} C & 0 & -\bar{S} \\ \hline 0 & I_{p-q} & 0 \\ \hline \bar{S} & 0 & C \end{array} \right) \left(\begin{array}{c|c} V_A^H & 0 \\ \hline 0 & V_B^H \end{array} \right).$$

The next step is to prove a result that is analogous to Lemma 3.2. In the unitary case the lemma has a very direct proof that does not depend on Theorem 3.1.

LEMMA 4.2. *Let G and H be unitary matrices of the form*

$$G = \left(\begin{array}{c|c|c} G_{11} & 0 & G_{13} \\ \hline 0 & I_{p-k_1} & 0 \\ \hline G_{31} & 0 & G_{33} \end{array} \right),$$

$$H = \left(\begin{array}{cc|c} I_{p-k_2} & 0 & 0 \\ 0 & H_{22} & H_{23} \\ \hline 0 & H_{32} & H_{33} \end{array} \right),$$

where G_{11} and H_{22} are $k_1 \times k_1$ and $k_2 \times k_2$, respectively. Let their product be

$$F = HG = \left(\begin{array}{c|c} F_{11} & F_{12} \\ \hline F_{21} & F_{22} \end{array} \right).$$

Let $\hat{s}^{(k)}$ for $k = 1, 2, \dots, p$ be the singular values of F_{21} ordered from largest to smallest. If $p > q$, then we define $s^{(k)} = 0$ for $k > q$. Then

$$\|F_{21}\|_F^2 = \sum_{k=1}^{\min(p,q)} |\hat{s}^{(k)}|^2 \leq \|H_{32}\|_F^2 + \|G_{31}\|_F^2,$$

$$\hat{s}^{(p)} \leq \sigma_k(G_{31}) \leq \hat{s}^{(1)}$$

for $k = 1, 2, \dots, k_1$ and

$$\hat{s}^{(p)} \leq \sigma_k(H_{32}) \leq \hat{s}^{(1)}$$

for $k = 1, 2, \dots, k_2$.

Proof. As in the proof of Lemma 3.2 we can multiply out HG to get

$$F_{21} = (H_{33}G_{31} \quad H_{32}).$$

The matrix G_{31} is a block of a unitary matrix so that $\|G_{31}\|_2 \leq 1$. This immediately yields the first inequality. The upper and lower bounds on the singular values of H_{32} and G_{31} also follow from the expression for F_{21} . $\square$

The consequences of the lemma are similar to Theorem 3.3 except the results are stated in terms of sines instead of the parameters $\hat{\rho}^{(k)}$. Given a unitary matrix H we define $\hat{s}^{(k)}$ by taking the CS decomposition and letting

$$S = \text{diag}(\hat{s}^{(1)}, \hat{s}^{(2)}, \dots, \hat{s}^{(p)}).$$

THEOREM 4.3. *Let H be unitary and let*

$$H = H^{(p)}U^{(p)}H^{(p-1)}U^{(p-1)}\dots H^{(1)}U^{(1)},$$

where

$$H^{(k)} = \begin{pmatrix} I_{k-1} & 0 & 0 & 0 & 0 \\ 0 & c^{(k)} & 0 & \overline{s^{(k)}} & 0 \\ 0 & 0 & I_{p-k+l-1} & 0 & 0 \\ 0 & s^{(k)} & 0 & c^{(k)} & 0 \\ 0 & 0 & 0 & 0 & I_{q-l} \end{pmatrix}$$

for arbitrary $1 \leq l \leq q$ and for $|c|^2 + |s|^2 = 1$. Also let $U^{(k)}$ be a block diagonal unitary matrix of the form (5). If $\hat{s}^{(k)}$ are the sines from the CS decomposition of H arranged in decreasing order by magnitude for $k = 1, 2, \dots, p$, then

$$(18) \quad \sum_k |s^{(k)}|^2 \geq \sum_k |\hat{s}^{(k)}|^2$$

and

$$(19) \quad \left| \hat{s}^{(p)} \right| \leq \left| s^{(k)} \right| \leq \left| \hat{s}^{(1)} \right|.$$

Proof. The inductive argument required to obtain this result from Lemma 4.2 is essentially identical to the argument used to obtain Theorem 3.3 from Lemma 3.2. $\square$

The unitary version of the theorem gives a lower bound on the sum of squares of sines in any factorization (4) of unitary H into elementary plane rotations. The bound is in terms of the sum of the squares of the sines obtained from the CS decomposition of H . At first glance this result appears to be similar to a well-known optimality result for the direct rotation [1].

In particular, let

$$U_1 = (u_1 \quad u_2 \quad \cdots \quad u_p), \quad V_1 = (v_1 \quad v_2 \quad \cdots \quad v_p)$$

be two orthonormal bases for two p -dimensional subspaces $\mathcal{U}$ and $\mathcal{V}$. Let U_2 and V_2 be orthonormal bases for the orthogonal complements of $\mathcal{U}$ and $\mathcal{V}$ and define

$$\hat{H} = \begin{pmatrix} U_1^H \\ U_2^H \end{pmatrix} (V_1 \quad V_2) = \begin{pmatrix} H_{11} & H_{12} \\ H_{21} & H_{22} \end{pmatrix} = \begin{pmatrix} U_A & 0 \\ 0 & U_B \end{pmatrix} \begin{pmatrix} C & -\bar{S} & 0 \\ \bar{S} & C & 0 \\ 0 & 0 & I \end{pmatrix} \begin{pmatrix} V_A^H & 0 \\ 0 & V_B^H \end{pmatrix},$$

where $S = \text{diag}(\hat{s}^{(1)}, \hat{s}^{(2)}, \dots, \hat{s}^{(p)})$ and where for simplicity we have assumed that $p < q$ in partitioning the CS decomposition of $\hat{H}$. If H is any unitary transformation mapping $\mathcal{V}$ to $\mathcal{U}$ and we define

$$(20) \quad s^{(k)} = \sin(\angle(v_k, Hv_k)),$$

then

$$(21) \quad \sum_{k=1}^p \left| s^{(k)} \right|^2 \geq \sum_{k=1}^p \left| \hat{s}^{(k)} \right|^2.$$

There is an optimal H , known as the direct rotation, which achieves equality in (21) [1].

Since both involve a lower bound in terms of $\sum_k |\hat{s}^{(k)}|^2$, it is worth the effort to contrast the optimality of the direct rotation with Theorem 4.3 to make sure that they are truly distinct results and not expressions in different languages of the same underlying fact. We start with the obvious: Theorem 4.3 gives bounds on sines that can be used in factoring a given unitary matrix H ; the optimality of the direct rotation introduces $\hat{H}$ and defines the sines associated with $\hat{H}$ not through a matrix factorization but through (20). The quantities $s^{(k)}$ appear to be defined in two distinct ways.

In addition to their different definitions, the two sets of sines display different behavior; the $s^{(k)}$ defined by (20) may satisfy (21) but they do not in general satisfy (19). This can be shown by considering a simple example.

Example 1. Let

$$U_1 = V_1 = \begin{pmatrix} I \\ 0 \end{pmatrix}, \quad U_2 = V_2 = \begin{pmatrix} 0 \\ I \end{pmatrix}.$$

Then all H mapping $\mathcal{U}$ to $\mathcal{V}$ are of the form

$$H = \begin{pmatrix} X & 0 \\ 0 & Y \end{pmatrix}$$

for unitary X and Y and

$$\begin{aligned} |s^{(k)}|^2 &= |\sin(\angle(v_k, H v_k))|^2 = 1 - |\cos(\angle(v_k, H v_k))|^2 = 1 - |v_k^H H v_k|^2 \\ &= 1 - |e_k^H H e_k|^2 = 1 - |[X]_{kk}|^2. \end{aligned}$$

The matrix $\hat{H}$ defined by the bases for $\mathcal{U}$ and $\mathcal{V}$ is just the identity and is already reduced to its CS decomposition. So $\hat{s}^{(k)} = 0$ for each k . Thus (19), if it held, would imply that $|s^{(k)}| = 0$. However, $[X]_{kk}$ can be chosen so that $|s^{(k)}|$ is any value between 0 and 1 and therefore (19) cannot hold for all possible choices of H .

REFERENCES

- [1] C. DAVIS AND W. M. KAHAN, *The rotation of eigenvectors by a perturbation. III*, SIAM J. Numer. Anal., 7 (1970), pp. 1–46.
- [2] A. EDELMAN, T. A. ARIAS, AND S. T. SMITH, *The geometry of algorithms with orthogonality constraints*, SIAM J. Matrix Anal. Appl., 20 (1998), pp. 303–353.
- [3] L. ELDÉN AND H. PARK, *Block downdating of least squares solutions*, SIAM J. Matrix Anal. Appl., 15 (1994), pp. 1018–1034.
- [4] K. GALLIVAN, S. THIRUMALAI, P. VAN DOOREN, AND V. VERMAUT, *High performance algorithms for Toeplitz and block Toeplitz matrices*, Linear Algebra Appl., 243 (1996), pp. 343–388.
- [5] E. J. GRIMME, D. C. SORENSON, AND P. VAN DOOREN, *Model reduction of state space systems via an implicitly restarted Lanczos method*, Numer. Algorithms, 12 (1995), pp. 1–31.
- [6] M. STEWART AND G. W. STEWART, *On hyperbolic triangularization: Stability and pivoting*, SIAM J. Matrix Anal. Appl., 19 (1998), pp. 847–860.
- [7] G. W. STEWART AND J.-G. SUN, *Matrix Perturbation Theory*, Academic Press, Boston, 1990.

ON STABILIZATION AND CONVERGENCE OF CLUSTERED RITZ VALUES IN THE LANCZOS METHOD*

WOLFGANG WÜLLING†

Abstract. The Lanczos algorithm can be considered as an iterative method for finding a few eigenvalues and eigenvectors of large sparse symmetric matrices. In the present paper we solve a conjecture posed by Zdeněk Strakoš and Anne Greenbaum in [*Open Questions in the Convergence Analysis of the Lanczos Process for the Real Symmetric Eigenvalue Problem*, IMA Research Report, 1992] on the clustering of Ritz values, which occurs in finite precision computations. In particular, we prove that the conjecture is valid in most cases and describe the rare case when it is not. The established upper bounds measuring the quality of Ritz approximations imply also that Ritz values cluster only close to an eigenvalue.

Key words. eigenvalues, Ritz values, Lanczos algorithm, tridiagonal matrix

AMS subject classifications. 65F15, 65F50

DOI. 10.1137/040608908

1. Introduction. Consider the eigenvalue problem

$$Au = \lambda u$$

for a real symmetric $N \times N$ matrix A . If A is a large, typically sparse matrix and if approximations to a few eigenpairs (λ, u) of A are sought, the Lanczos method [7] (or one of its variants; see, e.g., [16], [20, p. 391 ff], [21]) is the method of choice. It is iterative in nature and produces at the k th step a Jacobi matrix T_k of order k . The eigenvalues $\theta_j^{(k)}$, $1 \leq j \leq k$, of T_k , which are the Ritz values in the context of the Lanczos algorithm, are then used as approximations to a few eigenvalues of A .

One of the attractive features of the Lanczos method is that a quantity which measures the quality of Ritz approximations is at hand during the Lanczos run at almost no additional cost: Given a Ritz value $\theta_j^{(k)}$ and a Ritz vector $z_j^{(k)}$ as approximations to an eigenpair of A , the residual norm $\|Az_j^{(k)} - \theta_j^{(k)}z_j^{(k)}\|$ and also the distance of $\theta_j^{(k)}$ to the nearest eigenvalue of A can be controlled in exact and finite precision arithmetic computations by an easily computable quantity $\delta_{k,j}$ (its definition will be given in (2.4)); see [10].

Implemented on a computer, it happens that Ritz values cluster around eigenvalues. Since the number $\delta_{k,j}$ gives information about the convergence of Ritz approximations and in accordance with numerical observations, Zdeněk Strakoš and Anne Greenbaum formulated in [18] the hypothesis that the number $\delta_{k,j}$ is small for any Ritz value in a tight, well separated cluster. Furthermore, part of the hypothesis is that the size of $\delta_{k,j}$ depends mainly on the length of two intervals: the cluster diameter and a gap in the distribution of the Ritz values.

An idea how to prove the conjecture was mentioned in [18], and that is to bound $\delta_{k,j}$ using only information about Ritz values in several consecutive steps of the Lanczos method. In this paper we will take up that idea and demonstrate how it can be

*Received by the editors May 24, 2004; accepted for publication (in revised form) by Z. Strakoš September 27, 2005; published electronically January 27, 2006.

<http://www.siam.org/journals/simax/27-3/60890.html>

†Universität Bielefeld, Fakultät für Mathematik, Postfach 100131, D-33501 Bielefeld, Germany (wuelling@math.uni-bielefeld.de).

used in order to prove that the conjecture from [18] is true with only one exception of an alternating number of Ritz values in a cluster. For the exceptional case an example will be constructed in section 3.1.

Greenbaum used the quantities $\delta_{k,j}$ in her backward analysis (see [4]) of the Lanczos process to distinguish between converged and unconverged clusters of Ritz values. Now, we will prove that any cluster of Ritz values has to be stabilized. Following Paige's definition (see [10]) we call a cluster of Ritz values computed at the k th step stabilized if in all subsequent iterations of the Lanczos method at least one Ritz value will be found sufficiently close to that cluster.

The analysis in this article will be done mainly by exploiting so-called eigenvector-eigenvalue relations for the Jacobi matrices T_k . They were developed by Paige (see [8, p. 106]) using a formula for the adjugate matrix of a general Hermitian matrix (see [19]).

Though not directly related to this work, we mention that in several other works that deal with a tight cluster of eigenvalues of (symmetric) tridiagonal matrices the quantity $\delta_{k,j}$ and the corresponding eigenvector-eigenvalue relations play a key role in the analysis (see, e.g., [12], [13], [15], [22]). Furthermore, analogous formulae for (upper) Hessenberg matrices can be found in [24] and also in [23], where several Krylov subspace methods in finite precision computations are studied using eigenvector-eigenvalue relations of upper Hessenberg matrices.

The paper is organized as follows. In section 2 we introduce the Lanczos algorithm and we repeat some of its properties in exact and finite precision computations needed for our purposes. We quote the conjecture from [18] in section 3; it will be illustrated with numerical examples. We then state and prove our main results in section 4.

We use the symbol $\mathbb{R}$ to denote the real numbers. The real part and imaginary part of a complex number $z \in \mathbb{C}$ will be represented by $\Re(z)$ and $\Im(z)$, respectively; the imaginary unit is i .

We refer to the symbol I as the identity matrix and we use e_j for the j th column of I . The conjugate transpose of a matrix (or vector) B is B^* , and, finally, $\|\cdot\|$ is the Euclidean vector and corresponding matrix norm. The spread, i.e., the distance between the smallest and largest Ritz value at the k th step, will be denoted by $\sigma_k = \theta_k^{(k)} - \theta_1^{(k)}$. The numerical experiments were carried out on a PC using the software MATLAB.¹ We denote the unit round off with $\epsilon \approx 1.1102 \times 10^{-16}$. Two expressions are used frequently throughout the text: the cluster size, which is the number of Ritz values that lie in a certain interval, and the cluster diameter, which is the length of this interval.

2. The Lanczos algorithm. Let $A \in \mathbb{R}^{N \times N}$ be a real symmetric $N \times N$ matrix with simple eigenvalues λ_j and orthonormal eigenvectors u_j : $Au_j = \lambda_j u_j$, $1 \leq j \leq N$. The eigenvalues will be numbered in increasing order, $\lambda_1 < \lambda_2 < \dots < \lambda_N$,

$$A = U\Lambda U^*, \quad UU^* = U^*U = I, \quad \Lambda = \text{diag}(\lambda_1, \dots, \lambda_N).$$

An orthogonal projection method onto a subspace $\mathcal{K} \subset \mathbb{R}^N$, like the Rayleigh–Ritz procedure (see, e.g., [11, p. 209 ff]), seeks approximations (θ, z) to an eigenpair (λ, u) by imposing the following Galerkin condition on the approximations $\theta \in \mathbb{R}$, $z \in \mathcal{K}$, $z \neq 0$:

$$Az - \theta z \perp \mathcal{K}.$$

¹MATLAB is a registered trademark of The MathWorks, Inc.

The number θ will be called the Ritz value and z is the corresponding Ritz vector. The Lanczos method is the Rayleigh–Ritz procedure applied to the Krylov subspace

$$\mathcal{K} = \mathcal{K}_k(A, q) = \text{span} \{q, Aq, A^2q, \dots, A^{k-1}q\}.$$

If $q_1 \in \mathbb{R}^N$ is a given vector such that $q_1^* q_1 = 1$ and if we put $q_0 = 0 \in \mathbb{R}^N$ as well as $\beta_1 = 0$, the Lanczos algorithm constructs successively orthonormal basis vectors q_{j+1} , $1 \leq j \leq k-1$, of the Krylov subspace $\mathcal{K}_k(A, q_1)$ and is given (in Paige's so-called variant A1; see [10]) by the following iteration scheme.

ALGORITHM 2.1 (Lanczos algorithm).

$$\begin{aligned} \text{For } j = 1, 2, \dots \\ \alpha_j &= q_j^* (Aq_j - \beta_j q_{j-1}), \\ \tilde{q}_{j+1} &= Aq_j - \alpha_j q_j - \beta_j q_{j-1}, \\ \beta_{j+1} &= \|\tilde{q}_{j+1}\|, \\ q_{j+1} &= \frac{\tilde{q}_{j+1}}{\beta_{j+1}}. \end{aligned}$$

The computations of the Lanczos algorithm can be written elegantly as two matrix equations. For this reason we define the matrix Q_k from the Lanczos vectors as

$$Q_k = [q_1, q_2, \dots, q_k] \in \mathbb{R}^{N \times k},$$

and the coefficients α_j and β_j are collected to build the Lanczos matrix, which is a Jacobi matrix, i.e., a symmetric tridiagonal matrix where the subdiagonal entries are positive:

$$T_k = \begin{bmatrix} \alpha_1 & \beta_2 & & & \\ \beta_2 & \alpha_2 & \beta_3 & & \\ & \ddots & \ddots & \ddots & \\ & & \ddots & \ddots & \beta_k \\ & & & \beta_k & \alpha_k \end{bmatrix}, \quad \beta_j > 0.$$

Doing so we obtain

$$(2.1) \quad AQ_k = Q_k T_k + \beta_{k+1} q_{k+1} e_k^*,$$

$$(2.2) \quad Q_k^* Q_k = I_k.$$

The spectral decomposition of T_k is

$$T_k = S_k \Theta_k S_k^*, \quad S_k^* S_k = S_k S_k^* = I_k,$$

$$S_k = [s_1^{(k)}, \dots, s_k^{(k)}], \quad \Theta_k = \text{diag}(\theta_1^{(k)}, \dots, \theta_k^{(k)}),$$

and the eigenvalues of T_k —which will be simple because T_k is unreduced, i.e., $\beta_2 \beta_3 \dots \beta_k \neq 0$ —are also ordered increasingly, $\theta_1^{(k)} < \theta_2^{(k)} < \dots < \theta_k^{(k)}$. Since the columns of Q_k span the Krylov subspace $\mathcal{K}_k(A, q_1)$ and because $T_k = Q_k^* A Q_k$ holds, it can be checked easily that T_k 's eigensystem determines the Rayleigh–Ritz approximations to

A 's eigenpairs. The eigenvalues of T_k are the Ritz values and the corresponding Ritz vector $z_j^{(k)} \in \mathcal{K}_k(A, q_1)$ to each Ritz value $\theta_j^{(k)}$ is given by $z_j^{(k)} = Q_k s_j^{(k)}$, $1 \leq j \leq k$.

In exact arithmetic computations Algorithm 2.1 will terminate with a $\beta_d = 0$, where $d \leq N + 1$. A $\beta_d = 0$ means that the Krylov subspace $\mathcal{K}_{d-1}(A, q_1)$ is invariant under A and hence every eigenvalue of T_{d-1} is an eigenvalue of A . Another nice feature of the Lanczos method in exact arithmetic is the possibility to measure the quality of Ritz approximations without much effort. It is only necessary to postmultiply (2.1) by $s_j^{(k)}$, since we obtain together with $T_k s_j^{(k)} = \theta_j^{(k)} s_j^{(k)}$ (see [11, p. 69])

$$(2.3) \quad \min_{1 \leq n \leq N} |\lambda_n - \theta_j^{(k)}| \leq \frac{\|A z_j^{(k)} - \theta_j^{(k)} z_j^{(k)}\|}{\|z_j^{(k)}\|} = \beta_{k+1} |s_{k,j}^{(k)}|.$$

As we can see the numbers $|s_{k,j}^{(k)}|$, i.e., the last row $e_k^* S_k$ of T_k 's eigenvector matrix S_k , play an important role in the (analysis of the) Lanczos algorithm: it allows one to measure the quality of Ritz approximations before computing explicitly the Ritz vectors. For any Ritz pair at the k th step of the Lanczos method it is therefore usual (see, e.g., [4], [10], [14], [18]) to define the numbers

$$(2.4) \quad \delta_{k,j} := \beta_{k+1} |s_{k,j}^{(k)}|, \quad 1 \leq j \leq k.$$

According to (2.3) we will find an eigenvalue of A in an interval of length $2\delta_{k,j}$ centered at the Ritz value $\theta_j^{(k)}$. If there is more information on the separation of eigenvalues and Ritz values a so-called *gap*-theorem (see [11, p. 222]) gives us

$$|\lambda_n - \theta_j^{(k)}| = \min_{\lambda \in \Lambda(A)} |\lambda - \theta_j^{(k)}| \leq \frac{(\delta_{k,j})^2}{\gamma},$$

where $\gamma = \min_{\lambda \neq \lambda_n} |\lambda - \theta_j^{(k)}|$ is the separation of $\theta_j^{(k)}$ to A 's eigenvalues different from λ_n .

2.1. Finite precision arithmetic. In finite precision arithmetic computations the property of having orthogonal Lanczos vectors is lost even for a very small k (in comparison to N) due to rounding errors. That is why the relationship between A and T_k is not clear a priori. But Paige showed (see [9] and [10]) that there is structure in the way the Lanczos vectors lose their orthogonality. We will give a short description of Paige's analysis. In finite precision arithmetic one has to replace (2.1) and (2.2) by²

$$\begin{aligned} A Q_k &= Q_k T_k + \beta_{k+1} q_{k+1} e_k^* + F_k, \\ Q_k^* Q_k &= C_k^* + \Delta_k + C_k, \end{aligned}$$

where C_k is a strict upper triangular matrix and $\Delta_k = \text{diag}(q_1^* q_1, \dots, q_k^* q_k)$. Let ϵ_1 , ϵ_2 , and η_k ³ be small multiples of the unit round off ϵ . Under reasonable assumptions⁴

² Q_k , T_k , β_{k+1} , and q_{k+1} are obtained by computations in finite precision arithmetic. In order to distinguish them from exact quantities, it is common (see, e.g., [2] and [3]) to denote them with a hat: $\hat{Q}_k$, $\hat{T}_k$ etc. We will move on without this notation here.

³The exact values: $\epsilon_0 = 2(N+4)\epsilon$, $\epsilon_1 = 2(7+m \frac{\|A\|}{\|A\|})\epsilon$, $\epsilon_2 = \sqrt{2} \max\{6\epsilon_0, \epsilon_1\}$, $\eta_k = k^{2.5} \|A\| \epsilon_2$, where m is the maximum number of nonzero elements in the rows of A .

⁴It is assumed that $\epsilon_0 < \frac{1}{12}$, $k(3\epsilon_0 + \epsilon_1) < 1$, and higher order terms in ϵ are ignored.

Paige proved (see [9] and [10]) that the rounding error matrix $F_k \in \mathbb{R}^{N \times k}$ satisfies

$$\|F_k\| \leq \sqrt{k}\|A\|\epsilon_1,$$

and the Ritz values are enclosed essentially by A 's eigenvalues:

$$\lambda_1 - \eta_k \leq \theta_j^{(k)} \leq \lambda_N + \eta_k, \quad j = 1, \dots, k.$$

The tridiagonal matrix T_k is the result of computations carried out in finite precision arithmetic, but it is still assumed that the spectral decomposition of T_k is exact. The numbers $\delta_{k,j}$ defined above in (2.4) are also important to estimate the Ritz approximations' quality in finite precision arithmetic. If we apply—similar to (2.3)—the simple error bound from [11, p. 69], we obtain $\min_{1 \leq l \leq N} |\lambda_l - \theta_j^{(k)}| \leq \|z_j^{(k)}\|^{-1} (\delta_{k,j} + \sqrt{k}\|A\|\epsilon_1)$. Thus a small $\delta_{k,j}$ guarantees a satisfactory bound for the convergence of a Ritz value $\theta_j^{(k)}$ to an eigenvalue of A only if the length of the corresponding Ritz vector is approximately 1, $\|z_j^{(k)}\| \approx 1$. But in finite precision computations it can happen (see, e.g., [10]) that $\|z_j^{(k)}\| \ll 1$. The following theorem from [18] develops a bound that does not involve the Ritz vector's length. It summarizes some results from the two papers [9] and [10].

THEOREM 2.1 (Paige). *For any Ritz pair $(\theta_j^{(k)}, z_j^{(k)})$ determined at the k th step of the Lanczos method the following holds true:*

$$(2.5) \quad \min_{1 \leq l \leq N} |\lambda_l - \theta_j^{(k)}| \leq \max \left\{ 2.5 \left(\delta_{k,j} + \sqrt{k}\|A\|\epsilon_1 \right), (k+1)^3 \|A\|\epsilon_2 \right\},$$

$$(2.6) \quad \|z_j^{(k)} - \xi_{l,j}^{(k)} u_l\| \leq \frac{\delta_{k,j} + \sqrt{k}\|A\|\epsilon_1}{\min_{r \neq l} |\lambda_r - \theta_j^{(k)}|}, \quad \text{where } \xi_{l,j}^{(k)} = u_l^* z_j^{(k)}, \quad 1 \leq l \leq N,$$

$$(2.7) \quad |q_{k+1}^* z_j^{(k)}| = \frac{|\epsilon_{jj}^{(k)}|}{\delta_{k,j}}, \quad |\epsilon_{jj}^{(k)}| \leq k\|A\|\epsilon_2, \quad 1 \leq j \leq k.$$

Thus inequality (2.5) shows that a very small value $\delta_{k,j}$ implies convergence of a Ritz value to an eigenvalue λ_l . Furthermore inequality (2.6) assures that the corresponding Ritz vector $z_j^{(k)}$ is a good approximation to the eigenvector u_l if, in addition, λ_l is well separated from the rest of A 's spectrum.

Equation (2.7) displays the aforementioned relation between the loss of orthogonality and convergence. If the latest computed Lanczos vector q_{k+1} loses orthogonality against the Ritz vector $z_j^{(k)}$ the number $\delta_{k,j}$ has to be small, which means that an eigenvalue is well approximated by $\theta_j^{(k)}$. But the reverse statement is not true: If $\delta_{k,j}$ is very small it can still be that $|\epsilon_{jj}^{(k)}| \ll \delta_{k,j}$ (or even $|\epsilon_{jj}^{(k)}| \approx 0$). That is why a small $\delta_{k,j}$ does not imply necessarily that $|q_{k+1}^* z_j^{(k)}|$ is significantly larger than zero. Moreover it is theoretically possible that a Ritz value converges to an eigenvalue and the orthogonality of the Lanczos vectors is perfectly preserved.

3. Clustered Ritz values. In finite precision computations it can be observed that Ritz values cluster around eigenvalues (see, e.g., [4], [10], [14]): Orthogonality of the Lanczos vectors is lost and multiple copies of Ritz values are computed.

Since, according to Theorem 2.1, a very small $\delta_{k,j}$ guarantees convergence of a Ritz value toward an eigenvalue, the question arises if it is also true that $\delta_{k,j} \ll 1$ holds for any Ritz value in a tight, well separated cluster? Using the upper bounds from [18, section 5], this question can be answered positively for some but not for all clusters. In particular the case of a cluster with alternating cluster size seems to be problematical. The conjecture from [18] is as follows

CONJECTURE 1 (Strakoš, Greenbaum). *For any Ritz value $\theta_j^{(k)}$ in a tight, well separated cluster of Ritz values the value $\delta_{k,j}$ is small and consequently the corresponding Ritz vector $z_j^{(k)}$ (if nonvanishing) is a good approximation to some eigenvector of the matrix A .*

Furthermore, Strakoš and Greenbaum supposed that the value of $\delta_{k,j}$ depends on the relative position of the Ritz values: If we have a cluster of $c \geq 2$ Ritz values at the k th step of the Lanczos method, $\theta_{n+1}^{(k)}, \dots, \theta_{n+c}^{(k)}$ say, then the idea from [18] is that $\delta_{k,j}$, $j = n+1, \dots, n+c$, is proportional to

$$(3.1) \quad \sqrt{\frac{\delta}{\gamma}},$$

where $\delta = \theta_{n+c}^{(k)} - \theta_{n+1}^{(k)}$ is the cluster diameter and the gap in the spectrum is $\gamma = \min \left\{ \theta_{n+1}^{(k)} - \theta_n^{(k)}, \theta_{n+c+1}^{(k)} - \theta_{n+c}^{(k)} \right\}$. Experiments with matrices that possess the following eigenvalue distribution (see [17]) underlined the validity of Conjecture 1 and the guessed bound (3.1):

$$(3.2) \quad \lambda_1 = 0.1, \quad \lambda_N = 100, \\ \lambda_\nu = \lambda_1 + \frac{\nu-1}{N-1} (\lambda_N - \lambda_1) \rho^{N-\nu}, \quad \nu = 2, \dots, N-1.$$

Applying the Lanczos algorithm to (diagonal) matrices with eigenvalues (3.2) and randomly generated starting vector q_1 , the orthogonality of the Lanczos vectors is lost very quickly for some values of ρ , ($0.6 \leq \rho \leq 0.8$), even for a very small dimension N (see [5], [17], [18]). We repeat an experiment from [18].

Example 1 (Strakoš). Apply the Lanczos algorithm to a diagonal matrix A with eigenvalues (3.2) where $N = 24$, $\rho = 0.7$, and a randomly generated starting vector q_1 is used. We look at the cluster of two Ritz values near $\lambda_{24} = 100$ (see Figure 3.1), and we observe the cluster of two Ritz values near $\lambda_{22} \approx 44.7944$ (see Figure 3.2).

The typical behavior shown by both figures suggests that whenever a new Ritz value joins a cluster, it takes some steps of the Lanczos method such that the quantities $\delta_{k,j}$ defined in (2.4) drop down to the conjectured bound (3.1) for all Ritz copies in the cluster. It has been pointed out in [18] that the fact that (3.1) is larger than the numbers $\delta_{k,j}$ at some steps of the Lanczos process is not a contradiction to the conjectured bound, because the maximal ratio of $\delta_{k,j}$ (or $\delta_{k,j-1}$) to $\sqrt{\frac{\delta}{\gamma}}$ is of order $\|A\| = 100$, such that it is still possible that $\delta_{k,j}$ is of order $\mathcal{O}(\|A\| \sqrt{\frac{\delta}{\gamma}})$ for all Ritz values $\theta_j^{(k)}$ in a tight, well separated cluster.

3.1. A counterexample. The next example is designed to illustrate the only exception where $\delta_{k,j}$ is not small for all Ritz values in a tight, well separated cluster. It

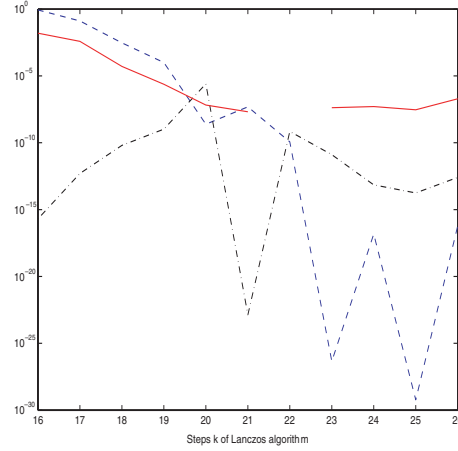


FIG. 3.1. The figure shows the case where two Ritz values approximate the simple eigenvalue $\lambda_{24} = 100$ at the steps $k = 16, \dots, 26$. The dash-dotted line displays the number $\delta_{k,k}$ for the first Ritz approximation $\theta_k^{(k)}$ and the dashed line shows the value of $\delta_{k,k-1}$ for the second Ritz approximation $\theta_{k-1}^{(k)}$. The solid line plots the value of (3.1) as conjectured. At step $k = 22$ the two computed Ritz approximations were numerically indistinguishable.

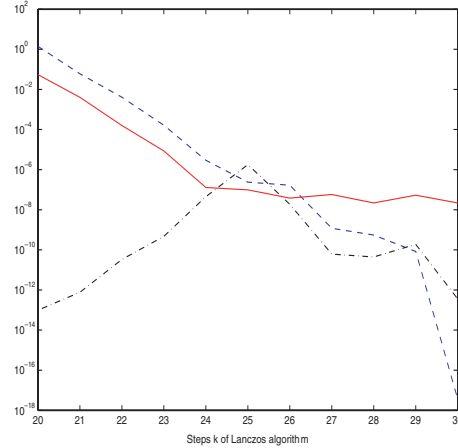


FIG. 3.2. This figure illustrates the case where two Ritz values approximate the eigenvalue $\lambda_{22} \approx 44.7944$; a second Ritz approximation emerged at step $k = 20$ for the first time. The figure is similar to Figure 3.1: Again, the dash-dotted line shows the value of $\delta_{k,j}$ for the first Ritz approximation, the dashed line displays the value of $\delta_{k,j-1}$ for the second one and the solid line shows (3.1).

will be proved in the next section that this example is the only case where Conjecture 1 is not true. Let N be an odd number and suppose that we have a diagonal matrix

$$(3.3) \quad A = \text{diag} \left(\lambda_1, \dots, \lambda_{\frac{N-1}{2}}, 0, -\lambda_1, \dots, -\lambda_{\frac{N-1}{2}} \right) \in \mathbb{R}^{N \times N}$$

and vectors with the following structure:

$$(3.4) \quad x = \left(x_1, x_2, \dots, x_{\frac{N-1}{2}}, x_{\frac{N+1}{2}}, -x_1, -x_2, \dots, -x_{\frac{N-1}{2}} \right)^* \in \mathbb{R}^N,$$

$$(3.5) \quad y = \left(y_1, y_2, \dots, y_{\frac{N-1}{2}}, y_{\frac{N+1}{2}}, y_1, y_2, \dots, y_{\frac{N-1}{2}} \right)^* \in \mathbb{R}^N.$$

TABLE 3.1

k	β_{k+1}	$ s_{k,j}^{(k)} $	$ s_{k,j+1}^{(k)} $	$\sqrt{\frac{\delta}{\gamma}}$
9	0.5164	9.0268×10^{-07}	-	-
10	99.4996	7.0710×10^{-01}	7.0710×10^{-01}	9.7023×10^{-05}
11	0.4924	4.6849×10^{-09}	-	-
12	99.4995	7.0710×10^{-01}	7.0710×10^{-01}	6.8258×10^{-06}
13	0.4626	2.1383×10^{-11}	-	-
14	99.4993	7.0710×10^{-01}	7.0709×10^{-01}	4.6544×10^{-07}
15	0.4254	1.0778×10^{-13}	-	-
16	99.4991	4.8387×10^{-01}	8.7513×10^{-01}	3.0700×10^{-08}
17	0.3782	4.5013×10^{-16}	-	-
18	99.4989	6.8904×10^{-01}	7.2471×10^{-01}	1.8764×10^{-09}
19	0.3185	2.4808×10^{-17}	-	-
20	98.9435	9.9589×10^{-01}	9.0499×10^{-02}	2.0156×10^{-10}

Then one can check easily that $x^*Ax = y^*Ay = 0$ holds; the vectors $Ax - y$ and $Ay - x$ will have the same *structure* as given in (3.4) or (3.5), respectively:

$$\begin{Bmatrix} Ax - y \\ Ay - x \end{Bmatrix} = \begin{Bmatrix} \left(\tilde{y}_1, \dots, \tilde{y}_{\frac{N-1}{2}}, \tilde{y}_{\frac{N+1}{2}}, \tilde{y}_1, \dots, \tilde{y}_{\frac{N-1}{2}} \right)^* \\ \left(\tilde{x}_1, \dots, \tilde{x}_{\frac{N-1}{2}}, \tilde{x}_{\frac{N+1}{2}}, -\tilde{x}_1, \dots, -\tilde{x}_{\frac{N-1}{2}} \right)^* \end{Bmatrix} \in \mathbb{R}^N.$$

Therefore applying the Lanczos process with a starting vector of the form (3.4) or (3.5) to a diagonal matrix (3.3) such that, on the one hand, zero is a part of A 's spectrum and, on the other hand, all the other eigenvalues lie symmetrically around the origin will generate a Jacobi matrix T_k such that all diagonal entries are zero, $\alpha_1 = \alpha_2 = \dots = \alpha_k = 0$, which implies that the Jacobi matrix T_k has a symmetric spectrum. This means in particular that zero is an eigenvalue of T_k for k being odd.

Example 2. In our experiment we chose a diagonal matrix $A \in \mathbb{R}^{23 \times 23}$ with eigenvalue distribution,

$$(3.6) \quad \begin{aligned} \lambda_1 &= -100, \quad \lambda_{j+1} = \lambda_j + 0.1 \quad \text{for } j = 1, \dots, 10, \\ \lambda_{12} &= 0 \text{ and } \lambda_{12+j} = -\lambda_{12-j} \quad \text{for } j = 1, \dots, 11, \end{aligned}$$

and we took $q_1 = \frac{1}{\sqrt{23}}(1, 1, \dots, 1)^* \in \mathbb{R}^{23}$ as starting vector of the Lanczos algorithm.

We observed the cluster around $\lambda_{12} = 0$; see Table 3.1: the second column displays the norm of the new Lanczos vectors, i.e., $\beta_{k+1} = \|\tilde{q}_{k+1}\|$; in the third and fourth columns one finds the absolute value of the bottom element of T_k 's eigenvectors belonging to the Ritz values near zero. The bar in columns four and five indicates that only one Ritz value near λ_{12} was found at that step. In the fifth column we displayed the value (3.1), which appears to be much smaller than $\delta_{k,j}$ and $\delta_{k,j+1}$. Obviously $\delta_{k,j}$ cannot in general be proportional to (3.1) for a cluster with alternating cluster size.

4. The main results. We mentioned in the introduction that an idea how to solve Conjecture 1 has already been presented in [18], and that is to develop bounds for $\delta_{k,j}$ by using only information about Ritz values in several consecutive steps. We will take up this idea and show how it can be used for the resolution of the hypothesis from [18]. It turns out that it suffices to consider Ritz values in two consecutive steps of the Lanczos method. The bounds for $|s_{k,j}^{(k)}|$ developed in this section are similar to those from [18] (see [18, Lemma 5.1, Theorem 5.3]).

We need some more notation:

$$\psi_k(\lambda) = \det(\lambda I - T_k).$$

By reason of T_k being unreduced, the roots of ψ_{k-1} and ψ_k interlace strictly:

$$(4.1) \quad \theta_1^{(k)} < \theta_1^{(k-1)} < \theta_2^{(k)} < \dots < \theta_{k-1}^{(k)} < \theta_{k-1}^{(k-1)} < \theta_k^{(k)}.$$

For the theorems' proofs in this section we will make use of a well-known formula for the normalized eigenvectors of T_k (see, e.g., [8, p. 103 ff] or [11, p. 129]):

$$(4.2) \quad (s_{k,j}^{(k)})^2 \psi'_k(\theta_j^{(k)}) = \psi_{k-1}(\theta_j^{(k)})$$

for $1 \leq j \leq k$. The number $\delta_{k,j}$ is the product of a subdiagonal element β_{k+1} of the Lanczos matrix times the absolute value of the bottom element of T_k 's normalized eigenvector $s_j^{(k)}$. We will concentrate on the bottom elements of T_k 's normalized eigenvector matrix, because the subdiagonal entries of a Jacobi matrix are bounded,

$$0 < 2\beta_l < \theta_k^{(k)} - \theta_1^{(k)} \leq \lambda_N - \lambda_1 + 2\eta_k, \quad l = 2, 3, \dots, k,$$

which can be checked by considering the Rayleigh quotient of $e_j + e_{j+1}$ with T_k , $1 \leq j \leq k-1$, and using the interlacing property (4.1).

Considering (4.2) one realizes that the bottom elements of T_k 's normalized eigenvectors depend on the eigenvalues of T_k and its principal submatrix T_{k-1} . For instance, if there is a large gap γ between the two largest Ritz values and if $\delta = \theta_k^{(k)} - \theta_{k-1}^{(k-1)} \ll \theta_{k-1}^{(k-1)} - \theta_{k-1}^{(k)} = \gamma$, we obtain (compare with [18, Lemma 5.1])

$$(4.3) \quad (s_{k,k}^{(k)})^2 = \frac{\psi_{k-1}(\theta_k^{(k)})}{\psi'_k(\theta_k^{(k)})} = \frac{\prod_{j=1}^{k-1} (\theta_k^{(k)} - \theta_j^{(k-1)})}{\prod_{j=1}^{k-1} (\theta_k^{(k)} - \theta_j^{(k)})} < \frac{\delta}{\gamma}$$

simply by exploiting (4.2) and the interlacing property (4.1). It is only a slight modification, but the cluster diameter δ and the spectral gap γ are now defined in terms of Ritz values at two consecutive steps of the Lanczos method, whereas in [18] and in section 3 those quantities were defined by use of Ritz values at a single step of the Lanczos method.

Unfortunately, an estimate like (4.3) that only makes use of the interlacing property (4.1) does not apply for all kind of clusters. Also, (4.3) can be improved, and the following theorems provide similar estimations.

Our focus is on the k th step of the Lanczos algorithm. By reason of the interlacing property (4.1) the cluster size in two consecutive steps of the Lanczos method can decrease or increase at most by one. We are able to establish upper bounds in terms of the cluster diameter δ and the spectral gap γ in the following four cases: the cluster size stays constant from step $k-1$ to step k (Theorem 4.2); the cluster size stays constant from step k to $k+1$ (Theorem 4.3); the cluster size decreases from step $k-1$ to k (Theorem 4.5); the cluster size increases from step k to $k+1$ (Theorem 4.6).

To prove Theorems 4.2 and 4.3 we need a result that tells us where to find the absolute value's extreme points of certain rational functions.

LEMMA 4.1. *Let $a \in \mathbb{R}$ and let $p(z) = \prod_{j=1}^l (z - x_j)$ and $q(z) = \prod_{j=1}^l (z - y_j)$ be polynomials with $x_j, y_j \in \mathbb{R}$, $1 \leq j \leq l$. If*

$$(4.4) \quad a - r < x_1 \leq y_1 \leq \cdots \leq x_l \leq y_l < a + r$$

holds for $r > 0$, then

$$\max_{|z-a|=r} \left| \frac{p(z)}{q(z)} \right| = \max \left\{ \left| \frac{p(a-r)}{q(a-r)} \right|, \left| \frac{p(a+r)}{q(a+r)} \right| \right\}.$$

Proof. For the reason of notational convenience we assume without loss of generality $a = 0$ and $r = 1$. Finding the extremal values of $\left| \frac{p(z)}{q(z)} \right|$ on the unit circle is equivalent to finding the maximum and minimum of the function

$$f(t) = \left| \frac{p(e^{it})}{q(e^{it})} \right|^2$$

in the interval $0 \leq t \leq \pi$. But the numerator of f 's derivative is

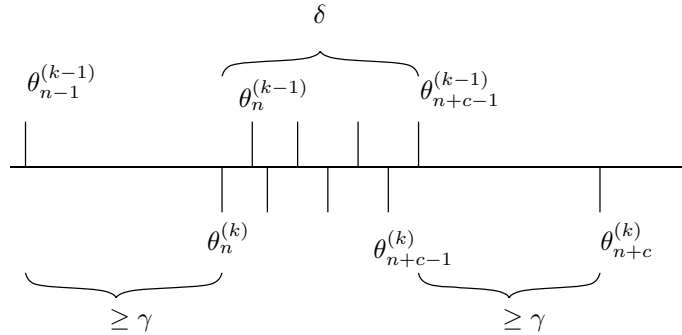
$$(4.5) \quad 2 \sin(t) \sum_{j=1}^l \frac{(x_j - y_j)(1 - x_j y_j)}{|z - x_j|^2 |z - y_j|^2} \prod_{\nu=1}^l |z - x_\nu|^2 |z - y_\nu|^2,$$

with $z = e^{it}$. Since $-1 < x_j \leq y_j < 1$ we see that $(x_j - y_j)(1 - x_j y_j)$ is either zero or negative, $1 \leq j \leq l$. Therefore, if $\frac{p(z)}{q(z)}$ is not a constant function, then (4.5) vanishes only if t equals zero or π according to the interlacing of the roots (4.4), which means that $f(t)$ is a monotone function in $[0, \pi]$. $\square$

Now we are in a position to prove an upper bound similar to (4.3) for the sum of the squares of the bottom elements of T_k 's normalized eigenvectors belonging to clustered Ritz values.

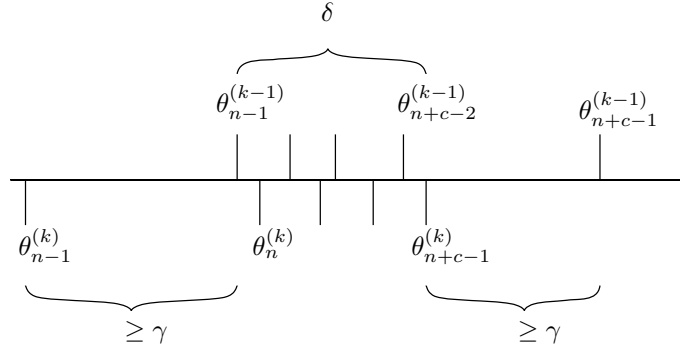
THEOREM 4.2. *Let $1 \leq c \leq k-1$ and suppose that c roots of ψ_{k-1} and c roots of ψ_k form a cluster with diameter δ and spectral gap γ , i.e., either*

$$(4.6) \quad \delta = \theta_{n+c-1}^{(k-1)} - \theta_n^{(k)}, \quad \gamma = \min \left\{ \theta_{n+c}^{(k)} - \theta_{n+c-1}^{(k-1)}, \theta_n^{(k)} - \theta_{n-1}^{(k-1)} \right\},$$



with $n \geq 1$ and $n + c \leq k$ or

$$(4.7) \quad \delta = \theta_{n+c-1}^{(k)} - \theta_{n-1}^{(k-1)}, \quad \gamma = \min \left\{ \theta_{n+c-1}^{(k-1)} - \theta_{n+c-1}^{(k)}, \theta_{n-1}^{(k-1)} - \theta_{n-1}^{(k)} \right\},$$



with $n \geq 2$ and $n + c - 1 \leq k$. If $\frac{\delta}{2} < \gamma$, then

$$\sum_{l=1}^c \left(s_{k,n+l-1}^{(k)} \right)^2 < 3 \frac{\delta}{\gamma - \frac{\delta}{2}}$$

holds.

Proof. We assume the Ritz value distribution (4.6)—the proof assuming (4.7) is analogous. Equation (4.2) is $(s_{k,j}^{(k)})^2 = \frac{\psi_{k-1}(\theta_j^{(k)})}{\psi'_k(\theta_j^{(k)})}$. Now an interesting observation is that $\theta_j^{(k)}$ is a simple root of the function $\psi_k(\lambda)\psi_{k-1}^{-1}(\lambda)$ and therefore

$$\left. \frac{d}{d\lambda} \frac{\psi_k(\lambda)}{\psi_{k-1}(\lambda)} \right|_{\lambda=\theta_j^{(k)}} = \frac{\psi'_k(\theta_j^{(k)})}{\psi_{k-1}(\theta_j^{(k)})} \neq 0$$

holds. If Γ is a simply connected curve that encloses the Ritz values in the cluster but no others, an application of the residue theorem (see, e.g., [1, p. 150]) shows

$$(4.8) \quad \sum_{l=1}^c \left(s_{k,n+l-1}^{(k)} \right)^2 = \sum_{l=1}^c \frac{\psi_{k-1}(\theta_{n+l-1}^{(k)})}{\psi'_k(\theta_{n+l-1}^{(k)})} = \frac{1}{2\pi i} \int_{\Gamma} \frac{\psi_{k-1}(z)}{\psi_k(z)} dz.$$

Thus we have to address the problem of estimating properly the line integral (4.8). We consider a circle since it can be easily parameterized. Therefore let a be the cluster's center, $a = \frac{1}{2}(\theta_n^{(k)} + \theta_{n+c-1}^{(k-1)})$, let $B = B_{\delta}(a)$ be the open disc of radius δ with the center at a and let $\Gamma = \partial B$. By virtue of the interlacing property (4.1) the following two inequalities hold for all $z \in \partial B$:

$$\left| \frac{z - \theta_j^{(k-1)}}{z - \theta_j^{(k)}} \right| < 1 \quad \text{if } 1 \leq j \leq n-1$$

and

$$\left| \frac{z - \theta_j^{(k-1)}}{z - \theta_{j+1}^{(k)}} \right| < 1 \quad \text{if } n+c \leq j \leq k-1.$$

Hence we can estimate the line integral (4.8) as follows:

$$\begin{aligned}
 \left| \frac{1}{2\pi i} \int_{\partial B} \frac{\psi_{k-1}(z)}{\psi_k(z)} dz \right| &\leq \delta \max_{z \in \partial B} \left| \frac{\psi_{k-1}(z)}{\psi_k(z)} \right| \\
 &< \frac{\delta}{\min_{z \in \partial B} |z - \theta_{n+c}^{(k)}|} \max_{z \in \partial B} \left| \prod_{l=1}^c \left(\frac{z - \theta_{n+l-1}^{(k-1)}}{z - \theta_{n+l-1}^{(k)}} \right) \right| \\
 &< 3 \frac{\delta}{\gamma - \frac{\delta}{2}}.
 \end{aligned}$$

We made use of Lemma 4.1 in order to see that

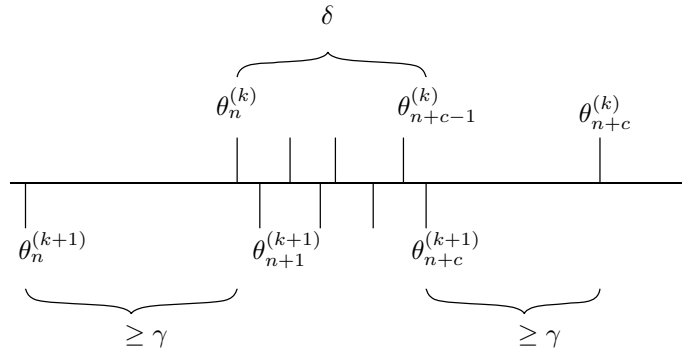
$$\begin{aligned}
 \max_{z \in \partial B} \left| \prod_{l=1}^c \frac{z - \theta_{n+l-1}^{(k-1)}}{z - \theta_{n+l-1}^{(k)}} \right| &= \max_{z \in \{a-\delta, a+\delta\}} \left| \prod_{l=1}^c \frac{z - \theta_{n+l-1}^{(k-1)}}{z - \theta_{n+l-1}^{(k)}} \right| \\
 &< \frac{a - \delta - \theta_{n+c-1}^{(k-1)}}{a - \delta - \theta_n^{(k)}} = 3
 \end{aligned}$$

is valid. This final inequality completes the proof. $\square$

Since the next theorem's assumptions seem to be the same as those of the previous one, we point out that we aim to develop an upper bound for $(s_{k,j}^{(k)})^2$. We assume again a constant cluster size, but different from Theorem 4.2 we will now derive an upper bound by considering the cluster size at the steps k and $k+1$ and not at the steps $k-1$ and k as before.

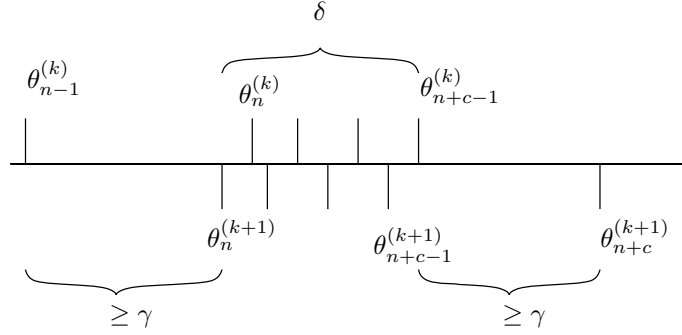
THEOREM 4.3. *Suppose that $c \geq 1$ roots of ψ_k and ψ_{k+1} build a cluster with diameter δ and spectral gap γ , i.e., either*

$$(4.9) \quad \delta = \theta_{n+c}^{(k+1)} - \theta_n^{(k)}, \quad \gamma = \min \left\{ \theta_{n+c}^{(k)} - \theta_{n+c}^{(k+1)}, \theta_n^{(k)} - \theta_n^{(k+1)} \right\},$$



for some $n \geq 1$ with $n+c \leq k$ or

$$(4.10) \quad \delta = \theta_{n+c-1}^{(k)} - \theta_n^{(k+1)}, \quad \gamma = \min \left\{ \theta_{n+c}^{(k+1)} - \theta_{n+c-1}^{(k)}, \theta_n^{(k+1)} - \theta_{n-1}^{(k)} \right\},$$



for some $n \geq 2$ with $n + c - 1 < k$. If $\frac{\delta}{2} < \gamma$, then

$$(4.11) \quad \sum_{l=1}^c \left(s_{k,n-l+1}^{(k)} \right)^2 < 3 \frac{(\sigma_{k+1} - \gamma + \frac{\delta}{2})^2}{\beta_{k+1}^2} \frac{\delta}{\gamma - \frac{\delta}{2}}.$$

Furthermore, if the cluster of c Ritz values lies at the right end of the spectrum⁵, i.e., $\delta = \theta_{k+1}^{(k+1)} - \theta_{k-c+1}^{(k)}$ and $\gamma = \theta_{k-c+1}^{(k+1)} - \theta_{k-c+1}^{(k)}$ and if $\frac{\delta}{2} < \gamma$, then

$$\sum_{l=1}^c \left(s_{k,k-l+1}^{(k)} \right)^2 < 3 \frac{\sigma_{k+1} + \frac{\delta}{2}}{\beta_{k+1}^2} \delta.$$

Proof. The proof does not differ much from the previous one. We show (4.11) for the Ritz value distribution (4.9). According to the recurrence formula for the characteristic polynomials of tridiagonal matrices we see that

$$\psi_{k+1}(\theta_j^{(k)}) = (\theta_j^{(k)} - \alpha_{k+1})\psi_k(\theta_j^{(k)}) - \beta_{k+1}^2 \psi_{k-1}(\theta_j^{(k)}) = -\beta_{k+1}^2 \psi_{k-1}(\theta_j^{(k)}).$$

We again take the center of the cluster $a = \frac{1}{2}(\theta_n^{(k)} + \theta_{n+c}^{(k+1)})$ and set $B = B_\delta(a)$. Equation (4.8) then has to be replaced by

$$\sum_{l=1}^c \left(s_{k,n+l-1}^{(k)} \right)^2 = -\frac{1}{\beta_{k+1}^2} \sum_{l=1}^c \frac{\psi_{k+1}(\theta_{n+l-1}^{(k)})}{\psi_k'(\theta_{n+l-1}^{(k)})} = -\frac{1}{\beta_{k+1}^2} \frac{1}{2\pi i} \int_{\partial B} \frac{\psi_{k+1}(z)}{\psi_k(z)} dz.$$

In the same manner as in the proof of Theorem 4.2 the standard estimate for line integrals, Lemma 4.1, and the interlacing property of ψ_k 's and ψ_{k+1} 's roots show that

$$\left| \frac{1}{2\pi i} \int_{\partial B} \frac{\psi_{k+1}(z)}{\psi_k(z)} dz \right| < 3 \left(\sigma_{k+1} - \gamma + \frac{\delta}{2} \right)^2 \frac{\delta}{\gamma - \frac{\delta}{2}}. \quad \square$$

If the cluster size either at the preceding step or at the subsequent one equals the cluster size at the k th step, we know that the value of $\delta_{k,j}$ is small for all Ritz values $\theta_j^{(k)}$ in such a cluster.

But in Example 2 we were confronted with an alternating cluster size. As Table 3.1 shows we cannot expect $\delta_{k,j}$ to be small if only $\delta \ll \gamma$ in this case. Nevertheless this

⁵It is obvious that the same estimate is valid if the cluster of c Ritz values lies at the left end of the spectrum.

is not the whole story because Table 3.1 also reveals that we have a very good Ritz approximation to (λ_{12}, u_{12}) if only one Ritz value occurs at step k .

LEMMA 4.4. *Suppose that we have two polynomials $p(z) = \prod_{j=1}^{l+1} (z - x_j)$ and $q(z) = \prod_{j=1}^l (z - y_j)$ as well as two numbers $a \in \mathbb{R}$, $r > 0$, such that*

$$(4.12) \quad a - \frac{r}{2} = x_1 < y_1 < x_2 < \cdots < y_l < x_{l+1} = a + \frac{r}{2}$$

holds. Then

$$\max_{|z-a|=r} \left| \frac{p(z)}{q(z)} \right| \leq \frac{4}{3} \sqrt{3} r.$$

Proof. We can assume without loss of generality that $a = 0$.

(a) If $|\Re(z)| \geq \frac{r}{2}$, then because of the interlacing (4.12) of p 's and q 's roots we see that

$$\left| \frac{p(z)}{q(z)} \right| \leq \max \{ |z - x_1|, |z - x_{l+1}| \}.$$

And $r^2 = (\Re(z))^2 + (\Im(z))^2$ shows $(\Im(z))^2 \leq \frac{3}{4}r^2$. Hence

$$\begin{aligned} \left| \frac{p(z)}{q(z)} \right| &\leq \sqrt{\max \{ (\Re(z) - x_1)^2, (\Re(z) - x_{l+1})^2 \} + (\Im(z))^2} \\ &\leq \sqrt{\frac{9}{4}r^2 + \frac{3}{4}r^2} = \sqrt{3} r \end{aligned}$$

holds for all $|z| = r$ with $|\Re(z)| \geq \frac{r}{2}$.

(b) On the other hand, if $|\Re(z)| < \frac{r}{2}$ we see $|\Im(z)| > \sqrt{\frac{3}{4}}r$ at once. And, again, according to the interlacing property (4.12) there exists an index $\iota \in \{1, \dots, l\}$ such that

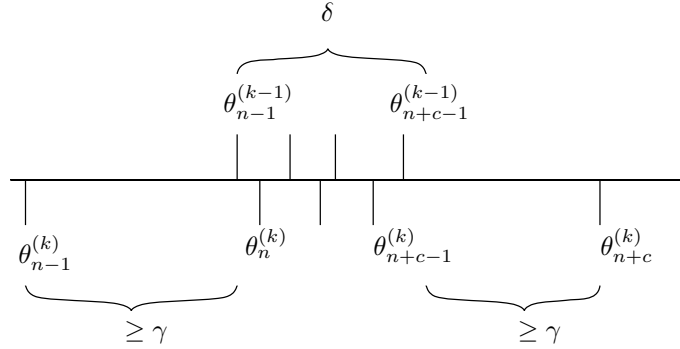
$$\begin{aligned} \left| \frac{p(z)}{q(z)} \right| &< \frac{|z - x_1| |z - x_{l+1}|}{|z - y_\iota|} \leq \frac{|z - x_1| |z - x_{l+1}|}{|\Im(z)|} \\ &< \sqrt{\frac{4}{3}} \frac{1}{\delta} \sqrt{((\Re(z) - x_1)^2 + (\Im(z))^2)((\Re(z) - x_{l+1})^2 + (\Im(z))^2)} \\ &< \sqrt{\frac{4}{3}} \frac{1}{r} \sqrt{(r^2 + r^2)(r^2 + r^2)} = \frac{4}{3} \sqrt{3} r, \end{aligned}$$

which proves the statement. $\square$

Consequently, we can state the following two theorems; their proofs are similar to the proofs of the previous theorems but now Lemma 4.4 is used for estimations instead of Lemma 4.1. For a decreasing cluster size we obtain the following theorem.

THEOREM 4.5. *Let $c \geq 1$ and suppose that ψ_{k-1} has $c+1$ roots in a cluster with diameter δ and spectral gap γ while ψ_k has only c roots in the same cluster, i.e.,*

$$\delta = \theta_{n+c-1}^{(k-1)} - \theta_{n-1}^{(k-1)}, \quad \gamma = \min \left\{ \theta_{n-1}^{(k-1)} - \theta_{n-1}^{(k)}, \theta_{n+c}^{(k)} - \theta_{n+c-1}^{(k-1)} \right\},$$



for some $n \geq 2$ with $n + c \leq k$. If $\frac{\delta}{2} < \gamma$, then

$$\sum_{l=1}^c \left(s_{k,n+l-1}^{(k)} \right)^2 < \frac{4}{3} \sqrt{3} \frac{\delta^2}{\left(\gamma - \frac{\delta}{2} \right)^2}.$$

For an increasing cluster size, we have the following theorem.

THEOREM 4.6. *If the number of ψ_{k+1} 's roots in a cluster with diameter δ and spectral gap γ is $c + 1$ whereas ψ_k has only c roots in the cluster, i.e.,*

$$\delta = \theta_{n+c}^{(k+1)} - \theta_n^{(k+1)}, \quad \gamma = \min \left\{ \theta_n^{(k+1)} - \theta_{n-1}^{(k)}, \theta_{n+c}^{(k)} - \theta_{n+c}^{(k+1)} \right\},$$

where either $n \geq 1$ and $n + c \leq k$ or $n \geq 2$ and $n + c \leq k + 1$ holds, we obtain for $\frac{\delta}{2} < \gamma$,

$$\sum_{l=1}^c \left(s_{k,n+l-1}^{(k)} \right)^2 < \frac{4}{3} \sqrt{3} \frac{(\sigma_{k+1} - \gamma + \frac{\delta}{2})^2}{\beta_{k+1}^2} \frac{\delta^2}{\left(\gamma - \frac{\delta}{2} \right)^2}.$$

The Figures 4.1 and 4.2 serve the purpose to illustrate the results. We used the same setting as for Example 1. Both figures show the sum $\sum (s_{k,j}^{(k)})^2$ belonging to a cluster of Ritz values as a solid line and the corresponding bound developed above as a dashed line. The figures display the typical behavior when a new Ritz value joins the cluster. Figure 4.1 illustrates the case when a second Ritz value starts to approximate the eigenvalue $\lambda_{24} = 100$ at step $k = 16$. The situation in Figure 4.2: A third Ritz value joins the cluster of two Ritz values close to $\lambda_{22} \approx$ at step $k = 33$.

The next corollary comprises the discussion. If there is a tight, well separated cluster of Ritz values at several (or at least two) consecutive steps of the Lanczos method such that the cluster's Ritz values lie in an interval of length δ , and if the separation γ to the Ritz values not part of the cluster satisfies $\gamma > \frac{\delta}{2}$, we obtain the following.

COROLLARY 4.7. *We denote with c_k the number of Ritz values in a cluster in the k th step. If $\theta_{\bullet}^{(k)}$ is a Ritz value in a cluster at the k th step such that $|s_{k,\bullet}^{(k)}|$ is minimal for all Ritz values in the cluster with diameter δ and spectral gap γ , and if we also consider the Ritz values computed at the previous step $k - 1$, we obtain*

$$(4.13) \quad \delta_{k,\bullet} \leq \begin{cases} \text{no estimate (see Table 3.1),} & c_k = c_{k-1} + 1 \\ \sqrt{3} \quad \beta_{k+1} \quad \frac{1}{\sqrt{c_k}} \quad \sqrt{\frac{\delta}{\gamma - \frac{\delta}{2}}}, & c_k = c_{k-1} \\ \frac{4}{\sqrt{3}} \quad \beta_{k+1} \quad \frac{1}{\sqrt{c_k}} \quad \frac{\delta}{\gamma - \frac{\delta}{2}}, & c_k = c_{k-1} - 1. \end{cases}$$

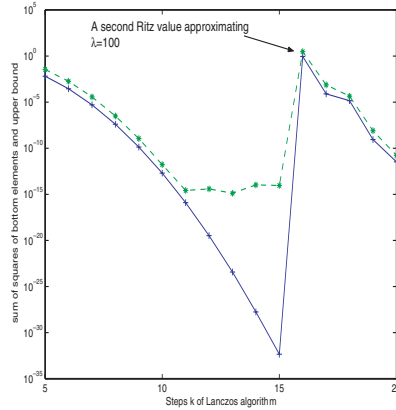


FIG. 4.1. Continuation of Example 1 for λ_{24} . The sum of the squares of the bottom elements of T_k 's normalized eigenvector matrix that belong to a cluster near λ_{24} is displayed as a solid line and the minimum of the upper bounds of Theorems 4.2, 4.3, 4.5, and 4.6 as a dashed line.

On the other hand, if we are not looking at the Ritz values at the previous step but at those at the next step, i.e., $k+1$, we have

$$(4.14) \quad \delta_{k,\bullet} \leq \begin{cases} \text{no estimate (see Table 3.1),} & c_k = c_{k+1} + 1 \\ \sqrt{3} \left(\sigma_{k+1} - \gamma + \frac{\delta}{2} \right) \frac{1}{\sqrt{c_k}} \sqrt{\frac{\delta}{\gamma - \frac{\delta}{2}}}, & c_k = c_{k+1} \\ \frac{4}{\sqrt{3}} \left(\sigma_{k+1} - \gamma + \frac{\delta}{2} \right) \frac{1}{\sqrt{c_k}} \frac{\delta}{\gamma - \frac{\delta}{2}}, & c_k = c_{k+1} - 1. \end{cases}$$

Proof. Since

$$c_k \delta_{k,\bullet}^2 = c_k \beta_{k+1}^2 \left(s_{k,\bullet}^{(k)} \right)^2 \leq \beta_{k+1}^2 \sum_{\text{cluster}} \left(s_{k,j}^{(k)} \right)^2$$

holds, (4.13) and (4.14) follow from the previously developed theorems. $\square$

Taking (2.5) into consideration, a consequence of (4.13) and (4.14) is that a tight, well separated cluster must approximate an eigenvalue of A . Moreover the so-called *persistence theorem* (see [10], [20, p. 171]), i.e.,

$$\min_{1 \leq l \leq t} \left| \theta_{\bullet}^{(k)} - \theta_l^{(t)} \right| \leq \delta_{k,\bullet}, \quad t > k,$$

guarantees that at least one Ritz value will always remain close to the eigenvalue. In an analogy to the definition in [10] we say that a cluster of Ritz values has stabilized to within Δ if we know that there is a Ritz value within Δ of the cluster for every subsequent step of the Lanczos algorithm. If furthermore Δ is of order $\mathcal{O}(\|A\|\epsilon)$ we say that the cluster has stabilized.

We have just seen that any tight, well separated cluster has stabilized to within $\Delta = \mathcal{O}(\|A\|\sqrt{\frac{\delta}{\gamma}})$, or even $\Delta = \mathcal{O}(\|A\|\frac{\delta}{\gamma})$; if furthermore $\delta \approx \epsilon$, this justifies the statement that any tight, well separated cluster is *stabilized*.

Another implication of (2.5), (4.13), and (4.14) is that an intermediate cluster, i.e., a cluster lying between two eigenvalues and not approximating one of them, can comprise at most one Ritz value. Knizhnerman developed a similar result in [6].

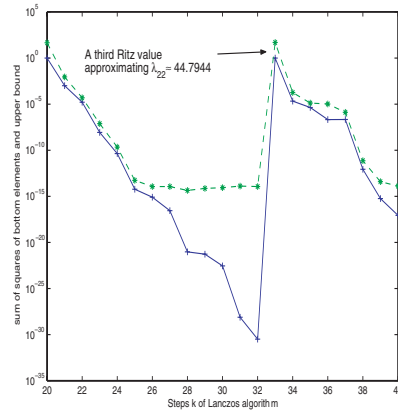


FIG. 4.2. Continuation of Example 1 for $\lambda_{22} \approx 44.7944$. This time we consider the case where we already have two Ritz approximations and a third emerges. The figure displays—similar to Figure 4.1—the sum of squares of the bottom elements of T_k 's normalized eigenvector matrix that belong to a cluster around λ_{22} as a solid line and the minimum of the upper bounds of Theorems 4.2, 4.3, 4.5, and 4.6 as a dashed line.

Conclusion. In this paper we worked on the question posed in [18], whether the residual bound $\delta_{k,j}$ is small for any Ritz value $\theta_j^{(k)}$ in a tight, well separated cluster. The only exception found is the case of a cluster with alternating cluster size. But it is still true that any cluster is stabilized to within the conjectured bound (3.1).

The recipe to resolve the hypothesis: We used the idea from [18] to express an upper bound for $\delta_{k,j}$ only in terms of Ritz values, we made use of the residue theorem, and exploited the Ritz values' interlacing property (4.1).

Acknowledgments. I would like to express my sincere gratitude to Zdeněk Strakoš for support, valuable discussions, and all the helpful comments. Many thanks are also given to Ludwig Elsner for helpful comments and support. The anonymous referees' input greatly improved the manuscript.

REFERENCES

- [1] L. V. AHLFORS, *Complex Analysis*, 3rd ed., McGraw-Hill, New York, 1978.
- [2] Z. BAI, *Error analysis of the Lanczos algorithm for the nonsymmetric eigenvalue problem*, Math. Comp., 62 (1994), pp. 209–226.
- [3] H. FASSBENDER, *Error analysis of the symplectic Lanczos method for the symplectic eigenvalue problem*, BIT, 40 (2000), pp. 471–496.
- [4] A. GREENBAUM, *Behavior of slightly perturbed Lanczos and conjugate-gradient recurrences*, Linear Algebra Appl., 113 (1989), pp. 7–63.
- [5] A. GREENBAUM AND Z. STRAKOŠ, *Predicting the behavior of finite precision Lanczos and conjugate gradient computations*, SIAM J. Matrix Anal. Appl., 13 (1992), pp. 121–137.
- [6] L. A. KNIZHNERMAN, *The quality of approximations to a well-isolated eigenvalue and the arrangement of Ritz numbers in a simple Lanczos process*, Comput. Math. Math. Phys., 35 (1995), pp. 1175–1187.
- [7] C. LANCZOS, *An iteration method for the solution of the eigenvalue problem of linear differential and integral operators*, J. Research Nat. Bur. Standards, 45 (1950), pp. 255–282.
- [8] C. C. PAIGE, *The Computation of Eigenvalues and Eigenvectors of Very Large Sparse Matrices*, Ph.D. thesis, London University, Institute of Computer Science, London, England, 1971.
- [9] C. C. PAIGE, *Error analysis of the Lanczos algorithm for tridiagonalizing a symmetric matrix*, J. Inst. Math. Appl., 18 (1976), pp. 341–349.

- [10] C. C. PAIGE, *Accuracy and effectiveness of the Lanczos algorithm for the symmetric eigenproblem*, Linear Algebra Appl., 34 (1980), pp. 235–258.
- [11] B. N. PARLETT, *The Symmetric Eigenvalue Problem*, Prentice Hall, Englewood Cliffs, NJ, 1980.
- [12] B. N. PARLETT, *Misconvergence in the Lanczos Algorithm*, Oxford Univ. Press, New York, 1990.
- [13] B. N. PARLETT AND J. LE, *Forward instability of tridiagonal QR*, SIAM J. Matrix Anal. Appl., 14 (1993), pp. 279–316.
- [14] B. N. PARLETT, *Do we fully understand the symmetric Lanczos algorithm yet?*, Proceedings of the Cornelius Lanczos International Centenary Conference, Raleigh, NC, 1993, pp. 93–107, SIAM, Philadelphia, 1994.
- [15] B. N. PARLETT, *Invariant subspaces for tightly clustered eigenvalues of tridiagonals*, BIT, 36 (1996), pp. 542–562.
- [16] D. C. SORENSEN, *Implicit application of polynomial filters in a k -step Arnoldi method*, SIAM J. Matrix Anal. Appl., 13 (1992), pp. 357–385.
- [17] Z. STRAKOŠ, *On the real convergence rate of the conjugate gradient method*, Linear Algebra Appl., 154–156 (1991), pp. 535–549.
- [18] Z. STRAKOŠ AND A. GREENBAUM, *Open Questions in the Convergence Analysis of the Lanczos Process for the Real Symmetric Eigenvalue Problem*, IMA Research Report 934, IMA Workshop on Iterative Methods, University of Minnesota, Minneapolis, MN, 1992.
- [19] R. C. THOMPSON AND P. MCENTEGGERT, *Principal submatrices, II: The upper and lower quadratic inequalities*, Linear Algebra Appl., 1 (1968), pp. 211–243.
- [20] J. H. WILKINSON, *The Algebraic Eigenvalue Problem*, Clarendon Press, Oxford, 1965.
- [21] K. WU AND H. SIMON, *Thick-restart Lanczos method for large symmetric eigenvalue problems*, SIAM J. Matrix Anal. Appl., 22 (2000), pp. 602–616.
- [22] Q. YE, *On close eigenvalues of tridiagonal matrices*, Numer. Math., 70 (1995), pp. 507–514.
- [23] J. P. M. ZEMKE, *Krylov Subspace Methods in Finite Precision: A Unified Approach*, Dissertation, Technische Universität Hamburg-Harburg, Hamburg, Germany, 2003.
- [24] J. P. M. ZEMKE, *(Hessenberg) Eigenvalue-Eigenmatrix Relations*, Report 78 des Arbeitsbereiches Mathematik 4–13, Technische Universität Hamburg-Harburg, Hamburg, Germany, 2004.

APPROXIMATE INVERSE PRECONDITIONING OF FINITE ELEMENT DISCRETIZATIONS OF ELLIPTIC OPERATORS WITH NONSMOOTH COEFFICIENTS*

MARIO BEBENDORF†

Abstract. A new class of approximate inverse preconditioners for large finite element stiffness matrices arising from second order elliptic partial differential operators is introduced which guarantees a bounded, problem-independent condition number. Since the proposed preconditioner is based on hierarchical matrices, it can be generated, stored, and multiplied by a vector with almost linear complexity. This preconditioner may be used as a black-box method, because it is robust with respect to varying coefficients and because it can be applied to arbitrary quasi-uniform grids.

Key words. approximate inverse, preconditioning, nonsmooth coefficients, hierarchical matrices

AMS subject classifications. 35C20, 65F05, 65F50, 65N30

DOI. 10.1137/S0895479803437621

1. Introduction. This article deals with the efficient solution of large sparse linear systems

$$(1) \quad Sx = b, \quad S \in \mathbb{R}^{N \times N},$$

arising from finite element (FE) discretization of general second order elliptic partial differential operators

$$(2) \quad Lu = -\operatorname{div} [A(x)\nabla u + bu] + c \cdot \nabla u + du$$

on a bounded Lipschitz domain $\Omega \subset \mathbb{R}^n$ with coefficients $a_{ij}, b_i, c_j, d \in L^\infty(\Omega)$, $i, j = 1, \dots, n$. Since the solution of (1) is often the most time-consuming part of a computation, much attention has been paid to developing efficient solution strategies. There are two main classes of methods: *iterative* and *direct* solvers. The latter are based on factorizations of the sparse coefficient matrix S into easily invertible matrices. These methods are widely used due to their robustness. However, they suffer from so-called *fill-in*; i.e., compared with the sparsity of S , considerably more entries of the factors will be nonzero. This usually happens to all entries within the bandwidth of the original matrix, which for Galerkin matrices of operators (2) scales like $N^{1-1/n}$ even if the bandwidth has been reduced, for instance by the reverse Cuthill–McKee, by the minimum degree algorithm, or by nested dissection. Hence, the fill-in will lead to a computational complexity of order $N^{3-2/n}$. Constants, however, are extremely small, making direct methods the methods of choice if N is not too large. This situation persists if we are to solve large-scale problems in two dimensions. Here, recent multifrontal solvers (see [1] and the references therein) can be used. However, for higher spatial dimensions usually *iterative* methods such as Krylov subspace methods are more efficient, especially, if an approximate solution

*Received by the editors November 11, 2003; accepted for publication by E. Ng February 10, 2005; published electronically February 16, 2006. This work was supported by the DFG priority program SPP 1146 “Modellierung inkrementeller Umformverfahren.”

<http://www.siam.org/journals/simax/27-4/43762.html>

†Fakultät für Mathematik und Informatik, Universität Leipzig, Augustusplatz 10/11, D-04109 Leipzig, Germany (bebendorf@math.uni-leipzig.de).

of relatively low accuracy is sought. The advantage of these solution techniques is that the coefficient matrix enters the computation only through the matrix-vector product. On the other hand, the convergence rate and hence the number of iterations usually depends on the distribution of the eigenvalues of the coefficient matrix. Since L is a second order operator, the condition number of the FE stiffness matrix S in the Hermitian case grows like $N^{2/n}$ for large N but also depends significantly on the coefficients of L . By preconditioning, i.e., replacing $Sx = b$ by $SCu = b$, $x = Cu$, or $CSx = Cb$, where C is a right or a left explicit preconditioner, respectively, one tries to improve the convergence properties of the Krylov subspace method.

One of the best known and most often used preconditioning techniques is the *incomplete LU factorization* (ILU) and its variants; see [28]. Although the ILU can be applied to any sparse coefficient matrix provided the factorization does not break down, it is well suited for M - and diagonally dominant matrices. Generating an ILU is extremely fast. However, it usually does not lead to a bounded number of iterations of the solver. This goal can be achieved by, for instance, *multigrid* [20] or the Bramble, Pasciak, and Xu (BPX) algorithm [12] if the problem class is restricted to stiffness matrices of elliptic operators with smooth coefficients, where special properties of the operator can be exploited. For nonsmooth coefficients these methods might still work but suffer from poor convergence rates if the respective method is not adapted to the coefficients. In the past few years, much work has been done to develop robust multilevel methods. The *algebraic multigrid* (AMG) solvers [27] try to achieve this robustness by mostly heuristic strategies. Domain decomposition methods [30] are commonly used if the computational domain can be subdivided into a small number of parts, on each of which the coefficients do not vary too much. In this case, a problem can be decomposed into smaller ones for the purpose of parallel processing. For a more detailed survey on precondition techniques the reader is referred to [9].

The aim of this article is to present a new preconditioning technique which adapts itself to arbitrarily nonsmooth coefficients and, even more important in practice, does not need a grid hierarchy. For preconditioning the best choice for C would obviously be $C = S^{-1}$. However, the computation of S^{-1} is usually more costly than solving $Sx = b$ for x . Hence, not S^{-1} but an approximation of S^{-1} is used for preconditioning. This idea gives rise to the class of approximate inverse preconditioners. Their main advantage is that, in contrast to nonapproximating preconditioners, which usually guarantee only a bounded condition number with respect to N , the condition number depends only on the accuracy of the approximation (see (9)), which can be arbitrarily chosen. This is of particular importance if a bad condition number is caused by the coefficients of the operator or by the computational domain. The inverse of an FE stiffness matrix, however, is a dense matrix in general since it corresponds to the inverse operator L^{-1} , which can be represented as a nonlocal integral operator

$$(3) \quad L^{-1}\varphi(x) = \int_{\Omega} G(x, y)\varphi(y) \, dy,$$

where G denotes the Green's function of L and Ω . In order to avoid large dense matrices, the so-called sparse approximate inverse (SAI) preconditioners (see [18, 14, 13] and [11] for an overview) have been introduced. In this approach the quantity $\|I - SC\|_F$ is usually minimized for matrices C having a certain sparsity pattern. If S is Hermitian positive definite, the matrix C obtained from the minimization is not positive definite in general and cannot be used with the conjugate gradient method. To overcome this problem, the class of *factorized sparse approximate inverse* (FSAI) *preconditioners* [23, 24] was introduced; see also the AINV algorithm [10]. Our

aim, in contrast, is to use a dense approximation of the inverse of the FE stiffness matrix. Since the sparsity pattern is crucial for robustness, the approximant will be enabled to capture more properties of S^{-1} that are important for preconditioning. For instance, C can be guaranteed to be Hermitian positive definite if S has this property. Nevertheless, it will be possible to generate, store, and multiply the approximant by a vector with almost linear complexity.

In the past few years, fast methods for the treatment of large dense matrices $M \in \mathbb{R}^{N \times N}$ have become widely popular. After the introduction of the fast multipole method [26], numerous methods based on degenerate approximations

$$(4) \quad G(x, y) \approx \sum_{\ell=1}^k u_{\ell}(x) v_{\ell}(y), \quad x \in D_1, y \in D_2,$$

with appropriate functions $u_{\ell}, v_{\ell}, \ell = 1, \dots, k$, on a pair of domains (D_1, D_2) lying in a certain distance relative to their diameters, have been developed. Looking at (4) from an algebraic point of view, this means nothing but a low-rank approximation of the associated block $M_{s \times t}$ in the rows and columns $s, t \in \{1, \dots, N\}$ of M :

$$M_{s \times t} \approx UV^T,$$

where $U \in \mathbb{R}^{s \times k}$, $V \in \mathbb{R}^{t \times k}$, and k is small compared with $|s|$ and $|t|$. The ideas of the fast multipole method originally aiming at an efficient approximate evaluation of matrix-vector products have recently been extended to a structure called hierarchical ($\mathcal{H}$ -) matrices [19, 22]; see also [32]. Basically, these are matrices that are low-rank on each block of a certain partition stemming from a recursive subdivision of the set of matrix indices. In addition to the efficient matrix-vector multiplication, this structure provides approximate operations like matrix addition, matrix-matrix multiplication, and matrix inversion of dense matrices with almost linear complexity. Furthermore, $\mathcal{H}$ -matrices can be stored in an almost linear amount of units of memory. Multiplication with $\mathcal{H}$ -matrices can easily be parallelized; see [6]. The parallelization of other matrix operations is currently under investigation. However, promising results have been published in [25]. The structure of $\mathcal{H}$ -matrices has originally been applied to integral equations; see [5, 7].

Although FE stiffness matrices are sparse, it can be seen that they can be represented by $\mathcal{H}$ -matrices without approximation. Since the bandwidth of a stiffness matrix is of order $N^{1-1/n}$, the rank of each block of S^{-1} above or below the diagonal will be of the same order. Since the band has many zeros in it, it may be expected that to approximate each block a much lower rank is sufficient. Recently, it was shown [8] that the inverse of an FE stiffness matrix for operators of type (2) can be approximated by $\mathcal{H}$ -matrices with blockwise rank of order $\log^{n+3} N$. For this purpose it was proved that the Green's function in (3) can be approximated in the sense of (4). Interestingly, this approximation is very robust with respect to varying coefficients.

In a recent publication [15] a similar approach for preconditioning was investigated. The inverse of S is approximated on a matrix partition similar to the partition of a hierarchical matrix. But in contrast to $\mathcal{H}$ -matrices, a blockwise constant approximation of the inverse was used; i.e., only one real number represents all entries of each block. Obviously, this needs less memory than storing a low-rank matrix, and for preconditioning the approximant does not have to be too accurate. However, a certain accuracy is needed, and the block constant approach is not adaptive; i.e., there is no possibility to guarantee a given accuracy on each block. In our approach

the approximate inverse will be calculated with a prescribed accuracy, although this accuracy will not be too high. This will enable us to guarantee a bounded number of iterations of any appropriate Krylov subspace method. The procedure in [15] to compute the real number for each block cannot be used if one tries to approximate the block by a matrix of low rank. For obtaining our approximate inverse preconditioner the $\mathcal{H}$ -arithmetic will be employed.

The structure of the rest of this article is as follows. In section 2 a brief introduction to the structure of $\mathcal{H}$ -matrices will be given. All results and notations from the field of $\mathcal{H}$ -matrices necessary for this article will be presented. In particular, we present all steps needed to build a matrix partition. It will be seen that this can be done automatically. Section 3 is devoted to the theory of preconditioning. A condition on the accuracy of the approximate inverse will be derived which guarantees the existence of an eigenvalue cluster leading to superlinear convergence of the preconditioned conjugate gradient method (PCG). In addition to Hermitian stiffness matrices, the case of non-Hermitian matrices also will be treated. In this case, not the condition number of S but the distribution of eigenvalues determines the convergence rate. In section 4 this condition is used when generating the approximate inverse by an approximate block inversion procedure known as the $\mathcal{H}$ -matrix inversion. Once the matrix partition has been generated from the mesh information, a preconditioner can be obtained from any stiffness matrix in a purely algebraic way. Finally, in section 5 numerical results for elliptic partial differential operators with nonsmooth coefficients will be presented. It will be seen that the proposed preconditioner can be computed, stored, and multiplied by a vector with almost linear complexity. Furthermore, it can be seen that with this preconditioner a bounded number of iterations can be achieved. We will test it on two- and three-dimensional scalar problems and apply it to a problem for which the algebraic multigrid method fails. In particular, we will demonstrate the efficiency of the proposed preconditioner on convection-diffusion problems.

2. Hierarchical matrices. This section gives a brief overview over the structure of $\mathcal{H}$ -matrices originally introduced by Hackbusch [19] and Hackbusch and Khoromskij [22]. We will describe the two principles on which the efficiency of $\mathcal{H}$ -matrices is based. These are the hierarchical partitioning of the matrix into blocks and the blockwise restriction to low-rank matrices; see Figure 2. These principles were also used in the mosaic-skeleton method [32].

In this article we will be interested in matrices $M \in \mathbb{R}^{N \times N}$ with indices

$$m_{ij} = a(\varphi_j, \varphi_i), \quad i, j = 1, \dots, N,$$

where φ_i are basis functions with support $X_i := \text{supp } \varphi_i$, $i \in I := \{1, \dots, N\}$, and a is a bilinear form. Note that this does not imply that M is sparse. Also the variational formulation of nonlocal operators is of this form. In order to be able to approximate each block $b = s \times t$, $s, t \subset I$, of M by a matrix of low rank, i.e.,

$$M_b \approx UV^T, \quad U \in \mathbb{R}^{s \times k}, V \in \mathbb{R}^{t \times k},$$

where k is small compared with $|s|$ and $|t|$, b has to satisfy a certain condition, which is caused by the operator L hidden in a . In the field of elliptic partial differential operators the corresponding Green's function $G(x, y)$ for L has a singularity for $x = y$. Hence, the following condition on $b = s \times t$ has proved useful:

$$(5) \quad \min\{\text{diam } X_s, \text{diam } X_t\} \leq \eta \text{dist}(X_s, X_t),$$

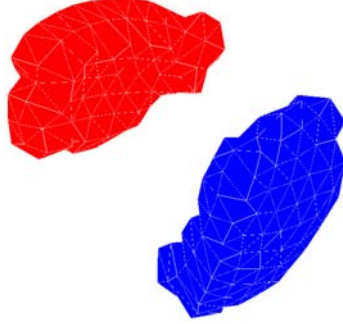


FIG. 1. An admissible cluster pair.

where $\eta > 0$ is a given real number which is typically chosen from the interval $[0.5, 1.5]$; see Figure 1. As usual we define

$$\text{diam } X_t = \sup_{x, y \in X_t} |x - y| \quad \text{and} \quad \text{dist}(X_t, X_s) = \inf_{x \in X_t, y \in X_s} |x - y|$$

and the support of a cluster¹ t is the union of the supports of the basis functions corresponding to the indices in t :

$$X_t := \bigcup_{i \in t} X_i.$$

Note that condition (5) implies that the partition we are looking for has to be refined towards the diagonal of M , since the diagonal entries arise from the interaction of the same basis functions, i.e., $\text{dist}(X_s, X_t) = 0$ for $s \subset t$.

2.1. The cluster tree. Since M cannot be approximated globally by a single low-rank matrix, we have to subdivide M into blocks satisfying (5). One possibility is to recursively subdivide a block $b = s \times t$ into four subblocks $s_1 \times t_1$, $s_1 \times t_2$, $s_2 \times t_1$, and $s_2 \times t_2$, where $s = s_1 \cup s_2$ and $t = t_1 \cup t_2$, until its parts satisfy condition (5). The rule how to subdivide a cluster t is given by the so-called *cluster tree* T_I satisfying the following conditions:

- (i) I is the root of T_I ,
- (ii) if $t \in T_I$ is not a leaf, then t has sons $t_1, t_2 \in T_I$, so that $t = t_1 \cup t_2$.

The set of sons of t is denoted by $\mathcal{S}(t)$, while $\mathcal{L}(T_I)$ stands for the set of leaves of the tree T_I . A cluster tree is usually generated by recursive subdivision of I so as to minimize the diameter of each part. For practical purposes the recursion should be stopped if a certain cardinality $n_{\min}$ of the clusters is reached, rather than subdividing the clusters until only one index is left.

There are different methods for building the cluster tree. We favor the following strategy which is based on the *principal component analysis* and can be applied to general quasi-uniform sets X_i with centers $p_i \in \mathbb{R}^n$, $i \in I$. A cluster $t \subset I$ is subdivided by the hypersurface through the centroid

$$m_t := \frac{\sum_{i \in t} |X_i| p_i}{\sum_{i \in t} |X_i|}$$

¹An index cluster should not be confused with a cluster of eigenvalues.

of t with normal w_t , where $|\cdot|$ denotes the volume of X_i and w_t is the *main direction* of t , i.e., the vector maximizing

$$\sum_{i \in t} |w^T(p_i - m_t)|^2$$

with respect to w . Note that with $D_t := \sum_{i \in t} (p_i - m_t)(p_i - m_t)^T \in \mathbb{R}^{n \times n}$ it holds that

$$\sum_{i \in t} |w^T(p_i - m_t)|^2 = \sum_{i \in t} w^T(p_i - m_t)(p_i - m_t)^T w = w^T D_t w.$$

Hence, w_t is the eigenvector of D_t corresponding to the largest eigenvalue. Using w_t , the sons $\mathcal{S}(t) = \{t_1, t_2\}$ of t are defined by

$$t_1 = \{i \in t : w_t^T(p_i - m_t) > 0\}$$

and $t_2 := t \setminus t_1$. The same division strategy is then recursively applied to the sons t_1, t_2 of t . The depth p of the resulting cluster tree T_I is of order $\log N$. The complexity of building it can be estimated as $\mathcal{O}(N \log N)$; cf. [4].

Remark 2.1. Sometimes, the number of sons of a cluster in the previous definition of a cluster tree is not restricted to two. This generalization has not proved useful in practice, however.

2.2. The block cluster tree. Based on a cluster tree T_I which contains a hierarchy of partitions of I , we are able to construct the so-called *block cluster tree* $T_{I \times I}$ describing a hierarchy of partitions of $I \times I$ by the following rule.

PROCEDURE *build_block_cluster_tree*($s \times t$).

begin

if (s, t) does not satisfy (5) and $s, t \notin \mathcal{L}(T_I)$ then

$\mathcal{S}(s \times t) := \{s' \times t' : s' \in \mathcal{S}(s), t' \in \mathcal{S}(t)\}$

for $s' \times t' \in \mathcal{S}(s \times t)$ do *build_block_cluster_tree*($s' \times t'$)

else $\mathcal{S}(s \times t) := \emptyset$

end

Applying *build_block_cluster_tree* to $I \times I$, we obtain a cluster tree for the index set $I \times I$. Upon completion of the algorithm, the set of leaves $P := \mathcal{L}(T_{I \times I})$ is a partition of $I \times I$ with blocks $b = s \times t \in P$ either satisfying (5) or consisting of clusters s and t with $\min\{|s|, |t|\} \leq n_{\min}$. The complexity of building the block cluster tree in the case of quasi-uniform grids can be estimated as $\mathcal{O}(\eta^{-n} N \log N)$; cf. [4].

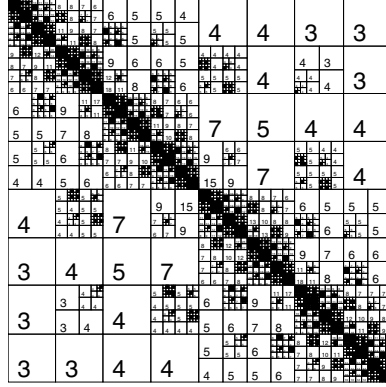
Remark 2.2. In the case of unstructured grids the computation of the distance in (5) between two supports X_s and X_t is too costly. Therefore, for practical purposes the supports are enclosed in sets of a simpler structure, e.g., boxes or spheres.

We are now in a position to define the set of $\mathcal{H}$ -matrices for a partition P with blockwise rank k

$$\mathcal{H}(P, k) := \{M \in \mathbb{R}^{I \times I} : \text{rank } M_b \leq k \text{ for all } b \in P\}.$$

Note that $\mathcal{H}(P, k)$ is not a linear space, since in general the sum of two rank- k matrices exceeds rank k .

Remark 2.3. For a block $B \in \mathbb{R}^{s \times t}$ the low-rank representation $B = UV^T$, $U \in \mathbb{R}^{s \times k}$, $V \in \mathbb{R}^{t \times k}$, is advantageous compared with the entrywise representation only if $k(|s| + |t|) \leq |s||t|$. For the sake of simplicity in this article we will, however, assume that each block has the low-rank representation. Employing the entrywise representation for appropriate blocks will accelerate the algorithms.

FIG. 2. An $\mathcal{H}$ -matrix with its rank distribution.

2.3. Storage and matrix-vector multiplication. The cost of multiplying an $\mathcal{H}$ -matrix $M \in \mathcal{H}(P, k)$ and its transpose M^T by a vector $x \in \mathbb{R}^N$ is inherited from the blockwise matrix-vector multiplication:

$$Mx = \sum_{s \times t \in P} M_{s \times t} x_t \quad \text{and} \quad M^T x = \sum_{s \times t \in P} (M_{s \times t})^T x_s.$$

Since each block $s \times t$ has the representation $M_{s \times t} = UV^T$, $U \in \mathbb{R}^{s \times k}$, $V \in \mathbb{R}^{t \times k}$ (see Remark 2.3), $\mathcal{O}(k(|s| + |t|))$ units of memory are needed to store $M_{s \times t}$, and the matrix-vector products $M_{s \times t} x_t = UV^T x_t$ and $(M_{s \times t})^T x_s = VU^T x_s$ can be done with $\mathcal{O}(k(|s| + |t|))$ operations. Exploiting the hierarchical structure of M , it can therefore be shown that both storing M and multiplying M and M^T by a vector have complexity $\mathcal{O}(\eta^{-n} k N \log N)$. For a rigorous analysis the reader is referred to [4]. Therefore, $\mathcal{H}$ -matrices are well suited for iterative schemes such as Krylov subspace methods. The hierarchical matrix-vector multiplication can be computed in parallel. See [6] for an appropriate scheduling algorithm.

2.4. Where can $\mathcal{H}$ -matrices be applied? The structure of $\mathcal{H}$ -matrices was originally designed to accelerate the building process and the matrix-vector multiplication of discrete integral operators with smooth kernels having an algebraic singularity at $x = y$. This kind of integral operator arises, for example, from the boundary element method. Condition (5) was also used to prove convergence of the adaptive cross approximation (ACA) algorithm [5, 7] for the generation of low-rank approximants based on few of the original matrix entries.

Although $\mathcal{H}$ -matrices are primarily aiming at dense matrices, the stiffness matrix S of the differential operator L from (2) is in $\mathcal{H}(P, n_{\min})$. If $b \in P$ satisfies (5), then the supports of the basis functions are pairwise disjoint. Hence, the matrix entries in this block vanish. In the remaining case b does not satisfy (5). Then the size of one of the clusters is less than or equal to $n_{\min}$. In both cases the rank of S_b does not exceed $n_{\min}$.

Recently, it was shown [8] that for Dirichlet problems together with S , its inverse also can be approximated by low-rank matrices on each block of a partition satisfying (5). Let L be uniformly elliptic; i.e., for the coefficient $A(x) \in \mathbb{R}^{n \times n}$ of L it holds that A is symmetric with $a_{ij} \in L^\infty(\Omega)$ and

$$0 < \lambda \leq \lambda(x) \leq \Lambda$$

for all eigenvalues $\lambda(x)$ of $A(x)$ and almost all $x \in \Omega$. Let $e_h(u) := \|u - P_h u\|_{L^2(\Omega)}$ be the FE error, where $P_h : H_0^1(\Omega) \rightarrow V_h$ is the Ritz projector mapping $u \in H_0^1(\Omega)$ to its FE solution, i.e., the solution of $a(u_h, v_h) = f(v_h)$ for all $v_h \in V_h$. The weakest form of the FE convergence is described by

$$e_h(u) \leq \varepsilon_h \|f\|_{L^2(\Omega)} \quad \text{for all } u = L^{-1}f, \quad f \in L^2(\Omega),$$

where $\varepsilon_h \rightarrow 0$ as $h \rightarrow 0$.

THEOREM 2.4. *Let p be the depth of the cluster tree T_I defined in section 2.1. Then there is a constant $c > 0$ defining $k := cp^2 \log^{n+1}(p/\varepsilon_h)$ and there is $C_{\mathcal{H}} \in \mathcal{H}(P, k)$ such that*

$$\|S^{-1} - C_{\mathcal{H}}\|_2 \leq c\varepsilon_h,$$

where c depends on the coefficients of L , the diameter of Ω , and η . If $\varepsilon_h = \mathcal{O}(h^\beta)$ for some $\beta > 0$, $k = \mathcal{O}(\log^{n+3} N)$ holds.

To prove the previous theorem the integral representation

$$L^{-1}\varphi(x) = \int_{\Omega} G(x, y)\varphi(y) \, dy$$

with the Green's function G of L and Ω was used. In contrast to operators arising from the boundary element method, the kernel function G is locally only in H^1 with respect to each variable. It is nevertheless shown in [8] that G and hence the inverse of the FE stiffness matrix can be approximated. In [3] this result is extended to general second order elliptic operators.

3. Spectral equivalence and eigenvalue clusters. When speaking about the complexity of an algorithm for the solution of a linear system arising from the discretization of a continuous problem, we investigate a sequence of systems

$$(6) \quad S_N x_N = b_N, \quad N \rightarrow \infty,$$

where each $S_N \in \mathbb{R}^{N \times N}$ is invertible. However, for the sake of readability the index N will be dropped in the rest of this article whenever this dependency is obvious from the context.

Since the Galerkin stiffness matrix S is ill-conditioned, (6) has to be preconditioned; i.e., instead of (6) we have to solve the equivalent system

$$SC\tilde{x} = b,$$

where the nonsingular matrix $C \in \mathbb{R}^{N \times N}$ is an explicit right preconditioner for S such that in the Hermitian case

$$\text{cond}_2(SC) < \text{cond}_2(S).$$

If S is non-Hermitian, the aim of the preconditioner is to obtain a spectrum of SC that is clustered away from the origin. The solution x of (6) can then be computed by $x = C\tilde{x}$. When a Krylov subspace method is used, it is not necessary to form the preconditioned matrix SC . Instead, matrix-vector multiplications with C and S should be performed consecutively.

In this article we propose to employ an approximate inverse preconditioner. It will be seen that a low-accuracy approximation of the inverse of S is sufficient to

guarantee a bounded number of preconditioned iterations for appropriate iterative schemes. Moreover, the number of iterations will depend neither on the operator nor on the computational domain but only on the accuracy of the approximate inverse.

In the first part of this section the case of Hermitian positive definite coefficient matrices S is treated; the second part concentrates on non-Hermitian matrices.

3.1. Hermitian positive definite coefficient matrices. Depending on the operator L , its FE Galerkin matrix S is often Hermitian positive definite. If (6) is to be solved iteratively, the *conjugate gradients method* is the method of choice. Its convergence rate and hence the number of iterations required to obtain a prescribed accuracy of the solution is determined by the distribution of eigenvalues of S ; see Theorem 3.2.

The size of the smallest and the largest eigenvalue of FE stiffness matrices $S \in \mathbb{R}^{N \times N}$ arising from the discretization of second order elliptic partial differential operators is of order $N^{-2/n}$ and 1, respectively. Hence, the spectral condition number of S grows with N like $N^{2/n}$ leading to a prohibitively large number of iterations if N is large.

The aim of this article is to present a preconditioner C such that the number of iterations for the preconditioned coefficient matrix SC is bounded by a constant. A bounded number of iterations, in turn, is guaranteed by an asymptotically well-conditioned matrix SC .

DEFINITION 3.1. *Let $\{M_N\}_{N \in \mathbb{N}}$ be a sequence of Hermitian matrices. Assume that there is a constant $c > 0$ such that*

$$(7) \quad \text{cond}_2(M_N) \leq c.$$

Then $\{M_N\}_{N \in \mathbb{N}}$ is said to be asymptotically well conditioned.

The following theorem (cf. [2]) describes the convergence of the conjugate gradient method.

THEOREM 3.2. *Let the spectrum of SC be decomposed in the following way:*

$$\sigma(SC) = \{\lambda'_i, i = 1, \dots, p\} \cup \Lambda \cup \{\lambda''_i, i = 1, \dots, q\}, \quad \Lambda \subset [a, b].$$

Then for the error $x_k - x$ of PCG it holds that

$$\|x_k - x\|_S \leq 2(\text{cond}_2(SC) + 1)^p \left(\frac{\sqrt{b/a} - 1}{\sqrt{b/a} + 1} \right)^{k-p-q} \|x_0 - x\|_S, \quad k = 0, \dots, N-1.$$

Hence, if SC is asymptotically well conditioned, we may take $\Lambda = \sigma(SC)$. In this case PCG converges linearly, and the number of iterations depends only on the condition number of SC and not on N .

Although asymptotically well-conditioned coefficient matrices lead to a bounded number of iterations, this number might still be large since the constant in (7) usually depends on the coefficients of the differential operator L . For instance, in the case of highly discontinuous coefficients this influence might be more severe than the dependence on N ; see the numerical results. The condition

$$(8) \quad \|I - SC\|_2 \leq \delta < 1,$$

in which δ does not depend on N , does not only give an asymptotically well-conditioned matrix SC , the condition number of SC is even bounded independently of the operator L . This can be seen from the triangle inequality

$$\|SC\|_2 \leq \|I\|_2 + \|I - SC\|_2 \leq 1 + \delta$$

and from the Neumann series

$$\|(SC)^{-1}\|_2 \leq \sum_{k=0}^{\infty} \|I - SC\|_2^k = \frac{1}{1 - \delta},$$

which together lead to

$$(9) \quad \text{cond}_2(SC) = \|SC\|_2 \|(SC)^{-1}\|_2 \leq \frac{1 + \delta}{1 - \delta}.$$

In addition, condition (8) guarantees that C is nonsingular. To see this, let λ be an eigenvalue of SC with associated eigenvector z , $\|z\|_2 = 1$, then

$$(10) \quad |1 - \lambda| = \|(I - SC)z\|_2 \leq \|I - SC\|_2 \leq \delta < 1.$$

Hence, with SC , C also is nonsingular.

Usually, matrix-vector multiplications with C should be much cheaper than solving $Sx = b$ for x . Therefore, the choice $C = S^{-1}$, which is optimal with respect to the condition number, is not valid for preconditioning. However, (8) can be satisfied by a low-accuracy approximation of the inverse of S . Note that if we would use an approximation C to the inverse in order to obtain a solution Cb with relative accuracy

$$\|x - Cb\|_2 \leq \varepsilon \|x\|_2,$$

we would have to guarantee $\|I - SC\|_2 \leq \varepsilon$ instead of (8), since

$$\|x - Cb\|_2 = \|(I - CS)x\|_2 \leq \|I - SC\|_2 \|x\|_2 \leq \varepsilon \|x\|_2.$$

In contrast to ε , which usually should be of the same order of magnitude as the FE error, δ from (8) can be chosen independently of the discretization parameter N . Note that by (9), for instance, $\delta = 0.5$ is sufficient to guarantee a condition number of at most 3.

According to Theorem 2.4, an approximation to the inverse of S with arbitrary accuracy can be sought in the set of $\mathcal{H}$ -matrices. These matrices have almost linear complexity and the approximation is cheap since the accuracy enters the complexity estimates only through the logarithm.

Remark 3.3. Usually, preconditioning is not based on approximation since a preconditioner C leads to the same condition number as, for instance, $2C$ does. By the approximation condition (8) it is, however, possible to guarantee problem-independent convergence rates.

Note that for (9) it is sufficient to assume that $\delta < 1$ in (8). However, in order to be able to apply PCG, C additionally needs to be Hermitian positive definite. It is interesting to see that this is already guaranteed by condition (8)

THEOREM 3.4. *Assume that S is Hermitian positive definite. Then any Hermitian matrix C satisfying (8) is positive definite, too.*

Proof. According to the assumptions, the square root $S^{1/2}$ of S can be defined. Since SC is similar to the Hermitian matrix $S^{1/2}CS^{1/2}$, the eigenvalues of SC are real. Moreover, from (10) for the smallest eigenvalue of $S^{1/2}CS^{1/2}$ it follows that

$$\lambda_{\min}(S^{1/2}CS^{1/2}) = \lambda_{\min}(SC) \geq 1 - \delta.$$

Let $x \neq 0$ and $y = S^{-1/2}x$. Then $y \neq 0$ and we have

$$x^T Cx = y^T S^{1/2}CS^{1/2}y \geq (1 - \delta)\|y\|_2^2 > 0,$$

which proves that C is positive definite. $\square$

From Theorem 3.2 it can also be seen that few small or large eigenvalues do not affect the rate of convergence. Therefore, even more important for the rate of convergence is the distribution of eigenvalues within the spectrum. The condition number depends only on the minimum and maximum eigenvalue of SC and does not provide any information about the distribution of the eigenvalues within this interval. A faster convergence of PCG can be obtained by a condition on this distribution neglecting the size of the interval [31].

DEFINITION 3.5. By $\gamma_M(\epsilon)$ we denote the number of eigenvalues of $M \in \mathbb{R}^{N \times N}$ lying outside a circle with radius $\epsilon > 0$ around the origin. The eigenvalues of a sequence of matrices $\{M_N\}_{N \in \mathbb{N}}$ are said to have a cluster at 0 if

$$\gamma_{M_N}(\epsilon) \leq c(\epsilon)$$

for all $N \in \mathbb{N}$ and all $\epsilon > 0$, where $c(\epsilon)$ does not depend on N . The eigenvalues of $\{M_N\}_{N \in \mathbb{N}}$ are said to have a cluster at $z \in \mathbb{C}$ if $\{M_N - zI_N\}_{N \in \mathbb{N}}$ has a cluster at 0.

If S and C are Hermitian positive definite and SC has a cluster at 1, then PCG converges superlinearly; i.e., for all $0 < q < 1$ and for all sufficiently large N the residual at the k th iteration is bounded by cq^k . Why this superlinear convergence happens is explained in [2].

The following theorem states that for the existence of eigenvalue clusters the approximation C of S^{-1} again does not have to be too accurate.

THEOREM 3.6. Let $\{M_N\}_{N \in \mathbb{N}} \subset \mathbb{R}^{N \times N}$ be a bounded sequence, i.e., $\|M_N\|_F \leq c$, where c does not depend on N and $\|\cdot\|_F$ denotes the Frobenius norm. Then both the singular values and the eigenvalues of $\{M_N\}_{N \in \mathbb{N}}$ cluster at 0.

Proof. By $\nu(\epsilon)$ we denote the number of nonincreasingly ordered singular values $\sigma_i(M)$ of M lying outside a circle with radius $\epsilon > 0$ around the origin. Assume that $\nu(\epsilon) > c^2/\epsilon^2$. Then

$$c^2 < \nu(\epsilon)\epsilon^2 \leq \sum_{i=1}^N \sigma_i^2(M) = \|M\|_F^2 \leq c^2,$$

which gives the contradiction. Hence, $\nu(\epsilon) \leq c^2/\epsilon^2$.

In order to prove the same property for the eigenvalues, let $M = QTQ^H$ be Schur's form with unitary Q and upper-triangular matrix T with the eigenvalues of M on its diagonal. The assertion follows from

$$(11) \quad \|M\|_F^2 = \|T\|_F^2 \geq \sum_{i=1}^N |\lambda_i(M)|^2$$

and the same arguments as above. $\square$

Therefore, if for C it holds that

$$(12) \quad \|I - SC\|_F \leq c$$

with a constant $c > 0$ which does not depend on N , the eigenvalues of SC cluster at 1 and PCG converges superlinearly. Note that in order to guarantee that C is positive definite, we have to assume that additionally $c < 1$.

3.2. Non-Hermitian coefficient matrices. If L is not self-adjoint (this happens, for example, in the presence of convection terms), then S cannot be expected to be Hermitian. In this case not the spectral condition number of the coefficient matrix but a cluster of eigenvalues away from the origin usually determines the convergence rate of appropriate Krylov subspace methods such as GMRES or Bi-CGSTAB. Nevertheless, a low-accuracy approximate inverse will be sufficient to guarantee a bounded number of iterations.

For the convergence of GMRES, for instance, the numerical range

$$F(SC) = \{x^H SCx : x \in \mathbb{C}^N, \|x\|_2 = 1\}$$

of SC is of particular importance. It is known (see [17]) that

$$\|b - Sx_k\|_2 \leq 2 \left(\frac{r}{|z|} \right)^k \|b\|_2$$

if $F(SC) \subseteq B_r(z)$, where $B_r(z)$ denotes the closed disc around z with radius r . A similar behavior can be observed for other non-Hermitian Krylov subspace methods. Condition (8) implies that $F(SC) \subseteq B_\delta(1)$, which can be seen from

$$|x^H SCx - 1| = |x^H (SC - I)x| \leq \|I - SC\|_2 \leq \delta \quad \text{for all } x \in \mathbb{C}^N, \|x\|_2 = 1.$$

Therefore, (8) also leads to a problem-independent convergence rate

$$\|b - Sx_k\|_2 \leq 2\delta^k \|b\|_2$$

of GMRES.

4. Algorithms. In this section it is described how the preconditioner C is generated from the stiffness matrix S . From Theorem 2.4 we know that an approximation of S^{-1} exists in the set of $\mathcal{H}$ -matrices. Therefore, our aim is to find it in this set using the $\mathcal{H}$ -arithmetic. Since S is in $\mathcal{H}(P, n_{\min})$ (see section 2.4), the computation of the preconditioner can be completely done in the format of $\mathcal{H}$ -matrices. After the preconditioner C has been set up, it enters the iteration through the hierarchical matrix-vector multiplication introduced in section 2.3. Since we will see that at most $k \sim \log N$ is required for condition (8) to hold, the complexity for storing and for applying the preconditioner is $\mathcal{O}(N \log^2 N)$.

Since $\mathcal{H}(P, k)$ is not a linear space, we have to replace the usual matrix operations by truncated ones. Starting with the $\mathcal{H}$ -matrix addition, an $\mathcal{H}$ -matrix multiplication is defined. Using these modified operations it is possible to define an $\mathcal{H}$ -matrix inversion based on formulas for the inverse of a partitioned matrix. These ideas already appeared in the early papers on $\mathcal{H}$ -matrices; cf. [19, 22, 21] and [16] for an implementation of these ideas. Due to the various representations of matrix blocks for the implementation many cases have to be distinguished. Therefore only the main ideas of the $\mathcal{H}$ -arithmetic are presented. Note that if not otherwise stated in this section the $\mathcal{H}$ -arithmetic is explained in terms of arbitrary matrices $A, B, C \in \mathcal{H}(P, k)$.

4.1. Truncated addition. In order to make the sum of two $\mathcal{H}(P, k)$ -matrices be in $\mathcal{H}(P, k)$, we have to add them blockwise and truncate each sum UV^T , $U = (U_1, U_2) \in \mathbb{R}^{s \times 2k}$, $V = (V_1, V_2) \in \mathbb{R}^{t \times 2k}$, of two rank- k blocks $U_1 V_1^T$ and $U_2 V_2^T$ to a rank- k matrix with accuracy ε . This truncated addition will be denoted by $\oplus_\varepsilon$, and

we define the addition of two submatrices A, B in the entries $\hat{b} \in T_{I \times I}$, where $T_{I \times I}$ denotes the cluster tree from section 2.2, by

$$A \oplus B = \{A_b \oplus_\varepsilon B_b \text{ for all } b \in P, b \text{ is a descendant of } \hat{b} \text{ in } T_{I \times I}\}.$$

The truncation can be done by the following algorithm, which was already presented in [4] to find the approximant with lowest rank to a prescribed accuracy.

PROCEDURE *truncate*($U, V, k, \text{var } \tilde{U}, \text{var } \tilde{V}, \text{var } \tilde{k}$).

begin

Compute the QR -decompositions $U = Q_U R_U$ and $V = Q_V R_V$.

Compute $M := R_U R_V^T \in \mathbb{R}^{2k \times 2k}$.

Compute the singular value decomposition $M = X \Sigma Y^T$.

Choose $\tilde{k}$ from $\sigma_{\tilde{k}+1} \leq \varepsilon \sigma_1$, where σ_i is the i th singular value of M .

Let $\Sigma_{\tilde{k}}$ and $Y_{\tilde{k}}$ be the first $\tilde{k}$ columns of Σ and Y , respectively.

Compute $\tilde{U} := Q_U X \Sigma_{\tilde{k}}$ and $\tilde{V} := Q_V Y_{\tilde{k}}$.

end

Obviously, $\tilde{U} \tilde{V}^T$ has rank $\tilde{k}$ and the relative error in spectral norm is bounded by

$$\|UV^T - \tilde{U} \tilde{V}^T\|_2 = \frac{\sigma_{\tilde{k}+1}}{\sigma_1} \|UV^T\|_2 \leq \varepsilon \|UV^T\|_2.$$

The previous truncation algorithm needs $\mathcal{O}(k^2(|s| + |t|))$ operations if $b = s \times t$. Hence, exploiting the block hierarchy, the complexity of the $\mathcal{H}$ -matrix addition of two matrices from $\mathcal{H}(P, k)$ can be shown to be of order $k^2 N \log N$ in the same way as was done for the matrix-vector multiplication.

4.2. Truncated matrix-matrix multiplication. Since the partition P consists of the leaves of the block cluster tree $T_{I \times I}$, it is possible to recursively define a modified matrix-matrix multiplication $C \oplus = A \odot B$, $A, B \in \mathcal{H}(P, k)$, making use of the partitioned matrix-matrix multiplication. Let $r \times s, s \times t, r \times t \in T_{I \times I} \setminus \mathcal{L}(T_{I \times I})$ be block clusters. In order to define what is meant by $C_{r \times t} \oplus = A_{r \times s} \odot B_{s \times t}$, we focus on the 2×2 partition of the blocks

$$\begin{bmatrix} C_{r_1 \times t_1} & C_{r_1 \times t_2} \\ C_{r_2 \times t_1} & C_{r_2 \times t_2} \end{bmatrix} \oplus = \begin{bmatrix} A_{r_1 \times s_1} & A_{r_1 \times s_2} \\ A_{r_2 \times s_1} & A_{r_2 \times s_2} \end{bmatrix} \odot \begin{bmatrix} B_{s_1 \times t_1} & B_{s_1 \times t_2} \\ B_{s_2 \times t_1} & B_{s_2 \times t_2} \end{bmatrix}$$

with the sons of the clusters s, t , and r . Hence, we define

$$\begin{aligned} C_{r_1 \times t_1} \oplus &= A_{r_1 \times s_1} \odot B_{s_1 \times t_1}, & C_{r_1 \times t_1} \oplus &= A_{r_1 \times s_2} \odot B_{s_2 \times t_1}, \\ C_{r_1 \times t_2} \oplus &= A_{r_1 \times s_1} \odot B_{s_1 \times t_2}, & C_{r_1 \times t_2} \oplus &= A_{r_1 \times s_2} \odot B_{s_2 \times t_2}, \\ C_{r_2 \times t_1} \oplus &= A_{r_2 \times s_1} \odot B_{s_1 \times t_1}, & C_{r_2 \times t_1} \oplus &= A_{r_2 \times s_2} \odot B_{s_2 \times t_1}, \\ C_{r_2 \times t_2} \oplus &= A_{r_2 \times s_1} \odot B_{s_1 \times t_2}, & C_{r_2 \times t_2} \oplus &= A_{r_2 \times s_2} \odot B_{s_2 \times t_2}. \end{aligned}$$

On the leaves of $T_{I \times I}$ we employ the usual matrix-matrix multiplication. Obviously, the result C is an $\mathcal{H}(P, k)$ -matrix. If the truncation would not produce any approximation error, then $C \oplus = A \odot B$ would coincide with $C := C + AB$. The complexity of the truncated matrix-matrix multiplication can be estimated as $\mathcal{O}(k^2 N \log^2 N)$; cf. [16].

4.3. Inversion. We assume that each block $A_{s \times s}$, $s \in T_I$, of $A \in \mathcal{H}(P, k)$ is invertible. This is, for example, the case if A is Hermitian positive definite. Let

$s \in T_I \setminus \mathcal{L}(T_I)$. The corresponding matrix block $A_{s \times s}$ is subdivided into the sons of $s \times s$:

$$A_{s \times s} = \begin{bmatrix} A_{s_1 \times s_1} & A_{s_1 \times s_2} \\ A_{s_2 \times s_1} & A_{s_2 \times s_2} \end{bmatrix}.$$

It is easy to see that for the inverse of the 2×2 block matrix A it holds that

$$A_{s \times s}^{-1} = \begin{bmatrix} A_{s_1 \times s_1}^{-1} + A_{s_1 \times s_1}^{-1} A_{s_1 \times s_2} S_A^{-1} A_{s_2 \times s_1} A_{s_1 \times s_1}^{-1} & -A_{s_1 \times s_1}^{-1} A_{s_1 \times s_2} S_A^{-1} \\ -S_A^{-1} A_{s_2 \times s_1} A_{s_1 \times s_1}^{-1} & S_A^{-1} \end{bmatrix},$$

where S_A is the *Schur complement* $S_A = A_{s_2 \times s_2} - A_{s_2 \times s_1} A_{s_1 \times s_1}^{-1} A_{s_1 \times s_2}$. The $\mathcal{H}$ -matrix inverse $C_{s \times s}$ of $A_{s \times s}$ is defined by replacing the matrix-matrix multiplication and the addition in the inversion formula from above by the $\mathcal{H}$ -versions. We need a temporary matrix $T \in \mathcal{H}(P, k)$ which, together with C , is initialized to zero and can be removed after the computation of C . Hence, in addition to a copy of the original sparse stiffness matrix we have to store two $\mathcal{H}$ -matrices leading to a total storage requirement of order $kN \log N$; compare section 2.3.

According to Theorem 2.4 we know that an approximate $\mathcal{H}$ -inverse to a prescribed accuracy of the FE stiffness matrix exists. However, this does not directly imply that each of the intermediate results of the following algorithm can be approximated by an $\mathcal{H}$ -matrix of small rank. Note that during the truncated addition we can specify only the accuracy of the approximation; the required rank follows from the distribution of singular values on each block. In order to justify that the rank is still small, the submatrices $A_{s_1 \times s_1}$ and S_A may be regarded as the discretization of elliptic problems defined on subdomains of Ω . For these, Theorem 2.4 can be used to guarantee an approximate inverse of small rank.

PROCEDURE *invertH*($s, A, \text{var } C$).

```

begin
  if  $s \in \mathcal{L}(T_I)$  then  $C_{s \times s} := A_{s \times s}^{-1}$  is the usual inverse.
  else begin
    invertH( $s_1, A, C$ ).
     $T_{s_1 \times s_2} \ominus= C_{s_1 \times s_1} \odot A_{s_1 \times s_2}$ .
     $T_{s_2 \times s_1} \ominus= A_{s_2 \times s_1} \odot C_{s_1 \times s_1}$ .
     $A_{s_2 \times s_2} \oplus= A_{s_2 \times s_1} \odot T_{s_1 \times s_2}$ .
    invertH( $s_2, A, C$ ).
     $C_{s_1 \times s_2} \oplus= T_{s_1 \times s_2} \odot C_{s_2 \times s_2}$ .
     $C_{s_2 \times s_1} \oplus= C_{s_2 \times s_2} \odot T_{s_2 \times s_1}$ .
     $C_{s_1 \times s_1} \oplus= T_{s_1 \times s_2} \odot C_{s_2 \times s_1}$ .
  end
end
```

Then $C \in \mathcal{H}(P, k)$ is an approximation of A^{-1} . The matrix A is destroyed during the previous algorithm. The cost for the computation of the $\mathcal{H}$ -inverse is mainly determined by the cost of the $\mathcal{H}$ -multiplication. Therefore, an approximation to the inverse of A can be obtained with complexity $\mathcal{O}(k^2 N \log^2 N)$.

If the approximate inverse is used for preconditioning the conjugate gradient method, we have to ensure that C is Hermitian. If A is Hermitian, the usual $\mathcal{H}$ -arithmetic can be simplified to a Hermitian one leading to approximately half the amount of operations and storage. A parallel version of the above inversion algorithm has been investigated in [25].

The same algorithm can also be used for preconditioning boundary element Galerkin matrices. We already mentioned in section 2.4 that boundary element Galerkin matrices can be efficiently treated by $\mathcal{H}$ -matrices. A reason for this is that the arising integral operators are elliptic pseudodifferential operators. The inverses of elliptic pseudodifferential operators are pseudodifferential operators, too. Hence, we are also able to approximate the inverses using the above inversion algorithm.

Since the stiffness matrix S is in $\mathcal{H}(P, n_{\min})$, the $\mathcal{H}$ -inversion procedure can be directly applied to S . Due to (8) the blockwise accuracy has to be chosen such that globally at most

$$(13) \quad \|I - SC\|_F \leq \delta < 1$$

holds. We need the following lemma [16], by which the spectral norm of an $\mathcal{H}$ -matrix can be estimated by its blockwise norms. P is again assumed to be generated as in section 2.

LEMMA 4.1. *There is a constant $c > 0$ such that for any matrix $M \in \mathcal{H}(P, k)$ the following inequality holds:*

$$\|M\|_2 \leq c p \max_{b \in P} \|M_b\|_2.$$

Therefore, a blockwise rank $k \sim \log N$ is sufficient to guarantee (13). Hence, C can be computed with complexity at most $N \log^4 N$. The dependence on the coefficients is difficult to analyze. From the numerical experiments, however, it will be seen that a dependence can hardly be observed.

4.4. Updating the preconditioner. Sometimes, many systems with coefficient matrices differing in only a small number of entries have to be solved. This problem arises, for instance, as a consequence of Newton's method for the solution of nonlinear systems. In this case after a small update the preconditioner generated for one matrix can be used to precondition the other. Assume that A and

$$\tilde{A} = A + \alpha e_i e_j^T$$

with the canonical vectors $e_i, e_j \in \mathbb{R}^n$ are invertible. Then due to the Sherman-Morrison formula for the inverse of $\tilde{A}$ it holds that

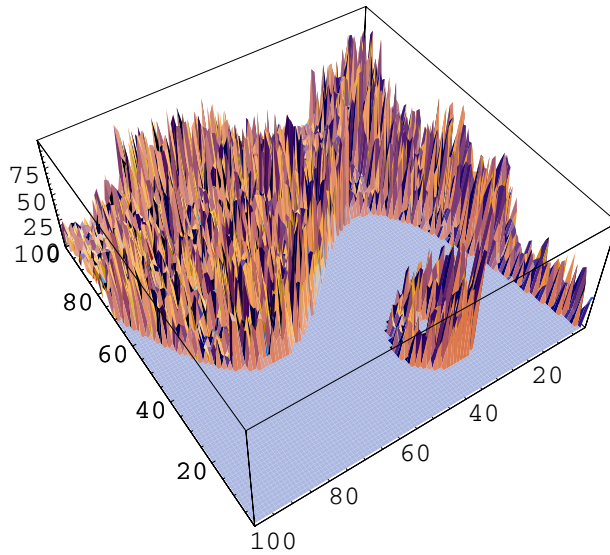
$$\tilde{A}^{-1} = (A + \alpha e_i e_j^T)^{-1} = A^{-1} - \frac{\alpha}{1 + \alpha e_j^T A^{-1} e_i} A^{-1} e_i e_j^T A^{-1}.$$

Hence, $\tilde{A}^{-1}$ and A^{-1} differ only by matrix of rank 1. In the case that A and $\tilde{A}$ have r different entries, the update will be of rank at most r . Using the $\mathcal{H}$ -matrix addition we are able to compute an approximation $C \in \mathcal{H}(P, k)$ of $\tilde{A}^{-1}$. Obviously, this is much faster than building a new preconditioner for $\tilde{A}$.

5. Numerical results. In this section we apply the proposed preconditioner to two- and three-dimensional scalar problems. The emphasis in these tests is placed on robustness with respect to varying coefficients.

5.1. Two-dimensional diffusion. We consider the Dirichlet boundary value problem

$$\begin{aligned} -\operatorname{div} A(x) \nabla u &= 0 & \text{in } \Omega, \\ u &= f & \text{on } \partial\Omega, \end{aligned}$$

FIG. 3. *The coefficient α .*

where Ω is the unit square in $\mathbb{R}^2$. The coefficients A are chosen to be of the form

$$A(x) = \begin{bmatrix} 1 & 0 \\ 0 & \alpha(x) \end{bmatrix}, \quad x \in \Omega,$$

where $\alpha(x) = 1$ in the lower region of Figure 3 and a random number from the interval $[0, a]$ in the remaining part of the unit square. In the following tables we compare the diagonally preconditioned conjugate gradients method (DPCG) and the $\mathcal{H}$ -approximate inverse preconditioned CG (HPCG). The relative accuracy of the residual is 10^{-6} . The minimal cluster size (see section 2.1) was chosen to be $n_{\min} = 60$. The tests were carried out on an Intel Pentium IV (2.666 GHz) with 2 GB of core memory. For compiling the $\mathcal{H}$ -matrix library² the Intel compiler was used.

In Table 1 the amplitude a of the coefficient $A(x)$ is chosen to be 100. For increasing problem sizes N in the third column the time that was needed to compute the approximate inverse C is shown. The memory consumption can be found in the fourth column. The second column contains the truncation accuracy ε we have used when computing C ; see section 4.1. In the last four columns the number of iterations and the CPU time of CG with the preconditioner from this article and the respective quantities for the DPCG are compared. In Tables 2 and 3 the same results are shown for the amplitudes $a = 10^4$ and $a = 10^6$, respectively. In Table 3 the iterations of DPCG were stopped after 120 minutes. Apparently, the diagonal preconditioner suffers from both the amplitude a and the discretization parameter N . The number of iterations of HPCG, however, is almost constant. The proposed preconditioner is able to adapt itself to the varying coefficients.

5.2. Comparison with geometric multigrid. The multigrid method is one of the most efficient methods that are known for the solution of elliptic problems. For smooth coefficients the efficiency of this method can hardly be beaten. However,

²A C++ implementation of the $\mathcal{H}$ -matrix structure can be obtained from the website <http://www.mathematik.uni-leipzig.de/~bebendorf/AHMED.html>.

TABLE 1
Amplitude $a = 10^2$.

N	ε	T_{appr} MB		HPCG		DPCG	
				Its	T	Its	T
38025	$4.5_{10}-2$	1.3s	26.6	29	1.0s	2046	7.4s
50176	$3.5_{10}-2$	2.0s	29.9	27	1.2s	2396	12.5s
82944	$2.8_{10}-2$	3.6s	65.6	29	2.4s	3156	27.6s
159201	$2.3_{10}-2$	8.5s	128.5	35	5.8s	4468	74.9s
233289	$2.0_{10}-2$	13.6s	174.2	41	9.7s	5467	133.3s
455625	$5.1_{10}-3$	39.5s	421.4	35	20.1s	7893	418.6s

TABLE 2
Amplitude $a = 10^4$.

N	ε	T_{appr} MB		HPCG		DPCG	
				Its	T	Its	T
38025	$4.5_{10}-2$	1.3s	25.3	45	1.5s	18806	68.2s
50176	$3.5_{10}-2$	1.9s	28.6	45	2.1s	21970	104.6s
82944	$2.8_{10}-2$	3.3s	62.9	47	3.8s	28892	248.5s
159201	$2.3_{10}-2$	8.0s	121.8	52	8.3s	41352	680.7s
233289	$2.0_{10}-2$	12.2s	164.9	58	13.3s	50508	1224.9s
455625	$5.1_{10}-3$	35.9s	388.8	71	38.1s	72815	3559.9s

TABLE 3
Amplitude $a = 10^6$.

N	ε	T_{appr} MB		HPCG		DPCG	
				Its	T	Its	T
38025	$4.5_{10}-2$	1.2s	23.6	74	2.3s	—	—
50176	$3.5_{10}-2$	1.8s	27.4	79	3.3s	—	—
82944	$2.8_{10}-2$	3.2s	59.8	75	5.8s	—	—
159201	$2.3_{10}-2$	7.1s	109.1	78	11.7s	—	—
233289	$1.9_{10}-2$	11.6s	155.1	76	16.7s	—	—
455625	$5.1_{10}-3$	31.5s	356.4	74	37.6s	—	—

for nonsmooth coefficients this situation changes if multigrid is not adapted to the coefficients of the operator. In the following example we demonstrate that varying coefficients really have an impact on the convergence properties of multigrid. We investigate the Dirichlet boundary value problem

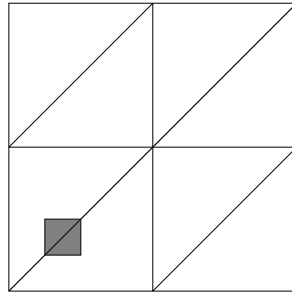
$$\begin{aligned} -\operatorname{div} \alpha(x) \nabla u &= 1 \quad \text{in } \Omega, \\ u &= 0 \quad \text{on } \partial\Omega \end{aligned}$$

on the unit square $\Omega = (0, 1) \times (0, 1)$, where the coefficient α is defined by

$$\alpha(x) = \begin{cases} a, & x \in \{x = (x_1, x_2) : 0.125 \leq x_1 \leq 0.25, 0.125 \leq x_2 \leq 0.25\}, \\ 1 & \text{else;} \end{cases}$$

see Figure 4. We use piecewise linear ansatz functions and apply the geometric multigrid procedure to a hierarchy of nested grids $\mathcal{T}_q$, $q = 4, 5, 6, 7$. For this example, in Table 4 the convergence rates of the V - and the W -cycle using two Gauss–Seidel steps for pre- and postsmoothing, respectively, are shown.

For large a , the convergence of both the V - and the W -cycle slows down significantly. Hence, without special adaptation to the coefficients, the multigrid procedure almost does not converge for large a . Table 4 shows the convergence rates of HPCG.

FIG. 4. Domain Ω , the coefficient α and the coarsest grid.TABLE 4
Convergence rates q of multigrid and HPCG.

a	q (V-cycle)				q (W-cycle)				q (HPCG)	
	4	5	6	7	4	5	6	7	6	7
10^0	0.110	0.122	0.129	0.131	0.089	0.092	0.087	0.078	0.45	0.49
10^2	0.798	0.802	0.804	0.805	0.664	0.464	0.234	0.098	0.44	0.49
10^4	0.998	0.998	0.998	0.998	0.996	0.991	0.983	0.967	0.45	0.49
10^6	0.999	0.999	0.999	0.999	0.999	0.999	0.999	0.999	0.43	0.50

TABLE 5
HPCG for diffusion-convection.

N	$a = 10$				$a = 100$			
	T_{appr}	MB	Its	T	T_{appr}	MB	Its	T
15625	1.9s	25.1	10	0.4s	1.6s	23.7	21	0.8s
39601	4.9s	50.9	21	1.7s	4.8s	50.7	20	1.6s
78961	10.4s	110.1	18	2.9s	10.4s	110.1	19	3.1s
159201	27.7s	228.0	19	6.6s	27.7s	228.1	20	7.0s
318096	60.4s	489.1	25	19.6s	60.8s	489.0	20	15.7s

For these tests $\delta = 0.2$ was chosen. The convergence rate is not influenced by the size of a .

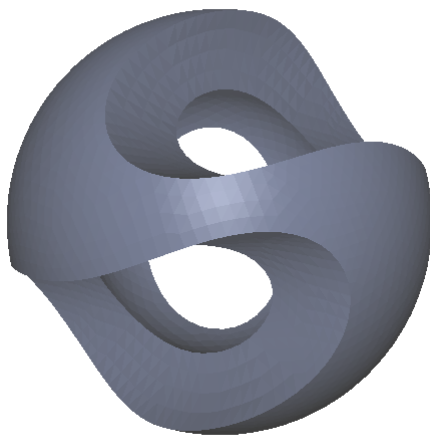
5.3. Two-dimensional convection-diffusion. In this subsection operators of the type

$$L = -\Delta + \mathbf{c} \cdot \nabla$$

will be considered. The convection coefficient $\mathbf{c}$ is randomly chosen, i.e., $\mathbf{c}(x) \in [-a, a]^2$ for $x \in \Omega := (0, 1)^2$. Table 5 shows the number of iterations for different parameters a . Note that in this case a symmetry of the stiffness matrix cannot be exploited. We could test only relatively small amplitudes a since the FE discretization becomes unstable in the convection-dominated case.

5.4. Three-dimensional diffusion. The aim of this subsection is to demonstrate that the proposed method is not restricted to the unit square; in fact it can equally be applied to any quasi-uniform FE discretization given just by the grid information. We test the proposed preconditioner on two tetrahedral discretizations ($N = 26270$ and $N = 227100$ of unknowns) of the volume shown in Figure 5. These meshes were generated by NETGEN [29].

The operator is $L = -\text{div } A(x) \nabla$, where $A(x) = \text{diag}(\alpha_1(x), \alpha_2(x), \alpha_3(x))$ is a diagonal matrix. The coefficients α_i , $i = 1, 2, 3$, are set to one in the left half space

FIG. 5. *The computational domain.*TABLE 6
HPCG for three-dimensional FE method.

N	$a = 10^2$					$a = 10^4$					$a = 10^6$				
	T_{appr}	MB	Its	T		T_{appr}	MB	Its	T		T_{appr}	MB	Its	T	
26270	3.4s	26.4	14	0.5s		3.4s	26.4	13	0.4s		3.4s	26.5	14	0.5s	
227100	46.0s	261.1	18	6.5s		46.1s	260.8	18	6.5s		46.5s	260.9	18	6.6s	

and to a random number from the interval $[0, a]$ in the right half space. Table 6 shows the time needed to compute the preconditioner, the memory consumption, the number of iterations, and the time required to obtain a relative residual below 10^{-6} .

Compared with the two-dimensional problems, the CPU time for generating the preconditioner has increased by a factor of three to four. Altogether, the same observation as in the previous tests can be made: the number of iterations depends neither on the amplitude a nor on the discretization parameter N . Furthermore, the computational effort scales almost linearly with N .

6. Conclusion. For the numerical solution of linear systems arising from the discretization of second order partial differential operators, direct or iterative schemes can be used. If N is relatively small, direct solvers will be more efficient, especially because they are robust with respect to the coefficients of the operator. For large N in two spatial dimensions, direct solvers might still be faster. However, in three dimensions the situation changes. Here, iterative solvers will be advantageous but suffer from possibly nonsmooth coefficients.

We have presented a preconditioning technique that somehow embodies the advantages of both iterative and direct solvers. It has almost linear complexity, although constants can and should be reduced, and it is robust with respect to the coefficients of the operator. Since the proposed preconditioner is explicit, it is particularly efficient if the same system with many right-hand sides has to be solved.

Acknowledgments. The author wishes to thank the referees for helpful suggestions and Michael Jung, HTW Dresden, for providing the results in the comparison with geometric multigrid.

REFERENCES

- [1] P. R. AMESTOY, I. S. DUFF, J.-Y. L'EXCELLENT, AND J. KOSTER, *A fully asynchronous multi-frontal solver using distributed dynamic scheduling*, SIAM J. Matrix Anal. Appl., 23 (2001), pp. 15–41.
- [2] O. AXELSSON AND G. LINDSKOG, *On the rate of convergence of the preconditioned conjugate gradients method*, Numer. Math. 48 (1986), pp. 499–523.
- [3] M. BEBENDORF, *Efficient inversion of Galerkin matrices of general second order elliptic differential operators*, Math. Comp., 74 (2005), pp. 1179–1199.
- [4] M. BEBENDORF, *Effiziente numerische Lösung von Randintegralgleichungen unter Verwendung von Niedrigrang-Matrizen*, Ph.D thesis, Universität Saarbrücken, 2001.
- [5] M. BEBENDORF, *Approximation of boundary element matrices*, Numer. Math., 86 (2000), pp. 565–589.
- [6] M. BEBENDORF AND R. KRIEMANN, *Fast Parallel Solution of Boundary Integral Equations and Related Problems*, Preprint 10/2004, Max-Planck-Institute MIS, Leipzig, Germany, 2004.
- [7] M. BEBENDORF AND S. RJSANOW, *Adaptive low-rank approximation of collocation matrices*, Computing, 70 (2003), pp. 1–24.
- [8] M. BEBENDORF AND W. HACKBUSCH, *Existence of $\mathcal{H}$ -matrix approximants to the inverse FE-matrix of elliptic operators with L^∞ -coefficients*, Numer. Math., 95 (2003), pp. 1–28.
- [9] M. BENZI, *Preconditioning techniques for large linear systems: A survey*, J. Comput. Phys., 182 (2002), pp. 418–477.
- [10] M. BENZI AND M. TŪMA, *A sparse approximate inverse preconditioner for nonsymmetric linear systems*, SIAM J. Sci. Comput., 19 (1998), pp. 968–994.
- [11] M. BENZI AND M. TŪMA, *A comparative study of sparse approximative inverse preconditioners*, Appl. Numer. Math., 30 (1999), pp. 305–340.
- [12] J. H. BRAMBLE, J. E. PASCIAK, AND J. XU, *Parallel multilevel preconditioners*, Math. Comp., 55 (1990), pp. 1–22.
- [13] E. CHOW, *A priori sparsity patterns for parallel sparse approximate inverse preconditioners*, SIAM J. Sci. Comput., 21 (2000), pp. 1804–1822.
- [14] E. CHOW AND Y. SAAD, *Approximate inverse preconditioners via sparse-sparse iterations*, SIAM J. Sci. Comput., 19 (1998), pp. 995–1023.
- [15] PH. GUILLAUME, A. HUARD, AND C. LE CALVEZ, *A block constant approximate inverse for preconditioning large linear systems*, SIAM J. Matrix Anal. Appl., 24 (2003), pp. 822–851.
- [16] L. GRASEDYCK AND W. HACKBUSCH, *Construction and arithmetics of $\mathcal{H}$ -matrices*, Computing, 70 (2003), pp. 295–334.
- [17] A. GREENBAUM, *Iterative Methods for Solving Linear Systems*, Frontiers Appl. Math. 17, SIAM, Philadelphia, 1997.
- [18] M. J. GROTE AND T. HUCKLE, *Parallel preconditioning with sparse approximate inverses*, SIAM J. Sci. Comput., 18 (1997), pp. 838–853.
- [19] W. HACKBUSCH, *A sparse matrix arithmetic based on $\mathcal{H}$ -matrices. I. Introduction to $\mathcal{H}$ -matrices*, Computing, 62 (1999), pp. 89–108.
- [20] W. HACKBUSCH, *Multi-Grid Methods and Applications*, Springer, Berlin, 1985.
- [21] W. HACKBUSCH AND B. N. KHOROMSKIJ, *A sparse $\mathcal{H}$ -matrix arithmetic: General complexity estimates*, J. Comput. Appl. Math., 125 (2000), pp. 479–501.
- [22] W. HACKBUSCH AND B. N. KHOROMSKIJ, *A sparse $\mathcal{H}$ -matrix arithmetic. II. Application to multi-dimensional problems*, Computing, 64 (2000), pp. 21–47.
- [23] L. YU. KOLOTILINA AND A. YU. YEREMIN, *Factorized sparse approximate inverse preconditionings. I. Theory*, SIAM J. Matrix Anal. Appl., 14 (1993), pp. 45–58.
- [24] L. YU. KOLOTILINA, A. A. NIKISHIN, AND A. YU. YEREMIN, *Factorized sparse approximate inverse preconditionings. IV. Simple approaches to rising efficiency*, Numer. Linear Algebra Appl. 6 (1999), pp. 515–531.
- [25] R. KRIEMANN, *Parallel $\mathcal{H}$ -Matrix Arithmetics on Shared Memory Systems*, Preprint 29/2004, Max-Planck-Institute MIS, Leipzig, Germany, 2004.
- [26] V. ROKHLIN, *Rapid solution of integral equations of classical potential theory*, J. Comput. Phys., 60 (1985), pp. 187–207.
- [27] J. W. RUGE AND K. STÜBEN, *Algebraic multigrid*, in Multigrid Methods, S. F. McCormick, ed., SIAM, Philadelphia, 1987, pp. 73–130.
- [28] Y. SAAD, *Iterative Methods for Sparse Linear Systems*, PWS Publishing, Boston, 1996.
- [29] J. SCHÖBERL, *NETGEN—An advancing front 2D/3D-mesh generator based on abstract rules*, Comput. Vis. Sci., 1 (1997), pp. 41–52.
- [30] B. F. SMITH, P. E. BJØRSTAD, AND W. D. GROPP, *Domain Decomposition: Parallel Multilevel Methods for Elliptic Partial Differential Equations*, Cambridge University Press,

- Cambridge, UK, 1996.
- [31] E. TYRTYSHNIKOV, *Clusters, preconditioners, convergence*, Linear Algebra Appl., 263 (1997), pp. 25–48.
 - [32] E. TYRTYSHNIKOV, *Mosaic-skeleton approximations*, Calcolo, 33 (1996), pp. 47–57.

SUPPORT-GRAPH PRECONDITIONERS*

MARSHALL BERN[†], JOHN R. GILBERT[‡], BRUCE HENDRICKSON[§], NHAT NGUYEN[¶],
AND SIVAN TOLEDO^{||}

Abstract. We present a preconditioning technique, called *support-graph* preconditioning, and use it to analyze two classes of preconditioners. The technique was first described in a talk by Pravin Vaidya, who did not formally publish his results. Vaidya used the technique to devise and analyze a class of novel preconditioners. The technique was later extended by Gremban and Miller, who used it in the development and analysis of yet another class of new preconditioners. This paper extends the technique further and uses it to analyze a class of existing preconditioners, modified incomplete Cholesky. The paper also contains a presentation of Vaidya's preconditioners, which was previously missing from the literature.

Key words. support preconditioners, combinatorial preconditioners, support theory, support-tree preconditioners, modified incomplete factorizations

AMS subject classifications. 65F10, 65F05, 65F50, 65Y20, 05C85

DOI. 10.1137/S0895479801384019

1. Introduction. This paper presents new applications of a technique for constructing and analyzing preconditioners called *support-graph preconditioning*. Predecessors of support-graph methods can be found in the work from the late 80s by Notay, Beauwens, and collaborators in which graph-theoretic notions (principally paths) are used in the analysis of preconditioners [2, 3, 15, 16, 17]. These insights were extended by Vaidya [20], who described his work in a talk in 1991 but did not publish a paper. Vaidya used support-graph techniques to design a family of novel preconditioners based on spanning trees in graphs. Later, Gremban, Miller, and Zagha [9, 10] extended the technique and used it to construct another family of preconditioners. This paper explains the technique, extends it further, and uses it to analyze a class of *known* preconditioners for model problems. Specifically, we use the extended technique to analyze certain modified incomplete Cholesky (MICC) preconditioners (see [12]).

The principal goal of this paper is to bring these techniques to the attention of a wider community of researchers. By doing so, we hope to encourage further work in this promising area. The primary original content of this paper, analyzing known preconditioners using the support-graph technique, serves several purposes. First, it shows that the techniques are more widely applicable than previously appreciated. Second, we feel that the new proofs provide useful insights into these preconditioners;

*Received by the editors January 23, 2001; accepted for publication (in revised form) by S. A. Vavasis July 18, 2005; published electronically February 16, 2006. This research was supported in part by DARPA contract DABT63-95-C-0087 and by NSF contract ASC-96-26298. The third and fourth authors were supported by the Applied Mathematics Research program, U.S. DOE, Office of Science, and performed this work at Sandia National Labs, operated for the U.S. DOE under contract DE-AC04-94AL85000. A preliminary version of this paper was presented at the Copper Mountain Conference on Iterative Methods, Copper Mountain, CO, 1998.

<http://www.siam.org/journals/simax/27-4/38401.html>

[†]Xerox Palo Alto Research Center, Palo Alto, CA 94304 (bern@parc.edu).

[‡]Department of Computer Science, University of California, Santa Barbara, CA 93106-5110 (gilbert@cs.ucsb.edu).

[§]Sandia National Laboratories, Albuquerque, NM 87185-1110 (bah@cs.sandia.gov).

[¶]Raytheon Missile Systems, 1151 E. Hermans Road, Tucson, AZ 85706 (Nhat_X_Nguyen@raytheon.com).

^{||}School of Computer Science, Tel-Aviv University, Tel Aviv 69978, Israel (stoledo@tau.ac.il).

these insights can be used to improve the preconditioners and to guide heuristics for the construction of additional preconditioners.

A secondary goal of this paper is to provide a complete presentation of the support-graph technique and of Vaidya's preconditioners. Vaidya's important contribution has never been published. Although most of the theory that he uses is presented in Gremban's Ph.D. thesis [10], Vaidya's preconditioners have not been described in any published form. We seek to rectify this situation. Our complete presentation of the support-graph technique is necessary since some important portions of the theory are missing from Gremban's thesis. Specifically, we present a formal proof of the congestion-dilation lemma and we state stronger versions of some important lemmas. We also provide detailed constructions and complete proofs for Vaidya's preconditioners, which are missing from his 1991 manuscript.

The support-graph technique analyzes a preconditioner B for a matrix A by splitting both A and B into $A = A_1 + A_2 + \cdots + A_m$ and $B = B_1 + B_2 + \cdots + B_m$. Proving that $\tau B_i - A_i$ is positive semidefinite for all i shows that $\tau B - A$ is positive semidefinite and hence that the largest finite generalized eigenvalue of the matrix pencil (A, B) is bounded by τ . The bound on the smallest generalized eigenvalue is proved by bounding the largest eigenvalue of (B, A) in the same way. The splittings of A and B are guided by their underlying graphs; often this allows us to reduce a complex problem to many problems with simple structures.

This paper has three main parts. The first part of the paper, sections 2 and 3, describes support-graph theory. The second part of the paper, sections 4 and 5, describes the preconditioners of Vaidya and of Gremban and Miller. The third part of the paper, section 6, describes support-graph analysis of MICC. Our conclusions from this research are presented in section 7.

1.1. A summary of the results. This subsection summarizes the results in this paper. We start with a brief discussion of the strengths and weaknesses of the preconditioners of Vaidya and of Gremban and Miller. We also discuss the significance of our condition-number estimates for MICC.

Vaidya proposed two classes of preconditioners. The first class, maximum-weight spanning-tree preconditioners, guarantee a condition-number bound of $O(n^2)$ for any $n \times n$ sparse diagonally dominant symmetric matrix. They can be constructed and factored at insignificant cost using relatively simple graph algorithms.

Vaidya's second class of preconditioners is based on maximum-weight spanning trees augmented with a few extra edges. They can be constructed at insignificant cost using a slightly more complex algorithm than the first class. The cost of factoring these preconditioners depends on how many edges are added to the tree. Vaidya proposes that the factorization cost be balanced with the iteration costs, and he provides balancing guidelines for some classes of matrices. This class of preconditioners guarantees that the work in the linear solver is bounded by $O(n^{1.75})$ for any sparse diagonally dominant Stieltjes matrix, and by $O(n^{1.2})$ for diagonally dominant Stieltjes matrices whose underlying graphs are planar.

The strengths of Vaidya's preconditioners, especially of his second class, are that they are general, easy to construct, and provide good condition-number bounds. For example, the work required to solve a model Poisson problem in two dimensions using Vaidya's preconditioner is $O(n^{1.2})$. This compares favorably with the $O(n^{1.25})$ work required for a solver based on a modified incomplete Cholesky factorization. Coupled with the facts that Vaidya's preconditioners are guaranteed to work well on irregular problems, and that the only numerical assumption they make is that the matrix is a diagonally dominant Stieltjes matrix, these are impressive results.

The main weaknesses of Vaidya's preconditioners are that they require a high-quality direct solver to factor the preconditioner, that balancing the preconditioner-factorization costs and the iteration costs may be a nontrivial task, and that they are not guaranteed to parallelize well.

The preconditioners that Gremban and Miller proposed are multilevel preconditioners. They are based on a hierarchical partitioning of the matrix, so they may be quite expensive to construct. The cost of preconditioning in every iteration is small, and the preconditioners parallelize well. The condition number of the preconditioned system is similar, for model problems, to the condition numbers offered by modified incomplete factorizations.

Thus, even on model problems, these preconditioners do not offer convergence rates as good as those of other multilevel preconditioners, like multigrid preconditioners. On the other hand, they are guaranteed to parallelize, so they may be preferable to incomplete factorizations on some computers. Gremban and Miller do not present condition-number bounds for important classes of matrices other than regular grids with constant coefficients. For such problems their preconditioned systems have condition-number bounds of $O(n \log n)$.

Our results of the condition-number bounds for modified incomplete factorizations are well known, but our proof provides a new perspective on why these methods work.

2. Basic support graph theory. This section describes the basic linear-algebra tools that Vaidya and Gremban and Miller have developed to analyze their preconditioners. These preconditioners are for *diagonally dominant Stieltjes matrices*, symmetric, diagonally dominant matrices with nonpositive off-diagonals. Vaidya and Gremban and Miller extended some of their results to symmetric diagonally dominant matrices with mixed off-diagonals. These extensions are specific to the preconditioners that they propose; their extensions and preconditioners are described in sections 4 and 5. Our own extensions are presented in section 3.

The number of iterations of the conjugate gradient method for the solution of systems of linear equations $Ax = b$ is bounded above by the square root of the spectral condition number $\kappa(A)$ of A . (The actual number of iterations can be significantly smaller in some cases.) The condition number is the ratio of the extreme eigenvalues of A , $\kappa(A) = \lambda_{\max}(A)/\lambda_{\min}(A)$. The conjugate gradient method can be used to solve consistent linear systems with a singular coefficient matrix A (in floating-point arithmetic, it helps to orthogonalize the search directions against the null space if A is singular). In such cases, the number of iterations is proportional to the square root of the ratio of the extreme positive eigenvalues. When a preconditioner B is used in the conjugate gradient method, the number of iterations is proportional to the square root of the ratio of the extreme finite generalized eigenvalues of the pencil (A, B) , defined below.

DEFINITION 2.1. *The number λ is a finite generalized eigenvalue of the matrix pencil (A, B) if there exists a vector $x \neq 0$ such that $Ax = \lambda Bx$ and $Bx \neq 0$. We denote the set of finite generalized eigenvalues by $\lambda_f(A, B)$.*

Henceforth whenever we refer to an eigenvalue of a matrix pencil, we mean a finite generalized eigenvalue.

To bound the amount of work in the preconditioned conjugate gradient method, we need to bound the finite eigenvalues of (A, B) . We need to prove two bounds: an upper bound on $\max \lambda_f(A, B)$ and a lower bound on $\min \lambda_f(A, B)$. We will prove the upper bound directly and the lower bound by proving an upper bound on $\max \lambda_f(B, A) = 1/\min \lambda_f(A, B)$. We therefore only need to show how to prove

upper bounds on the $\max \lambda_f(A, B)$, since the lower bound is proved in essentially the same way for the matrix pencil (B, A) .

2.1. The support lemma: Bounding eigenvalues of matrix pencils. The main tool that we use to bound $\max \lambda_f(A, B)$ is the so-called *support* of (A, B) , which is the smallest number τ such that $\tau B - A$ is positive semidefinite. Informally, we think of τ as the number of copies of B required to “support” the action of A . If τ is small, B supports A well; if τ is large, B supports A weakly. We denote the support of (A, B) by $\sigma(A, B)$,

$$\sigma(A, B) = \min\{\tau : \tau B - A \text{ is positive semidefinite}\} .$$

If there is no τ for which $\tau B - A$ is positive semidefinite, then we take $\sigma(A, B) = \infty$.

The following lemma shows that the support of a pencil bounds its eigenvalues. The lemma is used implicitly by Vaidya without a proof. Gremban and Miller state the lemma and give a proof [10, Lemma 4.4]. We state it under a weaker hypothesis than Gremban and Miller, which is nevertheless strong enough for Gremban and Miller’s proof. A more general version of this lemma can be found in Axelsson [1, Theorem 10.1].

LEMMA 2.2 (support lemma [10]). *If $\lambda \in \lambda_f(A, B)$ where B is positive semidefinite and $\text{null}(A) \subseteq \text{null}(B)$, then $\lambda \leq \sigma(A, B)$.*

Some matrix pencils do not have a finite support $\sigma(A, B)$. Let

$$A = \begin{pmatrix} 2 & -1 \\ -1 & 1 \end{pmatrix} \quad \text{and let} \quad B = \begin{pmatrix} 1 & 0 \\ 0 & 0 \end{pmatrix} .$$

We have $\lambda_f(A, B) = \{1\}$, but $\tau B - A$ has a negative eigenvalue for all τ . (Lemma 2.12 shows how to bound the extreme eigenvalues in some of these cases.) If a matrix pencil has finite support, however, then the support lemma is tight.

LEMMA 2.3. *If $\sigma(A, B)$ is finite, then*

$$\sigma(A, B) \in \lambda_f(A, B) .$$

Proof. The matrix $\sigma(A, B) \cdot B - A$ has a zero eigenvalue since the eigenvalues of $\tau B - A$ are continuous in τ . Therefore, there is a nonzero vector x such that

$$(\sigma(A, B) \cdot B - A)x = 0$$

or

$$Ax = \sigma(A, B) \cdot Bx . \quad \square$$

We use the support lemma to prove an upper bound τ on $\max \lambda_f(A, B)$ by proving that $\tau B - A$ is positive semidefinite. Much of the rest of the theory consists of tools to prove that a matrix is positive semidefinite.

2.2. The splitting lemma: Proving semidefiniteness by decomposition. One way to prove that a matrix is positive semidefinite is to split it into a sum of matrices and prove that each term is positive semidefinite. This lemma too was implicitly used by Vaidya and stated and proved by Gremban and Miller [10, Lemma 4.7].

LEMMA 2.4 (splitting lemma). *Let $Q = Q_1 + Q_2 + \cdots + Q_m$, where $Q_1, Q_2, \dots, Q_m$ are all positive semidefinite. Then Q is positive semidefinite.*

We first use the splitting lemma to reduce the problem of preconditioning symmetric diagonally dominant matrices to the problem of preconditioning symmetric matrices with zero row sums.

LEMMA 2.5. *Let A be a symmetric diagonally dominant matrix and let A' be the matrix with the same off-diagonal entries but with zero row sums. Let B' be a preconditioner for A' such that both $\beta B' - A'$ and $\alpha A' - B'$ are positive semidefinite and $\alpha, \beta \geq 1$. Let $B = B' + A - A'$ be a preconditioner for A (B has the same off-diagonal entries as B' and the same row sums as A). Then $\beta B - A$ and $\alpha A - B$ are positive semidefinite.*

Proof. We have

$$\begin{aligned}\beta B - A &= \beta(B' + A - A') - A \\ &= (\beta B' - A') + (\beta - 1)(A - A') .\end{aligned}$$

Both terms in the last sum are positive semidefinite: the first by the hypothesis, and the second since it is a nonnegative scalar multiple of a diagonal matrix with nonnegative entries. Similarly,

$$\begin{aligned}\alpha A - B &= \alpha A - (B' + A - A') \\ &= (\alpha A' - B') + (\alpha - 1)(A - A')\end{aligned}$$

is positive semidefinite. $\square$

The conditions $\alpha, \beta \geq 1$ do not limit the applicability of the lemma since the condition number is 1 or more. Therefore, if either α or β is less than 1, we scale B' without changing $\alpha\beta$, which is our bound on the condition number of the preconditioned system.

Using this lemma, we assume from now on that both A and B have zero row sums.

2.3. The congestion-dilation lemma: Splitting by paths in the graph.

Vaidya and Gremban and Miller split $\tau B - A$ in a special way to prove that it is positive semidefinite. We assume that A and B are symmetric. Given a symmetric matrix A , we define its underlying graph.

DEFINITION 2.6. *The underlying graph $G_A = (V_A, E_A)$ of an n -by- n symmetric matrix A is a weighted undirected graph whose vertex set is $V_A = \{1, 2, \dots, n\}$ and whose edge set is $E_A = \{(i, j) : i \neq j, A_{i,j} \neq 0\}$. The weight of an edge (i, j) is $A_{i,j}$. The weight of a vertex i is the sum of elements in row i of A .*

Let G_A be the undirected weighted graph underlying $-A$ and G_B the graph underlying $-B$. Since both A and B have zero row sums, the graph structure and the edge weights determine the matrices exactly, since all the vertex weights are 0. If the off-diagonal elements of A and B are all negative, then the edge weights in G_A and G_B are positive. Vaidya and Gremban and Miller interpret such graphs as resistive networks where the edge weight is the conductance of a resistor. They split $\tau B - A$ into $(\tau B_1 - A_1) + (\tau B_2 - A_2) + \dots + (\tau B_m - A_m)$ such that each A_i corresponds to exactly one edge in G_A , and each B_i corresponds to a path in G_B (by path we always refer to a simple path). Both the A_i 's and the B_i 's have nonpositive off-diagonals and zero row sums. Each A_i represents the entire weight of one edge, and each corresponding B_i represents a path that can contain fractions of edge weights. The endpoints of the path represented by B_i are the endpoints of the edge represented by A_i . An example of such a splitting is shown in Figure 2.1. Both Vaidya and Gremban and Miller use

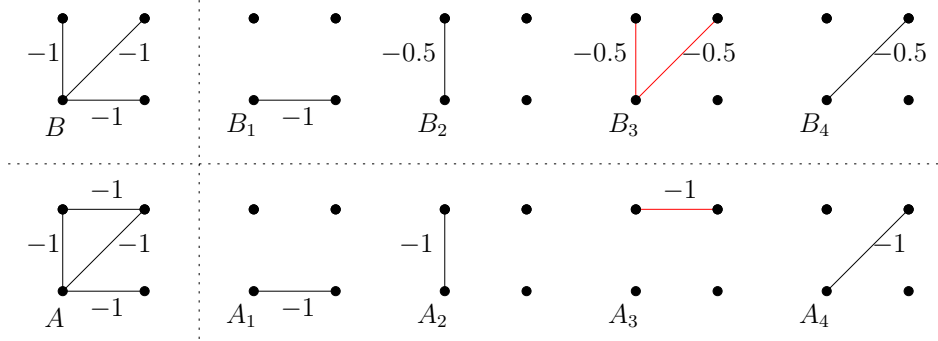


FIG. 2.1. A graph representation of a splitting of $A = A_1 + \dots + A_4$ and $B = B_1 + \dots + B_4$ such that each A_i represents a single edge and each B_i is a path that supports the edge A_i . This splitting proves that $\max \lambda_f(A, B) \leq \sigma(A, B) \leq 4$ since the worst congestion-dilation product is $2 \cdot 2 = 4$.

the congestion-dilation lemma, which they neither state nor prove, to show that each term $\tau B_i - A_i$ is positive semidefinite.

We prove the congestion-dilation lemma in three steps.

LEMMA 2.7. *Let*

$$A = \begin{pmatrix} a & 0 & \cdots & 0 & -a \\ 0 & 0 & & & 0 \\ \vdots & & \ddots & & \vdots \\ 0 & & & 0 & 0 \\ -a & 0 & \cdots & 0 & a \end{pmatrix}$$

and

$$B = \begin{pmatrix} a & -a & & & \\ -a & 2a & -a & & \\ & & \cdots & & \\ & & -a & 2a & -a \\ & & & -a & a \end{pmatrix}$$

be $(k+1)$ -by- $(k+1)$ matrices with $a > 0$. Then $kB - A$ is positive semidefinite.

Proof. We prove that $kB - A$ is positive semidefinite by showing that the matrix

$$C = (1/a)(kB - A) = \begin{pmatrix} k-1 & -k & & & 1 \\ -k & 2k & -k & & \\ & & \cdots & & \\ & & -k & 2k & -k \\ 1 & & & -k & k-1 \end{pmatrix}$$

is positive semidefinite.

We show by induction that C is positive semidefinite by performing symmetric Gaussian elimination on rows/columns 2 through $k-1$. The inductive claim is that after we eliminate row and column i (or before we eliminate $i+1$, when $i=1$), the

matrix C becomes

$$\begin{pmatrix} -1 + k/i & & & -k/i & & & 1 \\ & 2k & & & & & \\ & & 3k/2 & & & & \\ & & & ik/(i-1) & & & \\ -k/i & & & (i+1)k/i & -k & & \\ & & & -k & 2k & & \\ & & & & & \ddots & -k \\ 1 & & & & & -k & k-1 \end{pmatrix}.$$

The claim is true before we eliminate row and column 2, since in that case $i = 1$. Assume that the claim is true after we eliminate i but before $i+1$. The elimination of row and column $i+1$ modifies three entries in C : C_{11} , $C_{i+2,i+2}$, and $C_{1,i+2} = C_{i+2,1}$. The new values are

$$C_{11} = -1 + \frac{k}{i} - \frac{(-k/i)(-k/i)}{(i+1)k/i} = -1 + \frac{k}{i+1},$$

$$C_{i+2,i+2} = 2k - \frac{-k \cdot -k}{(i+1)k/i} = \frac{k(i+2)}{i+1},$$

and

$$C_{1,i+2} = 0 - \frac{-k(-k/i)}{(i+1)k/i} = \frac{k}{i+1},$$

which proves the inductive claim.

Therefore, after we eliminate row k the 2×2 submatrix consisting of the first and last row and column becomes

$$\begin{pmatrix} -1 + k/k & 1 - k/k \\ 1 - k/k & -1 + k/k \end{pmatrix} = 0$$

and the remainder of the matrix is positive diagonal, so C is positive semidefinite. $\square$

The combinatorial interpretation of Lemma 2.7 is that A represents a single edge with weight a and B represents a path with the same endpoints that consists of edges of weight a . The lemma states that the support of A in B is k , the *dilation* of the edge in the path, or simply the length of the path.

The next lemma is slightly more general.

LEMMA 2.8. *Let*

$$A = \begin{pmatrix} a & 0 & \cdots & 0 & -a \\ 0 & 0 & & & 0 \\ \vdots & & \ddots & & \vdots \\ 0 & & & 0 & 0 \\ -a & 0 & \cdots & 0 & a \end{pmatrix}$$

and

$$B = \begin{pmatrix} b & -b & & & \\ -b & 2b & -b & & \\ & & \cdots & & \\ & & & -b & 2b & -b \\ & & & & -b & b \end{pmatrix}$$

be $(k+1)$ -by- $(k+1)$ matrices with $a > 0$, $b > 0$. Then $(k \cdot a/b)B - A$ is positive semidefinite.

Proof. This case reduces by scaling to Lemma 2.7. $\square$

This lemma states that in the more general case in which the weight of the edge represented by A and the weight of the edges of the path represented by A are not the same, the support is the dilation k multiplied by the ratio a/b of the edge weights.

Finally, we state and prove the full congestion-dilation lemma.

LEMMA 2.9 (congestion-dilation lemma). *Let*

$$A = \begin{pmatrix} a & 0 & \cdots & 0 & -a \\ 0 & 0 & & & 0 \\ \vdots & & \ddots & & \vdots \\ 0 & & & 0 & 0 \\ -a & 0 & \cdots & 0 & a \end{pmatrix}$$

be a $(k+1)$ -by- $(k+1)$ matrix with $a > 0$ and let

$$B = \begin{pmatrix} d_1 & -b_1 & & & \\ -b_1 & d_2 & -b_2 & & \\ & & \cdots & & \\ & & & -b_{k-1} & d_k & -b_k \\ & & & & -b_k & d_{k+1} \end{pmatrix}$$

be a matrix with zero row sums, and with $d_i, b_i > 0$ for all i . Then $(k \cdot a / \min(b_i))B - A$ is positive semidefinite.

Proof. Let $b = \min(b_i)$. We split B into

$$B = B_1 + B_2 = \begin{pmatrix} b & -b & & & \\ -b & 2b & -b & & \\ & & \cdots & & \\ & & & -b & 2b & -b \\ & & & & -b & b \end{pmatrix} + \begin{pmatrix} d_1 - b & -b_1 + b & & & \\ -b_1 + b & d_2 - 2b & -b_2 + b & & \\ & & \cdots & & \\ & & & -b_{k-1} + b & d_k - 2b & -b_k + b \\ & & & & -b_k + b & d_{k+1} - b \end{pmatrix}.$$

The matrix B_2 is symmetric, diagonally dominant, and has nonpositive off-diagonals, so it is positive semidefinite. We have

$$\begin{aligned} (k \cdot a / \min(b_i))B - A &= (k \cdot a/b)B - A \\ &= [(k \cdot a/b)B_2] + [(k \cdot a/b)B_1 - A]. \end{aligned}$$

The first term is positive semidefinite since B_2 is positive semidefinite, and the second term is positive semidefinite by Lemma 2.8. Therefore, the sum is positive semidefinite. $\square$

The combinatorial interpretation of the congestion-dilation lemma is that $a/\min(b_i)$ is the *congestion* of the edge represented by A in the path represented by B , and k is the *dilation* of the edge. In the example depicted in Figure 2.1 the congestion of the edge A_3 in the path B_3 is 2 and the dilation is 2, for example.

The proof shows that the congestion-dilation bound $k \cdot (a/\min(b_i))$ on $\sigma(A, B)$ is tight only when all the edges along the path have the same weights; at the other extreme when one edge has small weight b and the rest have very large weights, the actual support $\sigma(A, B)$ is closer to a/b than to $k \cdot (a/b)$.

The support, splitting, and congestion-dilation lemmas are the only linear-algebra tools that Vaidya uses in his construction. Given a diagonally dominant Stieltjes matrix A , Vaidya constructs a preconditioner B whose underlying graph G_B consists of a subset of the edges of G_A and the same set of vertices. Vaidya uses the lemmas above to bound the condition number of the preconditioned system. He splits G_B into paths that support each edge of G_A . Since G_B is a subset of G_A , G_A supports the edges of G_B with paths of length 1 and unit congestion, so the smallest eigenvalue of (A, B) is at least 1. The bound that Vaidya obtains, therefore, is the worst congestion-dilation product for the edges of G_A . The specific constructions that Vaidya proposes are described in section 4.

The support, splitting, and congestion-dilation lemmas are analytical tools that are used to bound the condition number of a preconditioned system. Hence, they can also be used when one can prove that a low congestion-dilation embedding exists but cannot construct it explicitly. Gremban and Miller show how to construct and analyze preconditioners for some problems; they use congestion-dilation condition-number bounds using implicit embeddings [10, Theorem 5.1].

2.4. The clique-star lemma. Gremban and Miller introduce another way of bounding the support of one simple matrix by another. The matrix A being supported represents a fully connected subgraph, or a *clique*, of size k , and the supporting matrix B represents a k -edge star whose endpoints coincide with the members of the clique.

LEMMA 2.10 (clique-star lemma). *Let*

$$A = \begin{pmatrix} 0 & 0 & 0 & \cdots & 0 \\ 0 & (k-1)a & -a & \cdots & -a \\ 0 & -a & (k-1)a & \cdots & -a \\ \vdots & & \ddots & & \vdots \\ 0 & -a & -a & \cdots & (k-1)a \end{pmatrix}$$

and

$$B = \begin{pmatrix} kb & -b & -b & \cdots & -b \\ -b & b & 0 & \cdots & 0 \\ -b & 0 & b & \cdots & 0 \\ \vdots & & \ddots & & \vdots \\ -b & 0 & 0 & \cdots & b \end{pmatrix}$$

be $(k+1)$ -by- $(k+1)$ matrices with $a > 0$, $b > 0$. Then $(k \cdot a/b)B - A$ is positive semidefinite.

Proof. Let $C = (k \cdot a/b)B - A$. We have

$$\begin{aligned} \frac{1}{a}C &= \begin{pmatrix} k^2 & -k & -k & \cdots & -k \\ -k & k - (k-1) & 1 & \cdots & 1 \\ -k & 1 & k - (k-1) & \cdots & 1 \\ \vdots & & \ddots & & \vdots \\ -k & 1 & 1 & \cdots & k - (k-1) \end{pmatrix} \\ &= \begin{pmatrix} k^2 & -k & -k & \cdots & -k \\ -k & 1 & 1 & \cdots & 1 \\ -k & 1 & 1 & \cdots & 1 \\ \vdots & & \ddots & & \vdots \\ -k & 1 & 1 & \cdots & 1 \end{pmatrix}. \end{aligned}$$

After one step of symmetric Gaussian elimination on the first row and column, the matrix $(1/a)C$ becomes

$$\begin{pmatrix} k^2 & 0 & 0 & \cdots & 0 \\ 0 & 0 & 0 & \cdots & 0 \\ 0 & 0 & 0 & \cdots & 0 \\ \vdots & & \ddots & & \vdots \\ 0 & 0 & 0 & \cdots & 0 \end{pmatrix},$$

which is clearly positive semidefinite. Hence, C is positive semidefinite. $\square$

This lemma provides a stronger bound on the support than the bound that results from splitting the clique into edges and the star into 2-edge paths that support the edges.

2.5. Gaussian elimination and action support: Preconditioning in a larger space. Gremban and Miller construct their preconditioners in a space of higher dimension (or on a graph with more vertices) than the original matrix. They introduced a few more linear-algebra tools into the support-graph theory to deal with this generalization.

The next lemma, stated under a stronger hypothesis by Gremban and Miller [10, Lemma 4.9], shows that $\lambda_f(A, B)$ is invariant under nonsingular transformations that are applied to both A and B . Their proof holds for this more general statement.

LEMMA 2.11. *If G and H are nonsingular matrices (not necessarily symmetric), then*

$$\lambda_f(A, B) = \lambda_f(GAH, GBH).$$

We define the *action support* $\bar{\sigma}(A, B)$ of (A, B) as

$$\bar{\sigma}(A, B) = \min \{ \tau : x^T(\tau B - A)x \geq 0 \text{ for all } x \text{ such that } Ax \neq 0 \text{ and } Bx \neq 0 \}.$$

Roughly speaking, $\bar{\sigma}(A, B)$ measures how well B supports A outside their null spaces. Gremban and Miller do not use this lemma but use a similar one that is tailored more precisely to their technique. We state and prove here the more general case, which is a stronger version of the support lemma.

LEMMA 2.12. *If $\lambda \in \lambda_f(A, B)$, where $A \neq 0, B \neq 0$ are symmetric positive semidefinite, then $\lambda \leq \bar{\sigma}(A, B)$.*

Proof. Let $\tau = \bar{\sigma}(A, B)$. Note that the assumption $A \neq 0$ implies $\tau > 0$. Assume for contradiction that there is a $\lambda \in \lambda_f(A, B)$ such that $\lambda > \tau$, and let $\epsilon = \lambda - \tau > 0$. Let y be an eigenvector corresponding to λ . We have $Ay = \lambda By$, and thus $y^T(A - \lambda B)y = 0$. By definition of $\lambda_f()$, $By \neq 0$. If $By \neq 0$ and $Ay = 0$, then $\lambda = 0 < \tau$, a contradiction. So neither Ay nor By can be zero. By the definition of τ we have

$$\begin{aligned} 0 &\leq y^T(\tau B - A)y \\ &= y^T((\lambda - \epsilon)B - A)y \\ &= y^T(\lambda B - A)y - y^T(\epsilon B)y \\ &= -y^T(\epsilon B)y. \end{aligned}$$

Since B is symmetric positive semidefinite and $By \neq 0$, $y^T B y > 0$. So $-y^T(\epsilon B)y < 0$, which is a contradiction. $\square$

These two lemmas allow us to use a preconditioner in a space of higher dimension than the original matrix. We embed the original k -by- k matrix A_{11} in an n -by- n matrix A , where n is the order of the preconditioner B .

$$(2.1) \quad A = \begin{pmatrix} A_{11} & 0 \\ 0 & 0 \end{pmatrix}, \quad B = \begin{pmatrix} B_{11} & B_{12} \\ B_{12}^T & B_{22} \end{pmatrix}.$$

We cannot use congestion-dilation arguments directly to bound $\sigma(B, A)$; the underlying graph of A has $n - k$ disconnected vertices, so paths in A cannot support all the edges of B .

Instead, we use Lemma 2.11 to reduce (2.1) to a simpler case,

$$(2.2) \quad A = \begin{pmatrix} A_{11} & 0 \\ 0 & 0 \end{pmatrix}, \quad \tilde{B} = \begin{pmatrix} \tilde{B}_{11} & 0 \\ 0 & \tilde{B}_{22} \end{pmatrix}$$

using Gaussian elimination on the last $n - k$ rows and columns of B . Note that the Gauss transformations must also be applied to A , but they have no effect on it. We complete the analysis of (2.2) using Lemma 2.12. For any τ , the space $\mathbb{R}^n$ can be decomposed into two orthogonal subspaces that are invariant under $\tau A - B$. Vectors in one subspace V_1 (represented in the standard basis) have nonzeros only in the first k elements, and vectors in the other subspace V_2 have nonzeros only in the last $n - k$ elements. The subspace V_2 is contained in the null space of A . Therefore, to prove that $\max \lambda_f(B, A)$ is bounded by τ , we only need to show that $x^T(\tau A - B)x \geq 0$ for $x \in V_1$, which is equivalent to showing that $\tau A_{11} - B_{11}$ is positive semidefinite.

We will see in section 5 how Gremban and Miller use this technique to analyze the condition number of their preconditioners. The main drawback of this technique is that the elimination of the last $n - k$ rows and columns of B can significantly fill the leading k -by- k block of B . Unless B is particularly simple, this fill is difficult to analyze. We propose in the next section an alternative technique that leads to a simpler analysis of some preconditioners, since it does not require a complete elimination of the trailing block of B .

3. Support-graph theory: Extensions. We now describe *new* tools that extend the support-graph theory developed by Vaidya and Gremban and Miller. (Lemma 2.12, too, is an extension of a result of Gremban and Miller's). In particular, these tools enable or simplify the analysis of preconditioners with both positive and negative off-diagonal entries using support-graph theory. The results presented in section 3.2 were also derived independently by Guattery [11].

3.1. Stepwise Gaussian elimination. The first technique allows us to analyze support graphs that are larger than A , like Gremban and Miller's preconditioners, but without performing a complete elimination of the extra vertices that are in B but not in A . Our technique relies on the following two lemmas. The first lemma is a version of Lemma 2.11 but for action support rather than eigenvalues.

LEMMA 3.1. *Let G be a nonsingular matrix. Then $\bar{\sigma}(A, B) = \bar{\sigma}(G^T AG, G^T BG)$.*

Proof. Let $\tau = \bar{\sigma}(A, B)$. By the definition of $\bar{\sigma}(A, B)$ we have

$$x^T(\tau G^T BG - G^T AG)x = (Gx)^T(\tau B - A)(Gx) \geq 0$$

for all x such that $A(Gx) \neq 0$ and $B(Gx) \neq 0$. But $A(Gx) \neq 0$ if and only if $(G^T AG)x \neq 0$, and similarly for $G^T BG$. Therefore, we have

$$x^T(\tau G^T BG - G^T AG)x \geq 0$$

for all x such that $G^T AGx \neq 0$ and $G^T BGx \neq 0$, so $\bar{\sigma}(G^T AG, G^T BG) \leq \tau = \bar{\sigma}(A, B)$. The opposite inequality is proved in the exactly the same way. $\square$

The second lemma shows that we can subtract certain positive semidefinite matrices from A without increasing $\bar{\sigma}(A, B)$. Subtracting a positive semidefinite matrix C from A makes A easier to support, provided C 's null space includes A 's.

LEMMA 3.2 (shifting lemma). *Let A , B , and C be positive semidefinite matrices such that $\text{null}(A) \subseteq \text{null}(C)$. Then*

$$\bar{\sigma}(A - C, B) \leq \bar{\sigma}(A, B) .$$

Proof. Let $\tau = \bar{\sigma}(A, B)$. By the definition of $\bar{\sigma}(A, B)$ we have

$$x^T(\tau B - A)x \geq 0$$

for all x such that $Ax \neq 0$ and $Bx \neq 0$. Therefore, we also have

$$x^T(\tau B - (A - C))x = x^T(\tau B - A)x + x^T Cx \geq 0$$

for all x such that $Ax \neq 0$ and $Bx \neq 0$. Assume for contradiction that

$$x^T(\tau B - (A - C))x < 0$$

for some x such that $(A - C)x \neq 0$ and $Bx \neq 0$. Since $x^T Cx \geq 0$ for all x , we must have

$$x^T(\tau B - A)x < 0 ,$$

so $Ax = 0$. But $\text{null}(A) \subseteq \text{null}(C)$ implies $Cx = 0$, contradicting $(A - C)x \neq 0$. Hence,

$$\bar{\sigma}(A - C, B) \leq \tau = \bar{\sigma}(A, B) . \quad \square$$

These lemmas allow us to reduce (2.1) to (2.2) in a series of phases. In each phase, we perform one or more steps of Gaussian elimination on A , followed by a subtraction of a negative semidefinite matrix from A (addition of a positive semidefinite matrix). After all of these steps are complete, we prove a bound on the action support of the resulting matrices. These lemmas allow us to then retrace the steps that we have taken, performing transformations that reverse the effects of elimination steps and subtracting positive semidefinite matrices, all without changing the bound on the action support. Thus, the bound that we prove on the matrices after this series of transformations is also a bound on the original matrix and preconditioner.

3.2. Positive off-diagonal elements. Positive off-diagonal entries in B require a modification to the splitting strategy. Recall that the “canonical” use of the splitting lemma is to split $\tau B - A$ into $(\tau B_1 - A_1) + \cdots + (\tau B_m - A_m)$, where the B_i ’s are positive semidefinite. When the congestion-dilation lemma is used, the B_i ’s are usually paths of negative edges and the A_i ’s are negative edges, all with zero row sums (positive row sums can be handled separately as shown in Lemma 2.5). When B has positive off-diagonal entries, this strategy must be modified. One way to prove an upper bound on $\bar{\sigma}(A, B)$ in this case using the congestion-dilation framework is to support both A and the positive edges of B with the negative edges of B . We split $\tau B - A$ into

$$\begin{aligned} \tau B - A &= (\tau B_1 - A_1) + \cdots + (\tau B_m - A_m) \\ &\quad + (\tau B_{m+1} + \tau B_{m+k+1}) + \cdots + (\tau B_{m+k} + \tau B_{m+2k}), \end{aligned}$$

where each of B_{m+k+1} through B_{m+2k} represents a single positive edge, and B_1 through B_{m+k} represent paths of negative edges. Thus, B_1 through B_m support edges of A , while B_{m+1} through B_{m+k} support the positive edges of B .

An important point is that, in this case, increasing τ does not necessarily make all the terms positive semidefinite. Indeed, if $\tau B_{m+j} + \tau B_{m+k+j}$ is indefinite or negative definite, then it remains so for all $\tau \geq 0$. In other words, each positive edge of B must be supported by a path with support at most 1.

We can make the analysis simpler when the preconditioner B can be represented as $B = A - R$, where R is also positive semidefinite. Some common preconditioners that are produced by an incomplete factorization process can be represented in this way, as explained in section 6. The following lemma shows how to simplify the analysis.

LEMMA 3.3. *Let $B = A - R$ such that A , B , and R are positive semidefinite. If $\bar{\sigma}(R, A) = \tau' < 1$, then $\kappa(B^{-1}A) \leq 1/(1 - \tau')$.*

Proof. Let $\tau = 1/(1 - \tau')$. The matrix

$$\begin{aligned} \tau B - A &= \tau A - \tau R - A \\ &= (\tau - 1)A - \tau R \\ &= \frac{\tau'}{1 - \tau'}A - \frac{1}{1 - \tau'}R \end{aligned}$$

is positive semidefinite since $\bar{\sigma}(R, A) = \tau'$, so $\bar{\sigma}(A, B) \leq \tau$. We also have $\bar{\sigma}(B, A) \leq 1$, since $A - B = A - (A - R) = R$ is positive semidefinite. $\square$

In such cases, the lemma can be interpreted as an application of the strategy described in the previous paragraph. We use a τ' fraction of the negative edges of B to support the positive edges. The negative edges of B are exactly the edges of A . Therefore, we use a $1 - \tau'$ fraction of each edge of A to support itself, giving a support bound of $1/(1 - \tau')$.

4. Vaidya’s preconditioners. In this section we describe the two families of combinatorial preconditioners developed by Vaidya [20]. Vaidya’s first family applies to all symmetric, diagonally dominant matrices; the second family applies only to diagonally dominant Stieltjes matrices, but Vaidya suggests that it can be extended to all symmetric diagonally dominant matrices. We begin the discussion by assuming that A is an n -by- n diagonally dominant Stieltjes matrix and describe the extension to symmetric diagonally dominant matrices later. Let m be the number of off-diagonal nonzeros in A . If A has rows with positive row sums we increase the diagonal elements of the preconditioner B so that A and B have the same row sums. As shown in

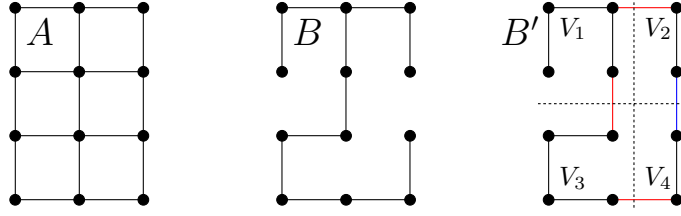


FIG. 4.1. A graph G_A , a spanning tree G_B of G_A (middle), and a spanning tree augmented with extra edges ($G_{B'}$, right). The augmentation is performed by cutting the tree into subgraphs $V_1, \dots, V_4$ of roughly equal size, and adding the heaviest edge between each pair of subgraphs.

Lemma 2.5 in section 2, this transformation does not change the condition-number bounds (although it may change the condition number itself).

Both families use preconditioners B whose underlying graphs G_B are subgraphs of G_A (using the same set of vertices and a subset of the edges), so we can always support an edge of G_B by the corresponding edge of G_A . Therefore, the congestion-dilation bound for $\sigma(B, A)$, and hence for $\max \lambda_f(B, A)$, is 1.

Vaidya's first preconditioner is constructed by finding a maximum-weight spanning tree T in G_A . In other words, T is a connected graph with no cycles (i.e., a spanning tree), and the total weight of its edges is maximal among all spanning trees of G_A . As illustrated in Figure 4.1, the preconditioner B is the diagonally dominant Stieltjes matrix whose underlying graph is $G_B = T$, and whose row sums are identical to those of A .

Let us analyze the congestion and dilation in T for an edge e of G_A . Since T is a maximum-weight spanning tree, there is exactly one path in T between the endpoints of e . Furthermore, all the edges along the path have edges at least as heavy as e . There are at most $m/2$ edges in G_A , where n is the order of A , so T is split into at most $m/2$ paths. We allocate a $2/m$ fraction of the weight of each edge of T to every path, so the congestion of an edge-path pair is at most $2/m$. The maximum length of a path is $n - 1$, so the dilation is at most $n - 1$. Hence, the congestion-dilation product for edge-path pairs is at most $m(n - 1)/2 = O(mn)$. By Lemmas 2.4 and 2.9, $\sigma(A, B) \leq O(mn)$, and by Lemma 2.2, $\lambda \leq O(mn)$ for any $\lambda \in \lambda_f(A, B)$. Since the smallest positive eigenvalue of (A, B) is at least 1, we have $\lambda_f(A, B) \subseteq [1, O(mn)]$.

Computing B takes at most $O(m \log n)$ work, using an efficient minimum-weight spanning-tree algorithm. Since the underlying graph of B is a tree, B can be factored in time $O(n)$ without producing any fill. Consequently, the costs associated with constructing and factoring B are insignificant relative to the cost of the iterative linear solver, and the cost of applying it in every iteration is $O(n)$, which is no more expensive than multiplying by A .

The maximum-weight spanning-tree preconditioners can be extended to handle any symmetric diagonally dominant matrix by taking G_B to be a maximum-weight basis for G_A rather than a maximum-weight spanning tree [4]. A maximum-weight basis is a generalization of the maximum spanning tree; see [7, section 17.4 and Problem 17-2] for background. We omit the details from this paper.

Vaidya's second family of preconditioners achieves a better condition number, but it is also more expensive to compute and apply. The construction, also illustrated in Figure 4.1, starts with the same maximum-weight spanning tree T . Let t be an integer parameter. We decompose G_A into a set of t subgraphs $V_1, V_2, \dots, V_t$ such that each subgraph is spanned by a connected subgraph of T and has at most m/t vertices. We

form G_B by adding to T the heaviest edge between V_i and V_j for all i and j . We add nothing if there are no edges between V_i and V_j or if the heaviest edge is already in T ; note that if V_i and V_j are connected by an edge in T , they remain connected by the same edge in G_B , since G_B is formed by simply adding edges to T . To analyze this preconditioner, we need the additional assumption that no row in A has more than $d + 1$ nonzeros for some constant d , which implies that $m \leq dn$. The details of the partitioning of T into V_i 's is given in [6, 4]. A more sophisticated partitioning algorithm, in which the quality of the decomposition is given by the average row density rather than the maximum row density, is given by Spielman and Teng [19]. We decompose the augmented tree G_B into a set of paths as follows. If both endpoints of an edge $e \in G_A$ are in the same V_i , we use the single path in T that connects them. If one endpoint belongs to V_i and the other to V_j , the path uses T to get from one endpoint of e to the heaviest edge that connects V_i to V_j , then the path uses this edge, and finally it uses T to get to the other endpoint of e . Again, the edges along a path are all at least as heavy as the edge that is supported by the path. Since the paths now have length at most $1 + (2dn/t)$, and since each edge in G_B carries at most d^2n/t paths, the condition number of (A, B) is bounded by $O(n^2/t^2)$.

What is the cost of factoring B ? Let us denote the endpoints of the edges that connect V_i with the other V_j 's by U_i . Since the V_i 's are disjoint, we have $U_i \cap U_j = \emptyset$. We begin the factorization of B by eliminating all the degree-1 and degree-2 vertices in B , until all the remaining vertices have degree greater than 2. This phase, which we refer to as *contraction*, requires only $O(n)$ work and generates only $O(n)$ fill elements. Once this is done, what remains of each V_i is a tree with no vertices of degree 1 or 2, and whose leaves are all in U_i (these are leaves in V_i but not in B or A). It follows that the number of nonleaf vertices in V_i is at most $|U_i|$. Hence, the total number of vertices in the contracted graph is at most $2(|U_1| + \cdots + |U_t|)$. We now factor the contracted graph, exploiting whatever structure it has; for example, if it is planar, we can use nested dissection to factor it. Hence, the total cost of factoring B is $O(n)$ plus the cost of factoring the contracted graph.

In the worst case, each subgraph V_i has a connection to all others and has $(t - 1)$ vertices in U_i . In this case, the contracted graph has no more than $2 \cdot t(t - 1)$ vertices, so factoring it requires at most $O((t^2)^3)$ work.

When G_A is planar, G_B is planar and so is the contracted graph. Furthermore, when G_A is planar, the contracted graph has only $O(t)$ vertices. This can be proved by contracting each V_i to a single vertex, which still preserves planarity. Since G_B has at most one edge between V_i and V_j , these edges do not disappear and are not merged into other edges in the contraction process. This supercontracted planar graph has only t vertices, so it has only $O(t)$ edges. But each one of the edges between the original subgraphs V_i, V_j is still represented in the supercontracted graph, so there are only $O(t)$ of them in B , which proves that there are only $O(t)$ vertices in $|U_1| + \cdots + |U_t|$. The factors of a matrix whose underlying graph is a planar graph with $O(t)$ vertices have $O(t \log t)$ nonzeros and the factorization can be performed in $O(t^{3/2})$ time.

The following lemma summarizes the discussion above. The result is used in Vaidya's manuscript but without a proof.

LEMMA 4.1. *The cost of factoring the augmented maximum-weight spanning-tree preconditioner B is $O(n + t^6)$ when A is a general diagonally dominant Stieltjes matrix, and $O(n + t^{1.5})$ when A is planar. The factors of B have $O(n + t^4)$ nonzeros in the general case, and $O(n + t \log t)$ when A is planar.*

The choice of t should balance the costs of constructing and factoring B with the cost of the iterations, which is determined by both the condition number and the cost

of applying B . The cost of constructing B is again insignificant. If A has no special nonzero structure (beyond a bound of $d + 1$ nonzeros per row), then the optimal t is $\Theta(n^{0.25})$. Factoring the preconditioner costs $O(t^6) = O(n^{1.5})$. The number of iterations is bounded by $O(\sqrt{n^2/t^2}) = O(n/t)$, and the cost of every iteration is $O(n + t^4)$, so the total cost is $O(n^2/t + nt^3) = O(n^{1.75})$. When A is planar, the cost of factoring B is $O(n + t^{1.5})$, and the cost of every iteration is $O(n + t \log t)$. The cost is minimized near $n^2/t = t^{1.5}$, or $t = n^{0.8}$. The total cost to solve the linear system is $O(n^{1.2})$, versus $O(n^{1.75})$ in the general case.

Vaidya analyzes other cases using other estimates of work and fill during the factorization of various classes of sparse matrices. Vaidya does not show, however, bounds for regular meshes and finite-element grids in three dimensions that are better than the $O(n^{1.75})$ bound that applies to all bounded-degree graphs.

Vaidya also proposes a recursive scheme that uses the same idea to solve the system $Bz = r$ that must be solved in every iteration. That is, instead of factoring the preconditioner B and performing two triangular solves in every iteration, Vaidya proposes to construct a preconditioner for B that is even sparser than B , and to solve $Bz = r$ iteratively. Similar ideas have been proposed in other contexts, such as domain decomposition solvers where an iterative solver can be used within each subdomain, leading essentially to a multilevel preconditioner. Vaidya does not analyze this idea in any detail.

One potential disadvantage of Vaidya's preconditioners is that they are not guaranteed to parallelize. The maximum-weight trees that are constructed may have a large diameter. The large diameter of the trees creates long chains of dependences in the triangular factors, and these chains limit the parallelism that is available within each iteration of the solver.

Chen and Toledo present experimental results with Vaidya's preconditioners in [6].

5. The preconditioners of Gremban and Miller. This section presents *support trees*, the family of preconditioners that Gremban and Miller developed. We again assume that A is symmetric and diagonally dominant, and that its off-diagonal entries are all nonpositive. (Gremban and Miller also show a technique to convert a problem with a symmetric diagonally dominant matrix to a larger problem in which the matrix has only nonpositive off-diagonals.) We will again assume that A and the preconditioner both have zero row sums.

Like Vaidya's method, Gremban and Miller's approach is essentially a graph algorithm that constructs $G_{B'}$ given G_A . However, Gremban and Miller construct a graph $G_{B'}$ with more vertices than G_A , so G_A is augmented with additional disconnected vertices so that both graphs use the same vertex set. In matrix terms, the construction embeds A as the leading block of a larger zero matrix,

$$(5.1) \quad A' = \begin{pmatrix} A & 0 \\ 0 & 0 \end{pmatrix} \quad \text{and} \quad B' = \begin{pmatrix} B_{11} & B_{12} \\ B_{12}^T & B_{22} \end{pmatrix}.$$

Gremban and Miller use the congestion-dilation lemma to bound $\sigma(A', B')$. However, there is no way to route all the edges of $G_{B'}$ in $G_{A'}$, since $G_{A'}$ is not connected. In other words, $\sigma(B, A)$ is infinite. Gremban and Miller, therefore, develop and use Lemmas 2.11 and 2.12 to eliminate the extra vertices in $G_{B'}$. Once these nodes are eliminated, they use the congestion-dilation lemma to bound $\sigma(B_{11} - B_{12}B_{22}^{-1}B_{12}^T, A)$, which provides a lower bound on the smallest positive finite eigenvalue of (A, B) .

The construction of $G_{B'}$, illustrated in Figure 5.1, is based on a hierarchical decomposition of G_A . The algorithm removes from G_A a set of edges, known as

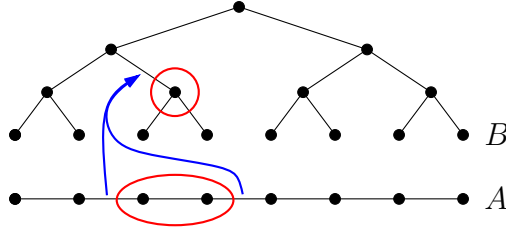


FIG. 5.1. An illustration of the preconditioners of Gremban and Miller. A is partitioned hierarchically, and the vertices of B represent subgraphs in that partition. The circled vertex of B represents the subgraph of A consisting of the two circled vertices. The weight of an edge of B is the sum of the weights of the edges of A that connect the subgraph to the rest of A .

a separator, that breaks it into a small number of subgraphs $G_1, G_2, \dots, G_k$. The algorithm then recursively partitions each G_i until the graph is decomposed into single vertices. The separator is chosen so that all the G_i 's have roughly the same number of vertices and such that the total weight of the separator is small. A variety of graph-partitioning algorithms can be used to find good edge separators (see, for example, [8]). The process is repeated until each subgraph consists of a single vertex. The graph $G_{B'}$, which is a tree, is constructed using this hierarchical decomposition. The algorithm assigns to each subgraph in the decomposition a vertex of $G_{B'}$. That is, $G_{B'}$ has one vertex that represents all of G_A , a vertex for each subgraph of G_A in the first level of the decomposition, and so on, down to vertices that represent single vertices of G_A , which are the smallest subgraphs in the decomposition. A leaf of $G_{B'}$ represents a single vertex of G_A , and is considered to be the same as that vertex of G_A . The matrices A' and B' are ordered accordingly. A vertex that represents a subgraph G_i in the decomposition is connected by edges to the subgraphs of G_i in the decomposition $G_{i_1}, G_{i_2}, \dots, G_{i_\ell}$, and to the subgraph that contains G_i in the previous level of the decomposition. The weight that is assigned to the edge that connects G_i to, say, G_{i_1} is the total weight of the edges that connect G_{i_1} to the remainder of the graph.

This construction makes it easy to prove a fairly low upper bound on $\max \lambda_f(A, B)$. We route each edge e of $G_{A'}$ along the unique path in $G_{B'}$ that connects its endpoints. Each edge in this path allocates a weight of w to support e , where w is the weight of e . This is always possible since if the path uses the edge between vertices that represent G_i and G_{i_1} in the decomposition, then e is part of the separator that divides G_{i_1} from the rest of the graph, so the w is included in the weight of each edge in the path. If every subgraph in the decomposition is split into at least k subgraphs whose sizes differ by at most a constant factor, the length of the path is $O(\log_k n)$, where n is the order of A . It follows that the congestion-dilation product is bounded by $O((w/w) \cdot \log_k n) = O(\log_k n)$, which provides an upper bound on $\max \lambda_f(A, B)$.

Proving a lower bound on $\min \lambda_f(A, B)$ is more difficult. As explained above, the bound results from applying the congestion-dilation lemma to the Schur complement $S = B_{11} - B_{12}B_{22}^{-1}B_{12}^T$ and to A . Specifically, Gremban and Miller prove upper bounds on the weights of the edges of S (which is a dense matrix) and show how to route them in A . For regular $n^{1/d} \times \dots \times n^{1/d}$ grids in d dimensions with uniform edge weight, Gremban and Miller essentially perform a symbolic elimination to bound the entries of S . The bound that is obtained on $\sigma(S, A)$ is $O(d^2 n)$, leading to an overall condition number bound $O(n \log n)$ for fixed d . They also prove similar bounds for somewhat

more general classes of matrices using some additional graph-theoretic tools. These classes of matrices do not subsume, however, matrices that represent planar graphs or finite element meshes.

Since $G_{B'}$ is guaranteed to be a tree with diameter $O(\log n)$, factoring B' and applying the factors in every iteration requires only $O(n)$ work and $O(\log n)$ parallel steps. The cost of computing $G_{B'}$ depends on the graph-partitioning algorithm that is used and may be substantial in practice.

Gremban and Miller also show how to convert a problem with a symmetric diagonally dominant matrix to a problem with a symmetric diagonally dominant matrix with nonpositive off-diagonals, so that their technique can be used. The graph of the modified problem has twice as many vertices and edges as the original graph. The modified graph represents each vertex of the original graph with two vertices and each edge with two edges. The transformation preserves separators, so if the original graph has a special structure that guarantees small separators, then the modified graph also has good separators and the same algorithm can be used to find them.

Gremban and Miller describe numerical experiments that show that their method outperforms a diagonal-scaling preconditioner and an incomplete Cholesky preconditioner. On matrices that represent two-dimensional meshes, Gremban and Miller's preconditioner performs fewer iterations and solves systems faster than the other preconditioners. On a three-dimensional problem, Gremban and Miller's preconditioner requires more iterations than incomplete Cholesky, but it leads to faster solution times on a vector computer.

6. Analysis of incomplete factorizations. The convergence rates of various incomplete factorization preconditioners have been studied extensively, starting with the work of Gustafsson [12]. For MICC (the method we will analyze below), Gustafsson's analysis required a perturbation of the diagonal of the matrix. This limitation was removed in later work by Beauwens [2, 3], Notay [16], and others. The somewhat more challenging rank deficient case was also addressed by Notay [15, 17].

In this section, we present another analysis of the performance of modified incomplete factorization preconditioners on model problems. Our results are not new, and in fact are special cases of some of the results alluded to above. But we hope that the simplicity of the analysis (particularly as it easily handles rank deficiency) serves to illustrate the utility of the tools we have developed. We should also note that Guattery [11] has performed a somewhat similar analysis of unmodified incomplete factorizations.

Let $B = LL^T$ be a level-0 modified incomplete factorization of a diagonally dominant Stieltjes matrix A . The incomplete factor L has the same nonzero structure as the lower triangle of A , and B has the same row sums as A . We can write $B = A - R$, where R consists of the fill elements that are dropped during the factorization plus the diagonal modification that is performed in order to maintain the row sums. Since the elements that are dropped are always negative and since A and B have the same row sums, R is a diagonally dominant Stieltjes matrix with zero row sums. A modified incomplete factorization of a model problem is shown in Figure 6.1.

We first analyze the MICC factorization for a two-dimensional model problem, a Laplace equation with Neumann boundary conditions on a regular grid.

Consider the regular grid depicted by the solid lines in Figure 6.1. If we perform an elimination of the vertices in the natural order and discard all fill, then the discarded values will correspond to the dashed diagonals in the figure. If A is the Laplacian matrix and B the MICC preconditioner, then $B = A - R$, where R is the matrix

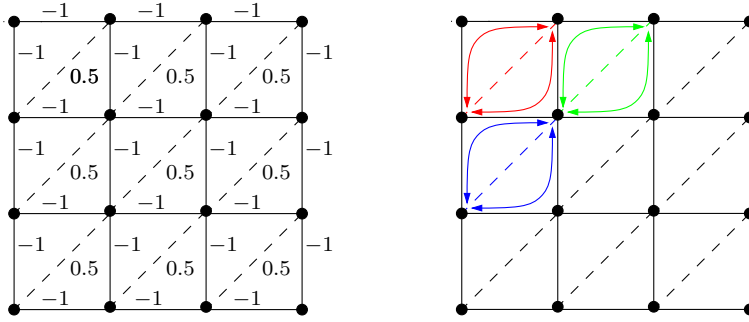


FIG. 6.1. An incomplete-factorization preconditioner for a model problem with zero row sums (left). The model problem has the same underlying graph but without the positive dashed edges. The (complete) Cholesky factor of this matrix is the incomplete Cholesky factor of the model problem. The figure on the right shows the two paths that route each positive edge.

of these discarded values. Using Lemma 3.3 we can bound $\kappa(B^{-1}A)$ by supporting R with A . The sketch on the right in Figure 6.1 shows how each entry of R can be supported by two paths of length 2 within A . If we were to divide the weight of each A edge evenly, using half to support the R edge above it or to the left and half to support the R edge below it or to the right, we would support every R edge exactly. Unfortunately, this gives $\tau' = 1$ in Lemma 3.3, which does not give a finite bound on $\kappa(B^{-1}A)$. Rather, we must realize that this even division underutilizes the A edges along the boundary of the grid, and use an uneven division that varies from the upper left to the lower right.

We can formalize this discussion to prove the following result.

THEOREM 6.1. *Let A represent a Laplace equation with Neumann boundary conditions (i.e., zero row sums) discretized on a $\sqrt{n}$ -by- $\sqrt{n}$ grid, as shown in Figure 6.1. Let B be a MICC factorization of A with no fill, using the natural (row-wise) ordering of the grid. Then $\kappa(B^{-1}A) \leq 2\sqrt{n} - 2$.*

Proof. We denote by (i, j) the vertex in row i and column j of the grid, and by $(i, j) \leftrightarrow (k, l)$ an edge connecting the vertices (i, j) and (k, l) . It is easy to see that B consists of the edges of A plus edges with weight $+1/2$ that connect $v_{i,j}$ with $v_{i+1,j-1}$, as shown in Figure 6.1. Also, the construction of a modified incomplete factorization ensures that B has the same row sums as A , namely zero.

Using Lemma 3.3, $\kappa(B^{-1}A) \leq (1/(1 - \bar{\sigma}(R, A)))$, where $R = A - B$ is the (positive semidefinite) matrix of dropped fill elements, the diagonal dashed lines in the figure. Thus we must use A to support the edges of R . Each R edge is supported by two paths of length 2 in A , as shown in Figure 6.1.

More formally, we split A and R as follows. The matrix A is split into $2(\sqrt{n} - 1)^2$ submatrices with the following edge sets, each a path of length 2:

$$\pi_{\wedge}(i, j) = \{(i, j) \leftrightarrow (i, j + 1), (i, j) \leftrightarrow (i + 1, j)\} \quad \text{for each } i < \sqrt{n}, j < \sqrt{n},$$

and

$$\pi_{\vee}(i, j) = \{(i, j) \leftrightarrow (i, j - 1), (i, j) \leftrightarrow (i - 1, j)\} \quad \text{for each } i > 1, j > 1.$$

Except along the boundary, each edge of A is divided between one $\pi_{\wedge}$ submatrix and one $\pi_{\vee}$ submatrix. The weight of an edge in $\pi_{\wedge}(i, j)$ that is allocated to that

submatrix in the splitting is

$$w_{\wedge}(i, j) = \frac{2\sqrt{n} - 2}{2\sqrt{n} - 3} - \frac{i + j - 1}{2\sqrt{n} - 3},$$

and the weight allocated to $\pi_{\vee}(i, j)$ is

$$w_{\vee}(i, j) = \frac{i + j - 3}{2\sqrt{n} - 3}.$$

By Lemma 2.8, the submatrix $\pi_{\wedge}(i, j)$ supports exactly a $w_{\wedge}(i, j)$ fraction of the R edge $(i, j+1) \leftrightarrow (i+1, j)$, and the submatrix $\pi_{\vee}(i+1, j+1)$ supports a $w_{\vee}(i+1, j+1)$ fraction of the same edge. Since

$$w_{\wedge}(i, j) + w_{\vee}(i+1, j+1) = \frac{2\sqrt{n} - 2}{2\sqrt{n} - 3},$$

we can apply Lemma 3.3 with $\tau' = (2\sqrt{n} - 3)/(2\sqrt{n} - 2)$ to conclude that

$$\kappa(B^{-1}A) = \frac{1}{1 - \tau'} = 2\sqrt{n} - 2.$$

We now show that this splitting of A is feasible; that is, that the contribution of each A edge to the paths that support R edges is not more than its weight. Each A edge contributes to either one π submatrix (if it is on the boundary of the grid) or to two (if it is in the interior). The total contribution of an interior edge, say $(i, j) \leftrightarrow (i, j+1)$, is

$$w_{\wedge}(i, j) + w_{\vee}(i, j+1) = \frac{2\sqrt{n} - 2}{2\sqrt{n} - 3} - \frac{i + j - 1}{2\sqrt{n} - 3} + \frac{i + (j+1) - 3}{2\sqrt{n} - 3} = \frac{2\sqrt{n} - 3}{2\sqrt{n} - 3} = 1.$$

The contribution of a boundary edge is at most

$$\max \left\{ \frac{2\sqrt{n} - 2}{2\sqrt{n} - 3} - \frac{1 + 1 - 1}{2\sqrt{n} - 3}, \frac{\sqrt{n} + \sqrt{n} - 3}{2\sqrt{n} - 3} \right\} = \frac{2\sqrt{n} - 3}{2\sqrt{n} - 3} = 1. \quad \square$$

It is easy to see that the same condition-number upper bound holds for the same model problem but with Dirichlet or mixed boundary conditions. The only difference in the structure of A between the Neumann boundary condition case and the Dirichlet or mixed case is that row sums for vertices on the boundary of the grid may be positive. Since B has the same row sums as A , they both can be split into a zero-row-sum matrix and a positive diagonal matrix. The diagonal parts of A and B support each other with support 1. The zero-row-sum parts are similar to the case that we analyzed, except that the positive edges in B may be smaller than 0.5 but never greater. Hence, it is easier to support them, so the same bound holds. That is, we use Lemma 2.9 rather than Lemma 2.8 in the proof above.

The following theorem formalizes this result.

THEOREM 6.2. *Let A represent a model Laplace equation with nonnegative row sums discretized on a $\sqrt{n}$ -by- $\sqrt{n}$ grid, as shown in Figure 6.1. Let B be a MICC factorization of A with no fill. Then $\kappa(B^{-1}A) \leq 2\sqrt{n} - 2$.*

Similar analyses can be performed for three-dimensional model problems, although the bookkeeping is somewhat more tedious. The bounds we have obtained this way are $O(n^{1/3})$ for Dirichlet boundary conditions and $O(n^{2/3})$ for Neumann boundary conditions. We omit the details from this paper.

7. Conclusions. Support-graph theory has already motivated the design of several novel families of preconditioners. Vaidya's preconditioners, in particular, are more general and guarantee lower condition-number bounds than modified incomplete factorizations, a widely used class of preconditioners. We show in this paper that the same theory can be used to prove tight condition-number bounds for MICC preconditioners on model problems.

Significant work has been done on support-graph preconditioning besides the results presented in this paper. Guattery used support-graph theory to bound the condition number of incomplete factorizations without diagonal modification [11]. Howle and Vavasis generalized the preconditioners of Gremban and Miller to complex systems [13]. Reif proposed and analyzed Vaidya-like preconditioners, including recursive variants [18]. Chen and Toledo conducted an experimental evaluation of Vaidya's preconditioners [6]. Boman and Hendrickson formulated a general theory of support for symmetric positive definite matrices; the conjection-dilation arguments that we used in this paper are a special case of their theory [5]. Boman et al. generalized the Vaidya preconditioners presented in this paper to all symmetric diagonally dominant matrices [4]. Spielman and Teng [19] improve Vaidya's preconditioners for diagonally dominant Stieltjes matrices by using sharper spectral bounds, by improving the tree-partitioning algorithm, and by exploiting a sophisticated spanning-tree construction algorithm. Maggs et al. show how to construct a Gremban and Miller-like preconditioner for any symmetric diagonally dominant matrix [14].

We believe that support-graph theory provides a new, largely unexploited tool for the analysis and design of preconditioners. We hope that this paper serves to make the techniques more accessible to the numerical analysis community and to stimulate further work in this promising area.

Acknowledgments. Thanks to the two anonymous referees for many helpful suggestions, corrections, and comments. Thanks to Tony Chan for many constructive discussions. Thanks also to Steve Guattery and Erik Boman for detailed comments on drafts of this paper. We are also indebted to Howard Elman and Henk van der Vorst for enlightening and helpful discussions.

REFERENCES

- [1] O. AXELSSON, *Iterative Solution Methods*, Cambridge University Press, Cambridge, UK, 1994.
- [2] R. BEAUWENS, *Upper eigenvalue bounds for pencils of matrices*, Linear Algebra Appl., 62 (1984), pp. 87–104.
- [3] R. BEAUWENS, *Approximate factorizations with modified S/P consistently ordered M-factors*, Numer. Linear Algebra Appl., 1 (1994), pp. 3–17.
- [4] E. G. BOMAN, D. CHEN, B. HENDRICKSON, AND S. TOLEDO, *Maximum-weight-basis preconditioners*, Numer. Linear Algebra Appl., 11 (2004), pp. 695–721.
- [5] E. G. BOMAN AND B. HENDRICKSON, *Support theory for preconditioning*, SIAM J. Matrix Anal. Appl., 25 (2003), pp. 694–717.
- [6] D. CHEN AND S. TOLEDO, *Vaidya's preconditioners: Implementation and experimental study*, Electron. Trans. Numer. Anal., 16 (2003), pp. 30–49.
- [7] T. H. CORMEN, C. E. LEISERSON, AND R. L. RIVEST, *Introduction to Algorithms*, MIT Press, Cambridge, MA, 1989.
- [8] U. ELSNER, *Graph Partitioning: A Survey*, Technical report SFB393/97-27, Technische Universität Chemnitz, Chemnitz, Germany, 1997.
- [9] K. D. GREMBAN, G. L. MILLER, AND M. ZAGHA, *Performance evaluation of a new parallel preconditioner*, in Proceedings of the 9th International Parallel Processing Symposium, Santa Barbara, CA, 1995, IEEE, Computer Society Press, Los Alamitos, CA, 1995, pp. 65–69.

- [10] K. D. GREMBAN, *Combinatorial Preconditioners for Sparse, Symmetric, Diagonally Dominant Linear Systems*, Ph.D. thesis, Carnegie Mellon University, Pittsburgh, PA., 1996.
- [11] S. GUATTERY, *Graph Embedding Techniques for Bounding Condition Numbers of Incomplete Factor Preconditioners*, Technical report ICASE 97-47, NASA Langley Research Center, Hampton, VA, 1997.
- [12] I. GUSTAFSSON, *A class of first-order factorization methods*, BIT, 18 (1978), pp. 142–156.
- [13] V. E. HOWLE AND S. A. VAVASIS, *An iterative method for solving complex-symmetric systems arising in electrical power modeling*, SIAM J. Matrix Anal. Appl., 26 (2005), pp. 1150–1178.
- [14] B. M. MAGGS, G. L. MILLER, O. PAREKH, R. RAVI, AND S. LEUNG MAVERICK WOO, *Finding effective support-tree preconditioners*, in Proceedings of the 17th ACM Symposium on Parallel Algorithms and Architectures, Las Vegas, NV, 2005.
- [15] Y. NOTAY, *Solving positive (semi)definite linear systems by preconditioned iterative methods*, in Preconditioned Conjugate Gradient Methods, Lecture Notes in Math. 1457, Springer-Verlag, New York, 1990, pp. 105–125.
- [16] Y. NOTAY, *Conditioning analysis of modified block incomplete factorizations*, Linear Algebra Appl., 154–156 (1991), pp. 711–722.
- [17] Y. NOTAY, *Conditioning of Stieltjes matrices by S/P consistently ordered approximate factorizations*, Appl. Numer. Math., 10 (1992), pp. 381–396.
- [18] J. H. REIF, *Efficient approximate solution of sparse linear systems*, Comput. Math. Appl., 36 (1998), pp. 37–58 (see also an errata in 38 (1999), p. 141).
- [19] D. A. SPIELMAN AND S.-H. TENG, *Solving sparse, symmetric, diagonally-dominant linear systems in time $O(m^{1.31})$* , in Proceedings of the 44th Annual IEEE Symposium on Foundations of Computer Science, Washington, DC, 2003, pp. 416–427.
- [20] P. M. VAIDYA, *Solving Linear Equations with Symmetric Diagonally Dominant Matrices by Constructing Good Preconditioners*, manuscript, 1991.

CONVERGENCE ANALYSIS OF A PAGERANK UPDATING ALGORITHM BY LANGVILLE AND MEYER*

ILSE C. F. IPSEN[†] AND STEVE KIRKLAND[‡]

Abstract. The PageRank updating algorithm proposed by Langville and Meyer is a special case of an iterative aggregation/disaggregation (SIAD) method for computing stationary distributions of very large Markov chains. It is designed, in particular, to speed up the determination of PageRank, which is used by the search engine Google in the ranking of web pages.

In this paper the convergence, in exact arithmetic, of the SIAD method is analyzed. The SIAD method is expressed as the power method preconditioned by a partial LU factorization. This leads to a simple derivation of the asymptotic convergence rate of the SIAD method.

It is known that the power method applied to the Google matrix always converges, and we show that the asymptotic convergence rate of the SIAD method is at least as good as that of the power method. Furthermore, by exploiting the hyperlink structure of the web it can be shown that the asymptotic convergence rate of the SIAD method applied to the Google matrix can be made strictly faster than that of the power method.

Key words. Google, PageRank, Markov chain, power method, stochastic complement, aggregation/disaggregation

AMS subject classifications. 15A51, 65F15, 65F10, 15A18, 15A42

DOI. 10.1137/S0895479804439808

1. Introduction. The PageRank of a web page is an important ingredient in how the search engine Google determines the ranking of web pages [4, 25]. Computing PageRank amounts to computing the stationary distribution of a stochastic matrix whose size is now in the billions [21, 22]. Langville and Meyer [12, 13, 15] propose an updating algorithm to compute stationary distributions of large Markov chains, with the aim of speeding up the computation of PageRank.

The hyperlink structure of the web can be represented as a directed graph, whose vertices are web pages and whose edges are links. From it one can construct a stochastic matrix¹ P as follows. If web page i has $d_i \geq 1$ outgoing links, then for each link from web page i to page j , entry (i, j) of the matrix P is $1/d_i$. If there is no link from page i to page j , then entry (i, j) of P is 0. If page i has no outgoing links at all (a dangling node, or sink), then all elements in row i of P are set to $1/n$, where n is the order of P [24]. Thus P is a stochastic matrix, and in order to construct a primitive stochastic matrix, one forms the convex combination

$$G = cP + (1 - c)\mathbf{1}v^T,$$

where $\mathbf{1}$ is the column vector of all ones, v is a positive probability vector, the superscript T denotes the transpose, and $0 < c < 1$ a scalar. This is the Google matrix.

*Received by the editors January 9, 2004; accepted for publication (in revised form) by I. S. Dhillon February 16, 2005; published electronically February 16, 2006.

<http://www.siam.org/journals/simax/27-4/43980.html>

[†]Center for Research in Scientific Computation, Department of Mathematics, North Carolina State University, P.O. Box 8205, Raleigh, NC 27695-8205 (ipsen@math.ncsu.edu, <http://www4.ncsu.edu/~ipsen/>). This research was supported in part by NSF grants DMS-0209931 and DMS-0209695.

[‡]Department of Mathematics and Statistics, University of Regina, Regina, SK S4S 0A2, Canada (kirkland@math.uregina.ca, <http://www.math.uregina.ca/~kirkland/>). This research was supported in part by NSERC grant OGP0138251.

¹A stochastic matrix P is a real square matrix with elements $0 \leq p_{ij} \leq 1$, whose rows sum to one, $\sum_j p_{ij} = 1$ for all i .

Element i of the stationary probability π of G represents the probability that a random surfer visits web page i and it is a major ingredient in the PageRank of page i .

Thus Google faces the problem of having to compute the stationary probability π of a very large stochastic matrix. Direct methods [29, section 2] are too time consuming. However, the matrix G is constructed to be primitive, and the power method applied to G converges, in exact arithmetic, to a multiple of π for any nonnegative starting vector [29, section 3.1]. The asymptotic convergence rate of the power method is given by the modulus of the second largest eigenvalue, in magnitude, of G . The original PageRank algorithm relied on the power method [25], but convergence can be slow [12, section 6]. Langville and Meyer present an updating algorithm to compute stationary probabilities of large Markov chains, by accelerating the power method with an aggregation/disaggregation step similar to [29, section 6.3]. We refer to their method as SIAD for *special iterative aggregation/disaggregation method* [12, 13, 15]. Numerical experiments illustrate the potential of the SIAD method in the context of PageRank computation [12, 15]. Langville and Meyer's SIAD method is a special case of more general iterative aggregation/disaggregation methods that can accommodate general aggregation and disaggregation operators [17, 18]. Here, however, we focus on the version by Meyer and Langville, whose restriction operator is an inexact stochastic complementation.

Related work. Our results for the Google matrix are built on the algorithm of Langville and Meyer [14, 12, 13, 15]. However, there are other approaches for speeding up the power method on the Google matrix. The following algorithms appear to work well in practice, but the convergence proofs, if available, are not rigorous. Extrapolation methods [2, 7, 11] construct a new iterate from a judicious linear combination of previous iterates where the methods in [2] were inspired by the rational expressions for the PageRank vector in terms of c from [28] and further developed in [3]. Adaptive methods [11, 9] exploit the fact that pages with lower page rank tend to converge faster and do not recompute these components. A two-stage algorithm [16] computes stationary probabilities associated with dangling nodes and with nondangling nodes and then combines the two. It is shown that there are starting vectors for which this algorithm converges at least as fast as the power method. A three-stage algorithm [10] which exploits block structure in the web graph to partition the matrix appears to be an iterative aggregation/disaggregation method like those in [29].

Overview. We analyze, in exact arithmetic, the asymptotic convergence of the SIAD method. In particular we show that the SIAD method amounts to a power method preconditioned by a partial LU factorization (sections 3, 4, and 5) and give a simple derivation of the asymptotic convergence rate (section 5). We consider the choice of different stochastic complements for better convergence (section 6) and computable upper bounds on the asymptotic convergence rate (section 7). It is well known that the power method applied to the Google matrix G always converges; in this paper we show that the asymptotic convergence rate of the SIAD method applied to G is at least as good as that of the power method. We consider conditions that assure convergence of the SIAD method in general (section 8). At last we prove a stronger result for convergence of the SIAD method applied to G (section 9). By exploiting the hyperlink structure of the web we conclude that the asymptotic convergence rate of the SIAD method applied to G can be made strictly better than that of the power method. We also show how one can save computations when applying the SIAD method to G .

Notation. All vectors and matrices are real. The identity matrix is I with columns e_j , $\mathbf{1} = (1 \ \dots \ 1)^T$ is the column vector of all ones. For a matrix P , the inequality $P \geq 0$ means that each element of P is nonnegative. Similarly, $P > 0$ means that each element of P is positive.

A stochastic matrix P is a square matrix with $P \geq 0$ and $P\mathbf{1} = \mathbf{1}$. A probability vector v is a column vector with $v \geq 0$ and $v^T\mathbf{1} = 1$. The one-norm of a column vector v is $\|v^T\| \equiv |v|^T\mathbf{1}$. The eigenvalues $\lambda_i(P)$ of P are labeled in order of decreasing magnitude, $|\lambda_1(P)| \geq |\lambda_2(P)| \geq \dots$. The directed graph associated with P is denoted by $\Delta(P)$.

2. The problem. Let P be a stochastic matrix. We want to compute the stationary distribution π of P , that is, a column vector π such that [1, Theorem 2.(1.1)] $\pi^T P = \pi^T$ or $\pi^T(I - P) = 0$ where $\pi \geq 0$ and $\pi^T\mathbf{1} = 1$. If P is irreducible, then the eigenvalue 1 is simple and it is the spectral radius, and the stationary distribution $\pi > 0$ is unique [1, Theorem 2.(1.3)], [30, Theorem 2.1]. If P is primitive, then the eigenvalue 1 is distinct and all other eigenvalues have smaller magnitude [1, Theorem 2.(1.7)].

The matrix P may be large, so the idea is to work with only a small part of the matrix. That is, partition

$$P = \begin{pmatrix} P_{11} & P_{12} \\ P_{21} & P_{22} \end{pmatrix},$$

where P_{11} and P_{22} are square. Ideally the dimension of P_{11} is small compared to the dimension of P . The methods considered here approximate the stationary distribution π of P by operating on matrices derived from P_{11} whose dimension is, essentially, the same as that of P_{11} .

3. Exact aggregation/disaggregation. We present an aggregation/disaggregation method that is the basis for the SIAD method [12, 13, 15], an updating algorithm designed to compute stationary distributions of large Markov matrices, in the context of Google's PageRank. Our presentation differs from traditional ones [12, 19, 29] because we focus on a 2×2 partitioning and we start from a block LDU decomposition. The aggregation algorithm computes the components of π as stationary distributions of smaller matrices.

Partition the irreducible stochastic matrix

$$P = \begin{pmatrix} P_{11} & P_{12} \\ P_{21} & P_{22} \end{pmatrix}, \quad \pi = \begin{pmatrix} \pi_1 \\ \pi_2 \end{pmatrix}$$

so that P_{11} and P_{22} are square. Since $I - P$ is an M-matrix [1, Theorem 8.(4.2)], as well as singular and irreducible, the nontrivial leading principal submatrix $I - P_{11}$ is nonsingular [1, Theorem 6.(4.16)]. Hence we can factor $I - P = LDU$, where [19, proof of Theorem 2.3]

$$(3.1) \quad \begin{aligned} L &\equiv \begin{pmatrix} I & 0 \\ -P_{21}(I - P_{11})^{-1} & I \end{pmatrix}, & D &\equiv \begin{pmatrix} I - P_{11} & 0 \\ 0 & I - S \end{pmatrix}, \\ U &\equiv \begin{pmatrix} I & -(I - P_{11})^{-1}P_{12} \\ 0 & I \end{pmatrix}, & S &\equiv P_{22} + P_{21}(I - P_{11})^{-1}P_{12}, \end{aligned}$$

so that $I - S$ is the Schur complement of $I - P_{11}$ in $I - P$ [5, section 1]. The matrix S is also known as the stochastic complement of P_{22} in P [19, Definition 2.1]. It is

a special case of a Perron complement in the context of nonnegative matrices [20, Definition 2.1].

Since U is nonsingular we have $\pi^T(I - P) = 0$ if and only if $\pi^T L D = 0$. Hence

$$(3.2) \quad \pi_2^T S = \pi_2^T, \quad \pi_1^T = \pi_2^T P_{21}(I - P_{11})^{-1},$$

which means that π_2 is a stationary distribution for the smaller matrix S . The expressions (3.2) represent a partial version of the coupling theorem for 2×2 block matrices [19, Corollary 4.1].

Since P is irreducible and stochastic, so is S [19, Theorem 2.3]. Hence S has a unique, positive stationary distribution σ ,

$$\sigma^T S = \sigma^T, \quad \sigma^T \mathbf{1} = 1, \quad \sigma > 0.$$

Therefore we can determine π_2 from the stationary distribution σ of S and then set $\pi_2 = \rho\sigma$ where the (as yet unknown) factor ρ is responsible for the normalization $\pi^T \mathbf{1} = 1$.

The component π_1 and the factor ρ can also be expressed as components of the stationary distribution of a smaller matrix, the “aggregated” matrix

$$A \equiv \begin{pmatrix} P_{11} & P_{12}\mathbf{1} \\ \sigma^T P_{21} & \sigma^T P_{22}\mathbf{1} \end{pmatrix}.$$

This is because (3.2), $\pi_2 = \rho\sigma$, and $\sigma^T \mathbf{1} = 1$ imply

$$\begin{pmatrix} \pi_1^T & \rho \end{pmatrix} (I - A) = 0, \quad \begin{pmatrix} \pi_1^T & \rho \end{pmatrix} \mathbf{1} = 1.$$

Thus $\begin{pmatrix} \pi_1^T & \rho \end{pmatrix}$ is a stationary distribution of A . Since A is stochastic and irreducible [19, Theorem 4.1], it has a unique, positive stationary distribution α ,

$$\alpha^T A = \alpha^T, \quad \alpha^T \mathbf{1} = 1, \quad \alpha > 0.$$

Uniqueness implies $\alpha^T = \begin{pmatrix} \pi_1^T & \rho \end{pmatrix}$.

We have shown that a block LDU decomposition of $\pi^T(I - P) = 0$ leads to the expression of π_1 and π_2 as stationary distributions of two smaller matrices.

ALGORITHM 1 (exact aggregation/disaggregation). *Determine the stationary distribution π of a stochastic irreducible matrix P .*

1. *Determine the stationary distribution σ of $S \equiv P_{22} + P_{21}(I - P_{11})^{-1}P_{12}$.*
2. *Determine the stationary distribution α of $A \equiv \begin{pmatrix} P_{11} & P_{12}\mathbf{1} \\ \sigma^T P_{21} & \sigma^T P_{22}\mathbf{1} \end{pmatrix}$.*
3. *Partition $\alpha = \begin{pmatrix} \pi_1 \\ \rho \end{pmatrix}$, and set $\pi \equiv \begin{pmatrix} \pi_1 \\ \rho\sigma \end{pmatrix}$.*

The first two steps of Algorithm 1 can be interpreted as an aggregation because they take place on “aggregated” matrices of smaller size, while the third step represents a disaggregation that produces a long vector π of original size.

Since exact aggregation/disaggregation is too time consuming for large matrices, it makes sense to consider approximate aggregation/disaggregation methods. To understand what happens in the approximate aggregation/disaggregation method in section 4 we express π in terms of the block LDU decomposition (3.1).

PROPOSITION 3.1. *Let P be stochastic and irreducible. Then*

$$\pi^T = \rho \begin{pmatrix} * & \sigma^T \end{pmatrix} \begin{pmatrix} 0 & 0 \\ 0 & I \end{pmatrix} L^{-1},$$

where the scalar ρ ensures $\pi^T \mathbf{1} = 1$, and $*$ represents an arbitrary vector of appropriate length.

Proof. This follows from (3.2) and Algorithm 1. $\square$

4. Approximate aggregation/disaggregation. The approximate aggregation/disaggregation algorithm does away with the time-consuming computations involving the stochastic complement S . Instead of the exact stationary distribution σ of the stochastic complement S , we pick any positive probability vector $\tilde{\sigma}$, i.e., $\tilde{\sigma} > 0$ and $\tilde{\sigma}^T \mathbf{1} = 1$. The approximate aggregation matrix is

$$\tilde{A} \equiv \begin{pmatrix} P_{11} & P_{12}\mathbf{1} \\ \tilde{\sigma}^T P_{21} & \tilde{\sigma}^T P_{22}\mathbf{1} \end{pmatrix},$$

and it differs from the exact matrix A only in the last row.

The matrix $\tilde{A}$ is stochastic because P is stochastic. If P is irreducible, then so is $\tilde{A}$ for the following reason. If P_{11} is of order k , then the directed graph $\Delta(\tilde{A})$ of $\tilde{A}$ contains $k+1$ vertices. The fact that $\tilde{\sigma}$ and $\mathbf{1}$ are positive vectors implies the following: $\Delta(\tilde{A})$ contains an edge from vertex $k+1$ to i , $1 \leq i \leq k$, if and only if $\Delta(P)$ contains an edge from vertex j , $j \geq k+1$, to i . Also, $\Delta(\tilde{A})$ contains an edge from vertex i , $1 \leq i \leq k$, to $k+1$ if and only if $\Delta(P)$ contains an edge from vertex i to j , $j \geq k+1$. Finally, $\Delta(\tilde{A})$ contains an edge from i to j , $1 \leq i, j \leq k$, if and only if $\Delta(P)$ contains an edge from vertex i to j . Since $\Delta(P)$ is strongly connected, $\Delta(\tilde{A})$ must be strongly connected as well. Hence $\tilde{A}$ is irreducible. Thus it has a unique, positive stationary distribution $\tilde{\alpha}$,

$$\tilde{\alpha}^T \tilde{A} = \tilde{\alpha}^T, \quad \tilde{\alpha}^T \mathbf{1} = 1, \quad \tilde{\alpha} > 0.$$

All by itself this approximate aggregation makes no progress. However following it up by a power method iteration is known to have “a very salutary effect” [29, section 6.3.1]. Below is a single iteration of the iterative aggregation/disaggregation algorithm. It makes up for the approximation $\tilde{\sigma}$ by appending one iteration of the power method with P to produce an approximation $\tilde{\pi}$ to π .

ALGORITHM 2 (approximate aggregation/disaggregation). *Determine an approximation $\tilde{\pi}$ to the stationary distribution π of a stochastic irreducible matrix P , in one iteration.*

1. Select a vector $\tilde{\sigma}$ with $\tilde{\sigma} > 0$ and $\tilde{\sigma}^T \mathbf{1} = 1$.
2. Determine the stationary distribution $\tilde{\alpha}$ of $\tilde{A} \equiv \begin{pmatrix} P_{11} & P_{12}\mathbf{1} \\ \tilde{\sigma}^T P_{21} & \tilde{\sigma}^T P_{22}\mathbf{1} \end{pmatrix}$.
3. Partition $\tilde{\alpha} = \begin{pmatrix} \omega_1 \\ \tilde{\rho} \end{pmatrix}$, and set $\omega \equiv \begin{pmatrix} \omega_1 \\ \tilde{\rho} \tilde{\sigma} \end{pmatrix}$.
4. Multiply $\tilde{\pi}^T \equiv \omega^T P$.

If it so happens that $\tilde{\sigma} = \sigma$, then according to Proposition 3.1, $\omega^T = \tilde{\pi}^T = \omega^T P$, and ω is the desired stationary distribution.

The approximate aggregation/disaggregation Algorithm 2 amounts, in exact arithmetic, to one iteration of the power method, preconditioned by a partial LU factorization.

REMARK 4.1. Let P be stochastic and irreducible and $\tilde{\sigma}$ a vector with $\tilde{\sigma} > 0$ and $\tilde{\sigma}^T \mathbf{1} = 1$. Then

$$\tilde{\pi}^T = \tilde{\rho} \begin{pmatrix} * & \tilde{\sigma}^T \end{pmatrix} \begin{pmatrix} 0 & 0 \\ 0 & I \end{pmatrix} L^{-1} P,$$

where $\tilde{\rho}$ ensures $\tilde{\pi}^T \mathbf{1} = 1$.

Algorithm 2 can be viewed as applying one step of the power method, with the preconditioned matrix

$$\tilde{P} \equiv \begin{pmatrix} 0 & 0 \\ 0 & I \end{pmatrix} L^{-1} P,$$

to the vector $\begin{pmatrix} * & \tilde{\sigma}^T \end{pmatrix}$ and then normalizing it.

In the next section we execute Algorithm 2 repeatedly.

5. Iterative aggregation/disaggregation (SIAD). The iterative aggregation/disaggregation (SIAD) method [12, 13, 15] is an updating algorithm designed to compute PageRank. The method consists of repeated application of Algorithm 2. We show that the SIAD method can be viewed as a preconditioned power method and give a simple derivation of the asymptotic convergence rate.

ALGORITHM 3 (iterative aggregation/disaggregation (SIAD)). *Determine an approximation $\pi^{(k)}$ to the stationary distribution π of a stochastic irreducible matrix P , in k iterations.*

1. Select a vector $\pi^{(0)} = (\pi_1^{(0)} \quad \pi_2^{(0)})$ with $\pi_2^{(0)} > 0$.
2. Do $k = 1, 2, \dots$
 - (a) Normalize $\sigma^{(k)} \equiv \pi_2^{(k-1)} / [\pi_2^{(k-1)}]^T \mathbf{1}$.
 - (b) Determine the stationary distribution $\alpha^{(k)}$ of $A^{(k)} \equiv \begin{pmatrix} P_{11} & P_{12}\mathbf{1} \\ [\sigma^{(k)}]^T P_{21} & [\sigma^{(k)}]^T P_{22}\mathbf{1} \end{pmatrix}$.
 - (c) Partition $\alpha^{(k)} = (\omega_1^{(k)} / \rho_k)$, and set $\omega^{(k)} \equiv (\omega_1^{(k)} / \rho_k \sigma^{(k)})$.
 - (d) Multiply $[\pi^{(k)}]^T \equiv [\omega^{(k)}]^T P$.

Algorithm 3 is mathematically equivalent, in exact arithmetic, to the power method preconditioned by a partial LU factorization.

PROPOSITION 5.1. *Let P be stochastic and irreducible and $\pi^{(0)} = (\pi_1^{(0)} \quad \pi_2^{(0)})$ a vector with $\pi_2^{(0)} > 0$. Then Algorithm 3 produces iterates*

$$[\pi^{(k)}]^T = \tilde{\rho} [\pi^{(0)}]^T \tilde{P}^k, \quad \text{where} \quad \tilde{P} \equiv \begin{pmatrix} 0 & 0 \\ 0 & I \end{pmatrix} L^{-1} P,$$

and $\tilde{\rho}$ ensures $[\pi^{(k)}]^T \mathbf{1} = 1$.

Proof. This follows from Remark 4.1. $\square$

To determine the asymptotic convergence rate of Algorithm 3, label the eigenvalues $\lambda_i(P)$ of a matrix P in order of decreasing magnitude, $|\lambda_1(P)| \geq |\lambda_2(P)| \geq \dots$.

THEOREM 5.2 (asymptotic convergence rate). *If P is stochastic and irreducible, then the asymptotic convergence rate of Algorithm 3 is $|\lambda_2(S)|$.*

Proof. According to Proposition 5.1, Algorithm 3 represents a power method with the matrix

$$\tilde{P} = \begin{pmatrix} 0 & 0 \\ P_{21}(I - P_{11})^{-1} & S \end{pmatrix}.$$

Because $\tilde{P}$ is block triangular, its nonzero eigenvalues are those of S . Since P is stochastic and irreducible, so is S [19, Theorem 2.3]. Hence its spectral radius is $|\lambda_1(S)| = 1$. Therefore the power method applied to $\tilde{P}$ has asymptotic convergence rate $|\lambda_2(S)|/|\lambda_1(S)| = |\lambda_2(S)|$. $\square$

As a consequence, Algorithm 3 converges like the power method applied to the stochastic complement S . The example below illustrates the effect of the asymptotic convergence rate.

EXAMPLE 5.1. *Consider the stochastic irreducible matrix $P = aI_n + (1-a)\mathbf{1}\pi^T$ of order n , where $0 < a < 1$ is a scalar and π is a vector with $\pi^T \mathbf{1} = 1$ and $\pi > 0$.*

Then π is the stationary distribution of P . Partitioning gives

$$P = \begin{pmatrix} P_{11} & P_{12} \\ P_{21} & P_{22} \end{pmatrix} = \begin{pmatrix} aI_m + (1-a)\mathbf{1}\pi_1^T & (1-a)\mathbf{1}\pi_2^T \\ (1-a)\mathbf{1}\pi_1^T & aI_{n-m} + (1-a)\mathbf{1}\pi_2^T \end{pmatrix}.$$

The stochastic complement is

$$S = P_{22} + P_{21}(I - P_{11})^{-1}P_{12} = aI_{n-m} + (1-a)\mathbf{1}\sigma^T,$$

where $\sigma^T = \pi_2^T/\pi_2^T\mathbf{1}$ is the stationary distribution of S . The exact aggregation matrix is

$$\begin{aligned} A &= \begin{pmatrix} P_{11} & P_{12}\mathbf{1} \\ \sigma^T P_{21} & \sigma^T P_{22}\mathbf{1} \end{pmatrix} = \begin{pmatrix} aI_m + (1-a)\mathbf{1}\pi_1^T & (1-a)(\pi_2^T\mathbf{1})\mathbf{1} \\ (1-a)\pi_1^T & a + (1-a)(\pi_2^T\mathbf{1})\mathbf{1} \end{pmatrix} \\ &= aI_{m+1} + (1-a)\mathbf{1}\alpha^T, \end{aligned}$$

where $\alpha^T = (\pi_1^T \ \pi_2^T\mathbf{1})$ is the stationary distribution of A . The aggregation matrix A happens to have the same structure as the original matrix P .

The subdominant eigenvalue of S is $\lambda_2(S) = a$. Hence the asymptotic convergence rate of Algorithm 3 is a . What does this mean for the iterates $\pi^{(k)}$? Any positive probability vector $\tilde{\sigma}$ produces $\tilde{A} = A$; thus $\alpha^{(k)} = \alpha$ for all k . This means after one iteration the leading components of $\pi^{(k)}$ have converged, i.e., $\pi_1^{(k)} = \pi_1$ for all k . As for the trailing components of $\pi^{(k)}$, $[\pi_2^{(k)}]^T$ is a multiple of $[\pi_2^{(0)}]^T S^k$, where $S^k = a^k I_{n-m} + (1-a^k)\mathbf{1}\sigma^T$. Hence

$$[\sigma^{(k)}]^T = a^{k-1} \frac{[\pi_2^{(0)}]^T}{\gamma} + (1-a^{k-1}) \frac{\pi_2^T}{\tilde{\rho}}, \quad \tilde{\rho} \equiv \pi_2^T\mathbf{1}, \quad \gamma \equiv [\pi_2^{(0)}]^T\mathbf{1}.$$

The iterates are then

$$[\pi^{(k)}]^T = \begin{pmatrix} \pi_1^T & a^k \frac{\tilde{\rho}}{\gamma} [\pi_2^{(0)}]^T + (1-a^k)\pi_2^T \end{pmatrix}.$$

The leading component π_1 is exact, and the trailing component $a^k \frac{\tilde{\rho}}{\gamma} [\pi_2^{(0)}]^T + (1-a^k)\pi_2^T$ shows clearly that the contribution of the initial vector $\pi_2^{(0)}$ decreases as a power of $a = \lambda_2(S)$, while the contribution of the target vector π_2 increases at the same rate.

6. Stochastic complements. In this section we consider how different choices of stochastic complements affect the asymptotic convergence rate of Algorithm 3.

Remember that the asymptotic convergence rate of the power method applied to a primitive stochastic matrix P is $|\lambda_2(P)|$. In Theorem 5.2 we showed that the asymptotic convergence rate of Algorithm 3 applied to

$$P = \begin{pmatrix} P_{11} & P_{12} \\ P_{21} & P_{22} \end{pmatrix}$$

is $|\lambda_2(S)|$, where S is the stochastic complement of P_{22} . Unfortunately we cannot take for granted that Algorithm 3 has a better asymptotic convergence rate than the power method, i.e., that $|\lambda_2(S)| < |\lambda_2(P)|$. In fact, the opposite is possible, as we illustrate in one of the examples below.

One remedy for accelerating Algorithm 3 in case of slow convergence is to be more flexible about the choice of stochastic complements and not to limit oneself to stochastic complements from *leading* principal submatrices. One can form a stochastic complement from any principal submatrix [19, Definition 2.1]. One way to think about this (to keep the notation simple) is to apply a permutation similarity transformation to P [19, Lemma 2.1], i.e.,

$$QPQ^T = \begin{pmatrix} P_{11} & P_{12} \\ P_{21} & P_{22} \end{pmatrix},$$

where Q is a permutation matrix, and use the stochastic complement of the permuted matrix

$$S \equiv P_{22} + P_{21}(I - P_{11})^{-1}P_{12}.$$

Note that a permutation similarity merely reorders the components of the stationary distribution and preserves the eigenvalues. That is, we can choose different stochastic complements by thinking of applying Algorithm 3 to a matrix QPQ^T that is permutationally similar to P . The question is, can we find permutations Q such that the stochastic complement S of QPQ^T has $|\lambda_2(S)| \leq |\lambda_2(P)|$? That is, can Algorithm 3 converge faster than the power method on a suitably reordered matrix?

The examples below illustrate different convergence possibilities for the power method and Algorithm 3.

EXAMPLE 6.1. *This example illustrates that Algorithm 3 can converge in one iteration, while the power method can fail to converge to the stationary distribution.*

Let v be a positive probability vector, and

$$P = \begin{pmatrix} 0 & v^T \\ \mathbf{1} & 0 \end{pmatrix}$$

a stochastic matrix. The eigenvalues of P are 1, -1 , and 0. Hence $|\lambda_2(P)| = 1$, which means that the power method applied to P fails to converge to the stationary distribution (unless the initial vector is the stationary distribution). The stochastic complement that arises from partitioning off the first row and column of P is $S = \mathbf{1}v^T$. Its eigenvalues are 1 and 0. Thus $|\lambda_2(S)| = 0$, so that Algorithm 3 converges in a single iteration.

EXAMPLE 6.2. *This example illustrates that the power method and Algorithm 3 can have the same asymptotic convergence rate, regardless of the choice of stochastic complement.*

Consider the matrix from Example 5.1, $P = aI + (1-a)\mathbf{1}\pi^T$, where π is a positive probability vector and $0 < a < 1$. The matrix P has just two distinct eigenvalues, $\lambda_1(P) = 1$ and $\lambda_i(P) = a$, $i > 1$. Each stochastic complement of P has the form $S = aI + (1-a)\mathbf{1}\sigma^T$, where σ is a positive probability vector. The eigenvalues of S are also 1 and a . In particular, $\lambda_2(S) = \lambda_2(P)$, so that the power method and Algorithm 3 converge at the same rate.

EXAMPLE 6.3. *This example illustrates that the asymptotic convergence rate of Algorithm 3 can be significantly worse than that of the power method, regardless of the choice of stochastic complement.*

The irreducible stochastic matrix

$$P = \begin{pmatrix} 5/6 & 0 & 1/6 \\ 3/4 & 1/6 & 1/12 \\ 2/3 & 1/3 & 0 \end{pmatrix}$$

has eigenvalues $\lambda_1(P) = 1$ and $\lambda_2(P) = \lambda_3(P) = 0$. It turns out that 0 is a defective eigenvalue, corresponding to a Jordan block of order 2. Thus the power method converges in at most two iterations for any initial probability vector.

Now let us consider all possible stochastic complements that can arise during Algorithm 3. Denote by S_i the stochastic complement that arises from P by simultaneously permuting row and column i to the first position and then forming the corresponding 2×2 stochastic complement. The stochastic complements have the following subdominant eigenvalues:

$$\lambda_2(S_1) = -1/6, \quad \lambda_2(S_2) = -2/15, \quad \lambda_2(S_3) = 5/36.$$

In all cases $|\lambda_2(S_i)| > |\lambda_2(P)| = 0$, $1 \leq i \leq 3$. Thus for any choice of stochastic complement, the asymptotic convergence rate of Algorithm 3 is slower than that of the power method.

7. An upper bound on the asymptotic convergence rate. We discuss an upper bound on the subdominant eigenvalue $|\lambda_2|$, which also furnishes upper bounds for the asymptotic convergence rates of the power method and Algorithm 3. The upper bound can be easier to compute than $|\lambda_2|$ and has some properties convenient for a convergence analysis.

Example 6.3 shows that for any stochastic complement S of an irreducible stochastic matrix P , $|\lambda_2(S)| > |\lambda_2(P)|$ is possible, which means that for such a matrix the asymptotic convergence rate of Algorithm 3 is always slower than that of the power method. As far as Algorithm 3 is concerned, this situation is undesirable. However, for the standard upper bound τ on $|\lambda(S)|$ described below, it will transpire that $\tau(S) > \tau(P)$ is impossible. Denote by $\|v^T\| \equiv |v|^T \mathbf{1}$ the one-norm of a column vector v , and by e_j the j th column of the identity matrix. With

$$\tau(P) \equiv \frac{1}{2} \max_{i,j} \|(e_i - e_j)^T P\|,$$

we have, for any stochastic matrix P [27, section 4],

$$(7.1) \quad |\lambda_2(P)| \leq \tau(P).$$

Observe that $\tau(P) \leq 1$ for any stochastic matrix P . The quantity $\tau(P)$ represents an upper bound on the asymptotic convergence rate of the power method, while $\tau(S)$ is an upper bound on the asymptotic convergence rate of Algorithm 3. In contrast to the eigenvalues, where $|\lambda_2(S)| > |\lambda_2(P)|$ is possible, this cannot happen for the upper bounds τ . The following result shows that $\tau(S) \leq \tau(P)$, when P is irreducible. This was already shown in [23, Theorem 3.1], but our proof is more direct. We also note that there is a one-line proof of this fact, which is based on rather more detailed knowledge of the properties of τ described in [27].

PROPOSITION 7.1. *If*

$$P = \begin{matrix} & \begin{matrix} k & n-k \end{matrix} \\ \begin{matrix} k \\ n-k \end{matrix} & \begin{pmatrix} P_{11} & P_{12} \\ P_{21} & P_{22} \end{pmatrix} \end{matrix}, \quad 1 \leq k \leq n-2,$$

is stochastic and irreducible, and $S \equiv P_{22} + P_{21}(I - P_{11})^{-1}P_{12}$, then

$$\tau(S) \leq \tau(P).$$

Proof. We claim that $\|u^T(I - P_{11})^{-1}P_{12}\| \leq \|u\|$ for all vectors u . To see that the claim is true, observe that $P\mathbf{1} = \mathbf{1}$ implies $(I - P_{11})^{-1}P_{12}\mathbf{1} = \mathbf{1}$. With $v_j \equiv |u_j|$, $1 \leq j \leq k$, the triangle inequality implies

$$|u^T(I - P_{11})^{-1}P_{12}e_i| \leq v^T(I - P_{11})^{-1}P_{12}e_i, \quad 1 \leq i \leq n-k.$$

Hence

$$\|u^T(I - P_{11})^{-1}P_{12}\| \leq \sum_{i=1}^{n-k} v^T(I - P_{11})^{-1}P_{12}e_i = v^T(I - P_{11})^{-1}P_{12}\mathbf{1} = v^T\mathbf{1} = \|u\|,$$

which proves the claim.

For any two indices $1 \leq i, j \leq n - k$, the triangle inequality and the above claim imply

$$\begin{aligned} \|(e_i - e_j)^T S\| &\leq \|(e_i - e_j)^T P_{22}\| + \|(e_i - e_j)^T P_{21}(I - P_{11})^{-1}P_{12}\| \\ &\leq \|(e_i - e_j)^T P_{22}\| + \|(e_i - e_j)^T P_{21}\| = \|(e_{k+i} - e_{k+j})^T P\|. \end{aligned}$$

Hence

$$\tau(S) \leq \frac{1}{2} \max_{k+1 \leq i, j \leq n} \|(e_i - e_j)^T P\| \leq \tau(P). \quad \square$$

Proposition 7.1 shows that the upper bound on the asymptotic convergence rate of Algorithm 3 is never worse than that of the power method. Note that Proposition 7.1 does not hold for $k = n - 1$ because S is 1×1 .

The result below implies that there are matrices (including the Google matrix in section 9) for which the power method always converges, and for which the asymptotic convergence rate of Algorithm 3 is at least as good as that of the power method.

COROLLARY 7.2. *Let $G \equiv cP + (1 - c)\mathbf{1}v^T$, where P is a stochastic matrix, $0 < c < 1$, and v is a positive probability vector. If $|\lambda_2(P)| = 1$, then for any stochastic complement S of G*

$$|\lambda_2(S)| \leq |\lambda_2(G)| = c < 1.$$

Proof. Since v is positive and $c < 1$, G is irreducible. Inequality (7.1) and Proposition 7.1 imply $|\lambda_2(S)| \leq \tau(S) \leq \tau(G)$. The results [6, Theorem 2], [14, Theorem 4.1], [28, Theorem 2.3], and (7.1) imply $c = |\lambda_2(G)| \leq \tau(G)$. At last, $\tau(G) \leq c$ follows from $\tau(G) = c\tau(P) \leq c$. $\square$

Note that the fact that $|\lambda_2(G)| = c$ in Corollary 7.2 was already shown in [6, Theorem 2] and [14, Theorem 4.1]. The crucial inequality in Corollary 7.2 is $|\lambda_2(S)| \leq |\lambda_2(G)|$. The matrices in Examples 5.1 and 6.2 are special cases of the matrices in Corollary 7.2 with $P = I$.

For certain matrices, the upper bound τ provides an easily computable upper bound on the asymptotic convergence rate of Algorithm 3. These include matrices where P_{21} or P_{22} has a column with all elements positive.

REMARK 7.1. *If P is irreducible and stochastic and $p_{im} > 0$ for some m and $k + 1 \leq i \leq n$, then*

$$|\lambda_2(S)| \leq 1 - \min_{k+1 \leq i \leq n} p_{im} < 1.$$

To see why this is true, note that rows i and j , $k + 1 \leq i, j \leq n$, have positive entries in position m and

$$\begin{aligned} \|(e_i - e_j)^T P\| &= \sum_{l=1}^n |p_{il} - p_{jl}| \leq 1 - p_{im} + 1 - p_{jm} + |p_{im} - p_{jm}| \\ &= 2(1 - \min\{p_{im}, p_{jm}\}). \end{aligned}$$

Proposition 7.1 implies $|\lambda_2(S)| \leq 1 - \min_{k+1 \leq i \leq n} p_{im} < 1$.

8. Conditions for convergence. We derive conditions that guarantee convergence, in exact arithmetic, of Algorithm 3.

The power method with a stochastic irreducible matrix P converges if $|\lambda_2(P)| < 1$, or equivalently if P is primitive [1, Definition 2.(1.8)]. However, just because P is primitive, a stochastic complement S is not necessarily primitive [19, section 5].

REMARK 8.1. *Let*

$$P = \begin{pmatrix} P_{11} & P_{12} \\ P_{21} & P_{22} \end{pmatrix}$$

be stochastic and irreducible. If P_{22} is primitive or P_{22} has at least one nonzero diagonal element, then Algorithm 3 converges.

This follows from [19, Theorems 5.1 and 5.2].

REMARK 8.2. *Primitivity of a stochastic complement S can also be related to properties of the directed graph $\Delta(S)$, which is also known as the compressed directed graph [8, sections 1 and 4]. A useful property of the compressed directed graph is the following: For any pair of indices i, j with $1 \leq i, j \leq n - k$, if there is a path from vertex $k + i$ to vertex $k + j$ in $\Delta(P)$, then there is a path from vertex i to vertex j in $\Delta(S)$.*

THEOREM 8.1. *If P is irreducible and stochastic, then the following holds: P has at least one primitive stochastic complement of order at least two $\iff P$ is not a cyclic permutation matrix.*

Proof. First, we claim that P is a cyclic permutation matrix if and only if each column of P has at most one positive entry. One direction of the claim is clear. To see the other direction, suppose that each column of P has at most one positive entry. Because P is irreducible, each column must have exactly one positive entry. Further, since the entries of P are at most 1, and the sum of all entries in P is equal to the order of the matrix, the single positive entry in each column of P must be a 1. Thus, P is an irreducible stochastic $(0, 1)$ matrix; hence P is a cyclic permutation matrix.

With the claim established, we now proceed with the proof of the result.

$\Leftarrow$: Suppose that P is not a cyclic permutation matrix. Then some column m of P has at least two positive entries, i.e., $p_{i_1, m}, \dots, p_{i_q, m} > 0$ for $i_1, \dots, i_q$ with $q \geq 2$. With a suitable permutation Q , partition

$$QPQ^T = \begin{pmatrix} P_{11} & P_{12} \\ P_{21} & P_{22} \end{pmatrix}$$

so that P_{22} is $q \times q$, and P_{21} or P_{22} has a positive column. Remark 7.1 implies that $|\lambda_2(S)| < 1$, so that S is primitive.

$\Rightarrow$: Suppose that

$$P = \begin{pmatrix} P_{11} & P_{12} \\ P_{21} & P_{22} \end{pmatrix}$$

is a cyclic permutation matrix. Since $\Delta(P)$ has exactly one cycle involving every vertex, and since $\Delta(P_{11})$ is a proper subgraph of $\Delta(P)$, $\Delta(P_{11})$ contains no cycles. Hence P_{11} is nilpotent, so $(I - P_{11})^{-1}$ is a finite sum of powers of P_{11} . In particular, $(I - P_{11})^{-1}$ is a matrix with nonnegative integer entries. Hence the stochastic complement $S = P_{22} + P_{21}(I - P_{11})^{-1}P_{12}$ is also a matrix of nonnegative integers. Since S is stochastic, it is a $(0, 1)$ matrix, and since S is irreducible, it is a cyclic permutation matrix. Therefore S is not primitive. $\square$

COROLLARY 8.2. *If the irreducible stochastic matrix P is not a cyclic permutation matrix, then P has at least one stochastic complement S for which Algorithm 3 converges, i.e., $|\lambda_2(S)| < 1$.*

9. The Google matrix. We show that, when Algorithm 3 is applied to the Google matrix G with an appropriate choice of stochastic complement, then the asymptotic convergence rate is better than that of the power method. We also show how one can save computations when applying Algorithm 3 to G .

In Corollary 7.2 we showed that if $|\lambda_2(P)| = 1$, then $|\lambda_2(S)| \leq |\lambda_2(G)| = c < 1$ for any stochastic complement S . This means that the power method applied to G converges, and the asymptotic convergence rate of Algorithm 3 is at least as good as that of the power method. Now we prove a stronger result by exploiting the structure of the Google matrix and by choosing a particular stochastic complement. In this case the asymptotic convergence rate of Algorithm 3 is strictly better than that of the power method. We make the following assumptions; these assumptions are not restrictive because they are almost certainly satisfied by the Google matrix.

Assumptions 1.

1. The stochastic matrix P contains $k \geq 2$ essential index classes $C_1, \dots, C_k$. The class of inessential indices (if any) is called D .

Recall that an index i is inessential if for some index j there is a chain of links from page i to page j , but there is no chain of links back from page j to page i (i.e., $(P^m)_{ij} > 0$ for some power P^m , but $(P^k)_{ji} = 0$ for all powers P^k); otherwise the index i is essential. An essential index class C is a set of essential indices, such that there is a chain of links between any two indices in C , but no outgoing link to an index outside of C [26, section 1.2].

2. Each essential index class C_j contains an index i_j whose corresponding diagonal entry in P is 0; i.e., at least one web page in each C_j does not link back to itself.

These assumptions imply that the rows of P that are equal to $\frac{1}{n}\mathbf{1}^T$ all correspond to inessential indices in D . Also, $|\lambda_2(P)| = 1$ because P has at least 2 essential index classes. Note that class D contains not only dangling nodes, i.e., nodes without out-links, but also nodes that link to dangling nodes.

When P satisfies Assumptions 1 there is a permutation matrix Q that orders the rows and columns of QPQ^T into the following form: $i_1, \dots, i_k, C_1 \setminus \{i_1\}, \dots, C_k \setminus \{i_k\}, D$. Partition

$$QGQ^T = \begin{pmatrix} G_{11} & G_{12} \\ G_{21} & G_{22} \end{pmatrix},$$

where G_{11} is $k \times k$ and corresponds to the indices $i_1, \dots, i_k$, and let $S_G \equiv G_{22} + G_{21}(I - G_{11})^{-1}G_{12}$ be the corresponding stochastic complement.

THEOREM 9.1. *Let $G = cP + (1 - c)\mathbf{1}v^T$ where $0 < c < 1$, v is a positive probability vector, and the stochastic matrix P satisfies Assumptions 1. Then*

$$|\lambda_2(S_G)| < |\lambda_2(G)|.$$

Proof. We exploit the structure of P , in several stages, to capture the structure of the stochastic complement S_G . Partition

$$QPQ^T = \begin{pmatrix} 0 & P_{12} \\ P_{21} & P_{22} \end{pmatrix},$$

where the leading diagonal block of order k is zero because the indices $i_1, \dots, i_k$ belong to different essential classes, and each index corresponds to a zero diagonal element of P . While QPQ^T is presented as a 2×2 block matrix, whose leading principal block

corresponds to indices $i_1, \dots, i_k$, we will have occasion to further partition the second block into two classes, $C_1 \setminus \{i_1\}, \dots, C_k \setminus \{i_k\}$, and D .

Partitioning

$$v = \begin{pmatrix} v_1 \\ v_2 \\ v_3 \end{pmatrix}$$

conformally with P gives, for the partition of QQ^T ,

$$\begin{aligned} G_{11} &= (1-c)\mathbf{1}v_1^T, & G_{12} &= cP_{12} + (1-c)\mathbf{1} \begin{pmatrix} v_2^T & v_3^T \end{pmatrix}, \\ G_{21} &= cP_{21} + (1-c)\mathbf{1}v_1^T, & G_{22} &= cP_{22} + (1-c)\mathbf{1} \begin{pmatrix} v_2^T & v_3^T \end{pmatrix}. \end{aligned}$$

We claim that $|\lambda_2(S_G)| < c$, from which the conclusion follows with Corollary 7.2.

To establish this claim, write $S_G = G_{22} + G_{21}(I - G_{11})^{-1}G_{12}$ as a convex combination $S_G = cT + (1-c)\mathbf{1}z^T$, where

$$T \equiv P_{22} + P_{21}(I - G_{11})^{-1}G_{12}, \quad z^T \equiv \begin{pmatrix} v_2^T & v_3^T \end{pmatrix} + v_1^T(I - G_{11})^{-1}G_{12}.$$

Using the three facts that $(I - G_{11})^{-1} = I + \beta\mathbf{1}v_1^T$, where

$$\beta \equiv \frac{1-c}{1-(1-c)\delta}, \quad \delta \equiv v_1^T\mathbf{1};$$

the stochastic complement $P_{22} + P_{21}P_{12}$ of P is again a stochastic matrix; and $P_{12}\mathbf{1} = \mathbf{1}$, we conclude that T is stochastic and z is a positive probability vector. Since $\lambda_2(S_G) = c\lambda_2(T)$ [14, Theorem 4.1] it suffices to prove $|\lambda_2(T)| < 1$. To this end, we will show that T has at least one positive row and is irreducible. This implies T is primitive [1, Corollary 2.(4.8)], hence $|\lambda_2(T)| < 1$ [1, Definition 2.(1.8)].

The principal submatrix of P corresponding to the essential index class C_j can be written as

$$M_j = \begin{pmatrix} 0 & x_j^T \\ y_j & B_j \end{pmatrix},$$

where the leading row and column correspond to index i_j , and the remaining rows and columns correspond to the indices of $C_j \setminus \{i_j\}$. Since C_j is an essential index class, y_j is not the zero vector. Hence M_j is irreducible and stochastic, and so is its stochastic complement $B_j + y_jx_j^T$.

With

$$B \equiv \begin{pmatrix} B_1 & & \\ & \ddots & \\ & & B_k \end{pmatrix}, \quad X \equiv \begin{pmatrix} x_1^T & & \\ & \ddots & \\ & & x_k^T \end{pmatrix}, \quad Y \equiv \begin{pmatrix} y_1 & & \\ & \ddots & \\ & & y_k \end{pmatrix},$$

it follows that

$$QPQ^T = \begin{pmatrix} 0 & P_{12} \\ P_{21} & P_{22} \end{pmatrix} = \left(\begin{array}{c|cc} 0 & X & 0 \\ Y & B & 0 \\ R_1 & R_2 & R_3 \end{array} \right).$$

Hence

$$T = \underbrace{P_{22} + cP_{21}P_{12}} + c\beta P_{21}\mathbf{1}v_1^T P_{12} + \underbrace{(1-c)(1+\beta\delta)P_{21}\mathbf{1}} \begin{pmatrix} v_2^T & v_3^T \end{pmatrix}.$$

Consider the last summand in T . The row $(v_2^T \ v_3^T)$ is positive because v is. Moreover

$$P_{21}\mathbf{1} = \begin{pmatrix} y_1 \\ \vdots \\ y_k \\ R_1\mathbf{1} \end{pmatrix},$$

where each vector y_j has at least one positive element. This means, T has at least one positive row corresponding to each $C_j \setminus \{i_j\}$, $1 \leq j \leq k$.

We still need to prove that T is irreducible. T being irreducible is equivalent to the directed graph $\Delta(T)$ being strongly connected [1, Theorem 2.(2.7)]. We will show that for any vertex in $\Delta(T)$ there is a path to any other vertex.

1. Paths within $C_j \setminus \{i_j\}$, $1 \leq j \leq k$: Consider the first two summands in T ,

$$P_{22} + cP_{21}P_{12} = \begin{pmatrix} B + cYX & 0 \\ R_2 + cR_1X & R_3 \end{pmatrix}.$$

The leading principal submatrix is

$$B + cYX = \begin{pmatrix} B_1 + cy_1x_1^T & & \\ & \ddots & \\ & & B_k + cy_kx_k^T \end{pmatrix}.$$

Every diagonal element $B_j + cy_jx_j^T$ is irreducible because its directed graph is the same as that of the irreducible matrix $B_j + y_jx_j^T$, i.e., $\Delta(B_j + cy_jx_j^T) = \Delta(B_j + y_jx_j^T)$. Therefore $\Delta(T)$ contains a path from every vertex in $C_j \setminus \{i_j\}$ to all other vertices in $C_j \setminus \{i_j\}$.

2. Paths leaving $C_j \setminus \{i_j\}$, $1 \leq j \leq k$: The above fact that T contains at least one positive row for each $C_j \setminus \{i_j\}$ implies that $\Delta(T)$ contains a path from every vertex in $C_j \setminus \{i_j\}$ to any other vertex in $\Delta(T)$. Together with item 1, this implies $\Delta(T)$ contains a path from every vertex in $C_j \setminus \{i_j\}$ to any other vertex in $\Delta(T)$, $1 \leq j \leq k$.
3. Paths leaving D : From the leading two summands of T , $P_{22} + cP_{21}P_{12}$ with $c > 0$, we see that $\Delta(T)$ contains $\Delta(P_{22} + cP_{21}P_{12}) = \Delta(S_P)$, where

$$S_P \equiv P_{22} + P_{21}P_{12} = \begin{pmatrix} B & 0 \\ R_2 & R_3 \end{pmatrix} + \begin{pmatrix} Y \\ R_1 \end{pmatrix} \begin{pmatrix} X & 0 \end{pmatrix}$$

is a stochastic complement of P . Since every inessential index $d \in D$ has a path to some essential index class C_j , $\Delta(P)$ contains a path from d to every vertex in C_j . Remark 8.1 implies that therefore $\Delta(S_P)$ also contains a path from d to every vertex of $C_j \setminus \{i_j\}$. Since $\Delta(T)$ contains $\Delta(S_P)$, every vertex in D must have a path to some vertex of $C_j \setminus \{i_j\}$. With item 2, we conclude that $\Delta(T)$ contains a path from every vertex in D to any other vertex in D . $\square$

Theorem 9.1 shows that for the Google matrix, there is always a choice of stochastic complement so that the asymptotic convergence rate of Algorithm 3 is strictly smaller than that of the power method. We give a simple illustration below.

EXAMPLE 9.1. *This example illustrates that the asymptotic convergence rate of Algorithm 3 applied to the Google matrix G can be the square of that of the power method.*

Let $G = cP + (1 - c)\mathbf{1}v^T$, where $0 < c < 1$, v is a positive probability vector, and

$$P = \begin{pmatrix} 0 & I \\ I & 0 \end{pmatrix}.$$

If P is of order $n = 2k$, then its directed graph $\Delta(P)$ is a union of k cycles, each of length 2.

Partitioning $v^T = (v_1^T \ v_2^T)$ conformally with P gives, for the stochastic complement from Theorem 9.1,

$$S_G = (1 - c)\mathbf{1}v_2^T + (cI + (1 - c)\mathbf{1}v_1^T) (I - (1 - c)\mathbf{1}v_1^T)^{-1} (cI + (1 - c)\mathbf{1}v_2^T).$$

Multiplying out yields

$$S_G = c^2I + \frac{1 - c^2}{1 - (1 - c)\delta} \mathbf{1}(cv_1^T + v_2^T), \quad \text{where } \delta \equiv v_1^T \mathbf{1}.$$

The eigenvalues of S_G are 1 and c^2 .

Hence the asymptotic convergence rate of Algorithm 3 for this matrix G is c^2 , while the convergence rate of the power method is only c .

Below we illustrate that when Algorithm 3 is applied to the Google matrix G , the computation of the stationary vectors of the aggregated matrices $A^{(k)}$ is cheap.

REMARK 9.1. Algorithm 3 applied to the Google matrix G involves the computation of stationary vectors of aggregated matrices

$$A^{(k)} = \begin{pmatrix} G_{11} & G_{12}\mathbf{1} \\ [\sigma^{(k)}]^T G_{21} & [\sigma^{(k)}]^T G_{22}\mathbf{1} \end{pmatrix}.$$

For the stochastic complement S_G from Theorem 9.1, $G_{11} = (1 - c)\mathbf{1}v_1^T$. Hence

$$A^{(k)} = \begin{pmatrix} (1 - c)\mathbf{1}v_1^T & \gamma\mathbf{1} \\ [z^{(k)}]^T & 1 - [z^{(k)}]^T \mathbf{1} \end{pmatrix},$$

$$[z^{(k)}]^T \equiv [\sigma^{(k)}]^T G_{21}, \quad \gamma \equiv 1 - (1 - c)\delta, \quad \delta \equiv v_1^T \mathbf{1}.$$

The simple structure of $A^{(k)}$ yields an explicit expression for its stationary vector,

$$\alpha^{(k)} = \frac{1}{\epsilon_k + \gamma} (\epsilon_k(1 - c)v_1^T + \gamma[z^{(k)}]^T \ \gamma), \quad \text{where } \epsilon_k \equiv [z^{(k)}]^T \mathbf{1}.$$

Thus, when Algorithm 3 is applied to the Google matrix G , the computation of the stationary vectors $\alpha^{(k)}$ is cheap.

Acknowledgments. We thank Amy Langville and Carl Meyer for many helpful discussions.

REFERENCES

- [1] A. BERMAN AND R. J. PLEMMONS, *Nonnegative Matrices in the Mathematical Sciences*, Classics Appl. Math., SIAM, Philadelphia, 1994.
- [2] C. BREZINSKI AND M. REDIVO-ZAGLIA, *The PageRank vector: Properties, computation, approximation and acceleration*, SIMAX, submitted.
- [3] C. BREZINSKI, M. REDIVO-ZAGLIA, AND S. SERRA-CAPIZZANO, *Extrapolation Methods for PageRank Computations*, Comptes Rendus Acad. Sci., Sér. I, 340 (2005), pp. 393–397.
- [4] S. BRIN AND L. PAGE, *The anatomy of a large-scale hypertextual web search engine*, Comput. Networks and ISDN Systems, 30 (1998), pp. 107–117.

- [5] R. W. COTTLE, *Manifestations of the Schur complement*, Linear Algebra Appl., 8 (1974), pp. 189–211.
- [6] T. H. HAVELIWALA AND S. D. KAMVAR, *The Second Eigenvalue of the Google Matrix*, Technical report 2003-20, Stanford University, Stanford, CA, 2003, <http://dbpubs.stanford.edu/pub/2003-20>.
- [7] T. H. HAVELIWALA, S. D. KAMVAR, D. KLEIN, C. D. MANNING, AND G. H. GOLUB, *Computing PageRank Using Power Extrapolation*, Technical report 2003-45, Stanford University, Stanford, CA, 2003, <http://dbpubs.stanford.edu/pub/2003-45>.
- [8] C. R. JOHNSON AND C. XENOPHONTOS, *Irreducibility and primitivity of Perron complements: Application of the compressed directed graph*, in Graph Theory and Sparse Matrix Computation, IMA Vol. Math. Appl. 56, A. George, J. R. Gilbert, and J. W. H. Liu, eds., Springer, New York, 1993, pp. 101–106.
- [9] S. D. KAMVAR, T. H. HAVELIWALA, AND G. H. GOLUB, *Adaptive methods for the computation of PageRank*, Linear Algebra Appl., 386 (2004), pp. 51–65.
- [10] S. D. KAMVAR, T. H. HAVELIWALA, C. D. MANNING, AND G. H. GOLUB, *Exploiting the Block Structure of the Web for Computing PageRank*, Technical report SCCM03-02, Stanford University, Stanford, CA, 2003, <http://www-sccm.stanford.edu/nf-publications-tech.html>.
- [11] S. D. KAMVAR, T. H. HAVELIWALA, C. D. MANNING, AND G. H. GOLUB, *Extrapolation methods for accelerating PageRank computations*, in Proceedings of the Twelfth International World Wide Web Conference (WWW2003), Toronto, 2003, ACM Press, New York, pp. 261–270.
- [12] A. N. LANGVILLE AND C. D. MEYER, *Updating PageRank Using the Group Inverse and Stochastic Complementation*, Technical report CRSC-TR02-32, North Carolina State University, Raleigh, NC, 2002, <http://www.ncsu.edu/crsc/reports/reports02.html>.
- [13] A. N. LANGVILLE AND C. D. MEYER, *Updating the Stationary Vector of an Irreducible Markov Chain*, Technical report CRSC-TR02-33, North Carolina State University, Raleigh, NC, 2002, <http://www.ncsu.edu/crsc/reports/reports02.html>.
- [14] A. N. LANGVILLE AND C. D. MEYER, *Fiddling with PageRank*, Technical report CRSC-TR03-34, North Carolina State University, Raleigh, NC, 2003, <http://www.ncsu.edu/crsc/reports/reports03.html>.
- [15] A. N. LANGVILLE AND C. D. MEYER, *Updating the Stationary Vector of an Irreducible Markov Chain with an Eye on Google's PageRank*, Technical report, North Carolina State University, Raleigh, NC, 2004.
- [16] C. P.-C. LEE, G. H. GOLUB, AND S. A. ZENIOS, *A Fast Two-Stage Algorithm for Computing PageRank*, Technical report SCCM03-15, Stanford University, Stanford, CA, 2003, <http://www-sccm.stanford.edu/nf-publications-tech.html>.
- [17] I. MAREK AND P. MAYER, *Convergence theory of some classes of iterative aggregation/disaggregation methods for computing stationary probability vectors of stochastic matrices*, Linear Algebra Appl., 363 (2003), pp. 177–200.
- [18] I. MAREK AND D. B. SZYLD, *Local convergence of the (exact and inexact) iterative aggregation method for linear systems and Markov operators*, Numer. Math., 69 (1994), pp. 61–82.
- [19] C. D. MEYER, *Stochastic complementation, uncoupling Markov chains, and the theory of nearly reducible systems*, SIAM Rev., 31 (1989), pp. 240–272.
- [20] C. D. MEYER, *Uncoupling the Perron eigenvector problem*, Linear Algebra Appl., 114/115 (1989), pp. 69–94.
- [21] C. B. MOLER, *The world's largest matrix computation*, MATLAB News and Notes, October, 2002.
- [22] C. B. MOLER, *Numerical Computing with MATLAB*, SIAM, Philadelphia, 2004.
- [23] M. NEUMANN AND J. XU, *On the stability of the computation of the stationary probabilities of Markov chains using Perron complements*, Numer. Linear Algebra Appl., 10 (2003), pp. 603–618.
- [24] A. Y. NG, A. X. ZHENG, AND M. I. JORDAN, *Link analysis, eigenvectors and stability*, in Proceedings of the 17th International Joint Conference on Artificial Intelligence (IJCAI), B. Nabel, ed., Morgan Kaufmann/Elsevier, San Francisco, 2001, pp. 903–910.
- [25] L. PAGE, S. BRIN, R. MOTWANI, AND T. WINOGRAD, *The PageRank Citation Ranking: Bringing Order to the Web*, 1999, <http://dbpubs.stanford.edu/pub/1999-66>.
- [26] E. SENETA, *Non-negative Matrices and Markov Chains*, Springer-Verlag, New York, 1981.
- [27] E. SENETA, *Explicit forms for ergodicity coefficients and spectrum localization*, Linear Algebra Appl., 60 (1984), pp. 187–197.
- [28] S. SERRA-CAPIZZANO, *Jordan canonical form of the Google matrix: A potential contribution to the PageRank Computation*, SIAM J. Matrix Anal. Appl., 27 (2005), pp. 305–312.
- [29] W. J. STEWART, *Introduction to the Numerical Solution of Markov Chains*, Princeton University Press, Princeton, NJ, 1994.
- [30] R. S. VARGA, *Matrix Iterative Analysis*, Prentice Hall, Englewood Cliffs, NJ, 1962.

UPDATING MARKOV CHAINS WITH AN EYE ON GOOGLE’S PAGERANK*

AMY N. LANGVILLE[†] AND CARL D. MEYER[‡]

Abstract. An iterative algorithm based on aggregation/disaggregation principles is presented for updating the stationary distribution of a finite homogeneous irreducible Markov chain. The focus is on large-scale problems of the kind that are characterized by Google’s PageRank application, but the algorithm is shown to work well in general contexts. The algorithm is flexible in that it allows for changes to the transition probabilities as well as for the creation or deletion of states. In addition to establishing the rate of convergence, it is proven that the algorithm is globally convergent. Results of numerical experiments are presented.

Key words. Markov chains, updating, stationary vector, PageRank, stochastic complementation, aggregation/disaggregation, Google

AMS subject classifications. 60J10, 65C40, 15A51, 65F10, 65F15, 65F30, 65F50, 68P20, 68P10, 15A99, 15-04, 15A18, 15A06

DOI. 10.1137/040619028

1. Introduction. Suppose that the stationary distribution vector

$$\phi^T = (\phi_1, \phi_2, \dots, \phi_m)$$

for an m -state homogeneous irreducible Markov chain with transition probability matrix $\mathbf{Q}_{m \times m}$ is known (by prior computation, such as solving $\phi^T \mathbf{Q} = \phi^T$, or by other means), but the chain requires updating by altering some of its transition probabilities or by adding or deleting some states. Suppose that the updated transition probability matrix $\mathbf{P}_{n \times n}$ is also irreducible. The updating problem is to compute the updated stationary distribution $\pi^T = (\pi_1, \pi_2, \dots, \pi_n)$ for $\mathbf{P}$ by somehow using the components in ϕ^T to produce π^T with less effort than is required by working blind (i.e., by computing π^T without knowledge of ϕ^T).

When the updating involves only perturbing entries in $\mathbf{Q}$ (i.e., no states are added or deleted), the problem is referred to as an *element-updating problem*. If states need to be added or deleted, the problem is called a *state-updating problem*. The state-updating problem is clearly more difficult, and it generally includes the element-updating problem as a special case. Our purpose is to present a general-purpose algorithm that simultaneously handles both kinds of updating problems. But before presenting our algorithm, we review the shortcomings of some existing approaches to updating.

2. Restarting the power method. Even for simple element updating, the restarted power method is not an overly effective technique. Suppose that the updating process calls for perturbing transition probabilities in $\mathbf{Q}$ to produce the updated matrix $\mathbf{P}$ (but no states are added or deleted), and suppose that it is known that

*Received by the editors November 16, 2004; accepted for publication (in revised form) by M. Benzi August 1, 2005; published electronically February 16, 2006. This research was supported in part by NSF grants CCR-ITR-0113121 and CCR-0318575.

<http://www.siam.org/journals/simax/27-4/61902.html>

[†]Department of Mathematics, College of Charleston, Charleston, SC 29424 (langvillea@cofc.edu).

[‡]Department of Mathematics, North Carolina State University, Raleigh, NC 27695-8205 (meyer@ncsu.edu).

the updated stationary distribution π^T for $\mathbf{P}$ is in some sense close to the original stationary distribution ϕ^T for $\mathbf{Q}$. For example, this might occur if the perturbations to $\mathbf{Q}$ are small. It is intuitive that if ϕ^T and π^T are close, then applying

$$(1) \quad \mathbf{x}_{j+1}^T = \mathbf{x}_j^T \mathbf{P} \quad \text{with} \quad \mathbf{x}_0^T = \phi^T$$

should produce an accurate approximation to π^T in fewer iterations than are required when an arbitrary initial vector is used. To some extent this is true, but intuition generally overestimates the impact.

It is well known that if λ_2 is the subdominant eigenvalue of $\mathbf{P}$, and if λ_2 has index one (linear elementary divisors), then the asymptotic rate of convergence [29, p. 621] of (1) is

$$(2) \quad R = -\log_{10} |\lambda_2|.$$

For linear stationary iterative procedures the asymptotic rate of convergence R is an indication of the number of digits of accuracy that can be expected to be eventually gained on each iteration, and this is independent of the initial vector. For example, suppose that the entries of $\mathbf{P} - \mathbf{Q}$ are small enough to ensure that each component π_i agrees with ϕ_i in the first significant digit, and suppose that the goal is to compute the update π^T to twelve significant places by using (1). Since $\mathbf{x}_0^T = \phi^T$ already has one correct significant digit, and since about $1/R$ iterations are required to gain each additional significant digit of accuracy, (1) requires about $11/R$ iterations, whereas starting from scratch with an initial vector containing no significant digits of accuracy requires about $12/R$ iterations. In other words, the effort is reduced by about 8% for each correct significant digit that can be built into $\mathbf{x}_0^T$. This dictates how much effort should be invested in determining a “good” initial vector. Of course, if one is willing to settle for fewer digits of accuracy or the number of agreeing digits is higher, then the savings could be more substantial.

To appreciate what this means concerning the effectiveness of using (1) as an updating technique, suppose, for example, that $|\lambda_2| = .85$, and suppose that the perturbations involved in updating $\mathbf{Q}$ to $\mathbf{P}$ are such that each component π_i agrees with ϕ_i in the first significant digit. If (1) is used to produce twelve significant digits of accuracy, then it follows from (2) that about 156 iterations are required. This is only about 16 fewer than are needed when starting blind with a random initial vector. Consequently, the power method is not an attractive approach to the element-updating problem even when changes are relatively small. While grossly ineffective, state updating can be accomplished by restarting the power method with an updated $\mathbf{Q}$ and an initial vector obtained by renormalizing results from prior computations after appropriate components are added or deleted. In other words, the restarted power method is not a viable technique for either element updating or state updating.

2.1. Faster converging states. There are times (e.g., see section 8) when there is a need to isolate the faster evolving states of a chain from the slower ones. This is especially true for chains such as Google’s PageRank application (discussed later) in which there are a relatively small number of slower evolving components that drag the entire limiting process down. The asymptotic rate of convergence (2) of the power method is of little help here because it is by design a conservative measure that accounts for the overall evolution rate of the process, and hence must account for the slowest converging states.

To get a sense of what determines which components converge faster than others, let $\{1, \lambda_2, \dots, \lambda_k\}$ be the *distinct* eigenvalues of $\mathbf{P}$, and suppose that $\mathbf{P}$ has a standard

spectral decomposition [15], [29, p. 517]

$$\mathbf{P} = \sum_{i=1}^k \lambda_i \mathbf{G}_i = \mathbf{e} \boldsymbol{\pi}^T + \sum_{i=2}^k \lambda_i \mathbf{G}_i \implies \mathbf{P}^n = \sum_{i=1}^k \lambda_i^n \mathbf{G}_i = \mathbf{e} \boldsymbol{\pi}^T + \sum_{i=2}^k \lambda_i^n \mathbf{G}_i,$$

where $\mathbf{e} = (1, 1, \dots, 1)^T$, $\mathbf{G}_i$ is the i th spectral projector, and $1 > |\lambda_2| \geq |\lambda_3| \geq \dots \geq |\lambda_k|$. If λ_2 is a simple eigenvalue, then $\mathbf{G}_2 = \mathbf{u}_2 \mathbf{v}_2^T / \mathbf{v}_2^T \mathbf{u}_2$, where $\mathbf{v}_2^T$ and $\mathbf{u}_2$ are left and right eigenvectors associated with λ_2 , respectively, and

$$(3) \quad \mathbf{x}_n^T = \mathbf{x}_0^T \mathbf{P}^n \mathbf{x}_0^T \mathbf{G}_i = \boldsymbol{\pi}^T + \xi \lambda_2^n \mathbf{v}_2^T + \sum_{i=3}^k \lambda_i^n \mathbf{x}_0^T \mathbf{G}_i,$$

where $\xi = \mathbf{x}_0^T \mathbf{u}_2 / \mathbf{v}_2^T \mathbf{u}_2$. If we exclude the cases in which $|\lambda_2| = |\lambda_3|$, $\xi = 0$ (i.e., $\mathbf{x}_0 \perp \mathbf{u}_2$), and that components in $\xi \lambda_2^n \mathbf{v}_2^T$ are canceled out by corresponding components in the remainder of the sum (all of which are unlikely in practice), then it is clear from (3) that a given component in $\mathbf{x}_n^T$ will converge at a rate faster than that dictated by (2) if and only if the corresponding component in the left-hand eigenvector $\mathbf{v}_2^T$ is zero. In other words, *the positions of the zero (or near-zero) entries in $\mathbf{v}_2^T$ dictate the faster evolving components*. There is a special class of matrices for which $\mathbf{v}_2^T$ is known to have many components of very small magnitude. We postpone discussion of these until section 8.2.

3. Exact updating. The purpose of this section is to show that element updating can, in fact, always be accomplished by means of a simple and direct formula that is guaranteed to return *exact* results (in exact arithmetic), even when perturbations are allowed to be large. However, you may not want to pay the computational cost to obtain exact results when significant updating is needed.

Consider perturbing some transition probabilities in $\mathbf{Q}$ (irreducible and stochastic) to produce an updated matrix $\mathbf{P}$ (also irreducible and stochastic) without adding or deleting states, but no longer constrain the perturbations to be small. Instead of considering perturbations that are small in magnitude, consider perturbations that affect only a small number of states.

The problem is cast in terms of updating $\mathbf{Q}$ one row at a time. The idea is similar to application of the Sherman–Morrison formula [29, p. 124] for updating a solution to a nonsingular linear system, but the techniques must be adapted to the singular matrix $\mathbf{A} = \mathbf{I} - \mathbf{Q}$. The mechanism for doing this is by means of the group inverse $\mathbf{A}^\#$ for $\mathbf{A}$, which is the unique matrix satisfying the three equations $\mathbf{A} \mathbf{A}^\# \mathbf{A} = \mathbf{A}$, $\mathbf{A}^\# \mathbf{A} \mathbf{A}^\# = \mathbf{A}^\#$, and $\mathbf{A} \mathbf{A}^\# = \mathbf{A}^\# \mathbf{A}$. This matrix is often involved in questions concerning Markov chains; see [6, 23, 29] for some general background and [6, 9, 11, 14, 23, 25, 27, 28, 31, 32, 38] for Markov chain applications. The precise formula to perform exact updating is as follows.

THEOREM 3.1. *Let $\mathbf{Q}$ be the transition probability matrix of an irreducible Markov chain and suppose that the i th row $\mathbf{q}^T$ of $\mathbf{Q}$ is updated to produce $\mathbf{p}^T = \mathbf{q}^T - \boldsymbol{\delta}^T$, the i th row of $\mathbf{P}$, which is also the transition matrix of an irreducible chain. If $\boldsymbol{\phi}^T$ and $\boldsymbol{\pi}^T$ denote the stationary probability distributions of $\mathbf{Q}$ and $\mathbf{P}$, respectively, and if $\mathbf{A} = \mathbf{I} - \mathbf{Q}$, then $\boldsymbol{\pi}^T = \boldsymbol{\phi}^T - \boldsymbol{\epsilon}^T$, where*

$$(4) \quad \boldsymbol{\epsilon}^T = \left[\frac{\phi_i}{1 + \boldsymbol{\delta}^T \mathbf{A}_{*i}^\#} \right] \boldsymbol{\delta}^T \mathbf{A}^\# \quad (\mathbf{A}_{*i}^\# = \text{the } i\text{th column of } \mathbf{A}^\#).$$

To handle multiple row updates to $\mathbf{Q}$, this formula must be sequentially applied one row at a time, which means that the group inverse must be sequentially updated. The formula for updating $(\mathbf{I} - \mathbf{Q})^\#$ to $(\mathbf{I} - \mathbf{P})^\#$ is as follows:

$$(5) \quad (\mathbf{I} - \mathbf{P})^\# = \mathbf{A}^\# + \mathbf{e}\mathbf{e}^T [\mathbf{A}^\# - \gamma\mathbf{I}] - \frac{\mathbf{A}_{*i}^\# \mathbf{e}^T}{\phi_i}, \quad \text{where } \gamma = \frac{\mathbf{e}^T \mathbf{A}_{*i}^\#}{\phi_i}.$$

Since exact updating is not the primary focus of this article, the formal proof of Theorem 3.1 is omitted, but the interested reader can find the details that constitute a proof in [31].

While Theorem 3.1 provides theoretical answers to the element updating problem, it is not computationally satisfying, especially if more than just one or two rows are involved. If every row needs to be touched, then using formulas (4) and (5) requires $O(n^3)$ floating point operations, which is comparable to the cost of starting from scratch.

Other updating formulas exist [9, 12, 16, 20, 36], but all are variations of the same rank-one updating idea involving a Sherman–Morrison [13, 15], [29, p. 124] type of formula, and all are $O(n^3)$ algorithms for a general update. Moreover, all of these rank-one updating techniques apply only to the simpler element-updating problem and are not easily adapted to handle more complicated state-updating problems. The bottom line is that while exact element-updating formulas might be useful when only a row or two need to be changed and no states are added or deleted, they are not practical for making more general updates.

4. Approximate updating using approximate aggregation. If, instead of aiming for the exact value of the updated stationary distribution, one is willing to settle for an approximation, then the door opens wider. For example, an approximation approach based on state-lumping has been suggested in [7] to estimate Google’s PageRank vector. State-lumping is part of a well-known class of methods known as *approximate aggregation techniques* [37] that have been used in the past to estimate stationary distributions of nearly uncoupled chains. Even though it produces only estimates of $\boldsymbol{\pi}^T$, approximate aggregation can handle both element updating as well as state updating and is computationally cheap.

The idea behind the application of approximate aggregation to perform updating is to use the previously known distribution

$$\boldsymbol{\phi}^T = (\phi_1, \phi_2, \dots, \phi_m)$$

together with the updated transition probabilities in $\mathbf{P}$ to build an aggregated Markov chain having a transition probability matrix $\mathbf{A}$ that is smaller in size than $\mathbf{P}$. The stationary distribution $\boldsymbol{\alpha}^T$ of $\mathbf{A}$ is used to generate an estimate of the true updated distribution $\boldsymbol{\pi}^T$ as outlined below.

The state space $\mathcal{S}$ of the updated Markov chain is first partitioned as $\mathcal{S} = G \cup \overline{G}$, where G is the subset of states whose stationary probabilities are likely to be most affected by the updates (newly added states are automatically included in G , and deleted states are accounted for by changing affected transition probabilities to zero). The complement $\overline{G}$ naturally contains all other states. The intuition is that the effect of perturbations involving only a few states in large sparse chains (such as those in Google’s PageRank application) is primarily local, and most stationary probabilities are not significantly affected. Deriving good methods for determining G is a pivotal issue, and this is discussed in detail in section 8.

Partitioning the states of the updated chain as $\mathcal{S} = G \cup \overline{G}$ induces a partition (and reordering) of the updated transition matrix

$$(6) \quad \mathbf{P}_{n \times n} = \begin{matrix} & \begin{matrix} G & \overline{G} \end{matrix} \\ \begin{matrix} G \\ \overline{G} \end{matrix} & \begin{pmatrix} \mathbf{P}_{11} & \mathbf{P}_{12} \\ \mathbf{P}_{21} & \mathbf{P}_{22} \end{pmatrix} \end{matrix},$$

where $\mathbf{P}_{11}$ is $g \times g$ with $g = |G|$ being the cardinality of G and $\mathbf{P}_{22}$ is $(n-g) \times (n-g)$. Similarly, the associated stationary distribution

$$\boldsymbol{\pi}^T = (\pi_1, \dots, \pi_g \mid \pi_{g+1}, \dots, \pi_n) = (\pi_1, \dots, \pi_g \mid \overline{\boldsymbol{\pi}}^T)$$

is reordered and partitioned in an associated manner.

The stationary probabilities from the original distribution $\boldsymbol{\phi}^T$ that correspond to the states in $\overline{G}$ are placed in a row vector $\overline{\boldsymbol{\phi}}^T$, and the states in $\overline{G}$ are lumped into one superstate to create a smaller aggregated Markov chain whose transition matrix is the $(g+1) \times (g+1)$ matrix given by

$$(7) \quad \tilde{\mathbf{A}} = \begin{pmatrix} \mathbf{P}_{11} & \mathbf{P}_{12}\mathbf{e} \\ \tilde{\mathbf{s}}^T \mathbf{P}_{21} & 1 - \tilde{\mathbf{s}}^T \mathbf{P}_{21}\mathbf{e} \end{pmatrix}, \text{ where } \tilde{\mathbf{s}}^T = \frac{\overline{\boldsymbol{\phi}}^T}{\boldsymbol{\phi}^T \mathbf{e}} \text{ (}\mathbf{e} \text{ is a column of ones).}$$

The approximation procedure in [7] computes the stationary distribution

$$\tilde{\boldsymbol{\alpha}}^T = (\tilde{\alpha}_1, \tilde{\alpha}_2, \dots, \tilde{\alpha}_g, \tilde{\alpha}_{g+1})$$

for $\tilde{\mathbf{A}}$ and uses the first g components in $\tilde{\boldsymbol{\alpha}}^T$ along with those in $\overline{\boldsymbol{\phi}}^T$ to create an approximation $\tilde{\boldsymbol{\pi}}^T$ to the true updated distribution $\boldsymbol{\pi}^T$ by setting

$$(8) \quad \tilde{\boldsymbol{\pi}}^T = (\tilde{\alpha}_1, \tilde{\alpha}_2, \dots, \tilde{\alpha}_g \mid \overline{\boldsymbol{\phi}}^T).$$

In other words, use

$$(9) \quad \pi_i \approx \tilde{\pi}_i = \begin{cases} \tilde{\alpha}_i & \text{if state } i \text{ belongs to } G, \\ \phi_i & \text{if state } i \text{ belongs to } \overline{G}. \end{cases}$$

The reason that (9) can produce good estimates is because it can be demonstrated (see section 6) that when there is absolutely no change in the stationary probabilities that correspond to states in $\overline{G}$ (i.e., when $\overline{\boldsymbol{\phi}}^T = \overline{\boldsymbol{\pi}}^T$), then

$$\tilde{\alpha}_i = \begin{cases} \pi_i & \text{for } 1 \leq i \leq g, \\ \overline{\boldsymbol{\pi}}^T \mathbf{e} & \text{for } i = g+1. \end{cases}$$

Therefore, when there is only a small change in the stationary probabilities that correspond to states in $\overline{G}$ (i.e., when $\overline{\boldsymbol{\phi}}^T \approx \overline{\boldsymbol{\pi}}^T$), it is reasonable to expect that $\tilde{\alpha}_i \approx \pi_i$ for $1 \leq i \leq g$. This along with $\overline{\boldsymbol{\phi}}^T \mathbf{e} \approx \overline{\boldsymbol{\pi}}^T \mathbf{e}$ also ensures that $\tilde{\boldsymbol{\pi}}^T \mathbf{e} \approx 1$, and thus the approximation (8) can be close enough to a probability vector to avoid the need for renormalization. The accuracy of this approximation scheme along with other theoretical details is discussed in sections 5 and 6.

5. Exact aggregation. The technique described in section 4 is simply one particular way to approximate the results of *exact* aggregation that is developed in [26] and is briefly outlined below. For an irreducible n -state Markov chain whose state space has been partitioned into k disjoint groups $\mathcal{S} = G_1 \cup G_2 \cup \dots \cup G_k$, the associated transition probability matrix assumes the block-partitioned form

$$(10) \quad \mathbf{P}_{n \times n} = \begin{matrix} & \begin{matrix} G_1 & G_2 & \cdots & G_k \end{matrix} \\ \begin{matrix} G_1 \\ G_2 \\ \vdots \\ G_k \end{matrix} & \begin{pmatrix} \mathbf{P}_{11} & \mathbf{P}_{12} & \cdots & \mathbf{P}_{1k} \\ \mathbf{P}_{21} & \mathbf{P}_{22} & \cdots & \mathbf{P}_{2k} \\ \vdots & \vdots & \ddots & \vdots \\ \mathbf{P}_{k1} & \mathbf{P}_{k2} & \cdots & \mathbf{P}_{kk} \end{pmatrix} \end{matrix} \quad (\text{with square diagonal blocks}).$$

This *parent* Markov chain defined by $\mathbf{P}$ induces k smaller Markov chains, called *censored chains*, as follows. The *censored Markov chain* associated with a group of states G_i is defined to be the Markov process that records the location of the parent chain only when the parent chain visits states in G_i . Visits to states outside of G_i are ignored. The transition probability matrix for the i th censored chain is the i th *stochastic complement* [26]

$$(11) \quad \mathbf{S}_i = \mathbf{P}_{ii} + \mathbf{P}_{i\star}(\mathbf{I} - \mathbf{P}_i^*)^{-1}\mathbf{P}_{\star i},$$

in which $\mathbf{P}_{i\star}$ and $\mathbf{P}_{\star i}$ are, respectively, the i th row and the i th column of blocks with $\mathbf{P}_{ii}$ removed, and $\mathbf{P}_i^*$ is the principal submatrix of $\mathbf{P}$ obtained by deleting the i th row and i th column of blocks. For example, if the partition consists of just two groups $\mathcal{S} = G_1 \cup G_2$, then there are only two censored chains, and their respective transition matrices are the two stochastic complements

$$\mathbf{S}_1 = \mathbf{P}_{11} + \mathbf{P}_{12}(\mathbf{I} - \mathbf{P}_{22})^{-1}\mathbf{P}_{21} \quad \text{and} \quad \mathbf{S}_2 = \mathbf{P}_{22} + \mathbf{P}_{21}(\mathbf{I} - \mathbf{P}_{11})^{-1}\mathbf{P}_{12}.$$

If the stationary distribution for $\mathbf{P}$ is $\boldsymbol{\pi}^T = (\boldsymbol{\pi}_1^T | \boldsymbol{\pi}_2^T | \dots | \boldsymbol{\pi}_k^T)$ (partitioned conformably with $\mathbf{P}$), then the i th *censored distribution* (the stationary distribution for $\mathbf{S}_i$) is known [26] to be equal to

$$(12) \quad \mathbf{s}_i^T = \frac{\boldsymbol{\pi}_i^T}{\boldsymbol{\pi}_i^T \mathbf{e}} \quad (\mathbf{e} \text{ is an appropriately sized column of ones}).$$

For primitive chains [29, p. 693] (also known as aperiodic or regular chains [20, 36, 35]), the j th component of $\mathbf{s}_i^T$ is the limiting conditional probability of being in the j th state of group G_i given that the process is somewhere in G_i .

To compress each group G_i into a single state in order to create a small k -state aggregated chain, squeeze the parent transition matrix $\mathbf{P}$ down to the *aggregated transition matrix* (sometimes called the *coupling matrix*) by setting

$$(13) \quad \mathbf{A}_{k \times k} = \begin{pmatrix} \mathbf{s}_1^T \mathbf{P}_{11} \mathbf{e} & \cdots & \mathbf{s}_1^T \mathbf{P}_{1k} \mathbf{e} \\ \vdots & \ddots & \vdots \\ \mathbf{s}_k^T \mathbf{P}_{k1} \mathbf{e} & \cdots & \mathbf{s}_k^T \mathbf{P}_{kk} \mathbf{e} \end{pmatrix}.$$

If $\mathbf{P}$ is stochastic and irreducible, then so is $\mathbf{A}$ [26]. For primitive chains, transitions between states in the aggregated chain defined by $\mathbf{A}$ correspond to transitions between groups G_i in the unaggregated parent chain when the parent chain is in equilibrium.

The remarkable feature surrounding this aggregation idea is that it allows a parent chain to be decomposed into k small censored chains that can be independently solved, and the resulting censored distributions $\mathbf{s}_i^T$ can be combined with the stationary distribution of $\mathbf{A}$ to construct the parent stationary distribution $\boldsymbol{\pi}^T$. This is the exact aggregation theorem.

THEOREM 5.1 (exact aggregation [26]). *If $\mathbf{P}$ is the block-partitioned transition probability matrix (10) for an irreducible n -state Markov chain whose stationary probability distribution is*

$$\boldsymbol{\pi}^T = \boldsymbol{\pi}_1^T | \boldsymbol{\pi}_2^T | \cdots | \boldsymbol{\pi}_k^T \quad (\text{partitioned conformably with } \mathbf{P}),$$

and if $\boldsymbol{\alpha}^T = (\alpha_1, \alpha_2, \dots, \alpha_k)$ is the stationary distribution for the aggregated chain defined by the matrix $\mathbf{A}_{k \times k}$ in (13), then $\alpha_i = \boldsymbol{\pi}_i^T \mathbf{e}$, and the stationary distribution for $\mathbf{P}$ is

$$\boldsymbol{\pi}^T = (\alpha_1 \mathbf{s}_1^T | \alpha_2 \mathbf{s}_2^T | \cdots | \alpha_k \mathbf{s}_k^T),$$

where $\mathbf{s}_i^T$ is the censored distribution associated with the stochastic complement $\mathbf{S}_i$ in (11).

6. Exact versus approximate aggregation. While exact aggregation as presented in Theorem 5.1 is elegant, it is an inefficient numerical procedure for computing $\boldsymbol{\pi}^T$ because costly inversions are embedded in the stochastic complements (11) that are required to produce the censored distributions $\mathbf{s}_i^T$. Consequently, it is common to attempt to somehow approximate the censored distributions, and there are at least two methods for doing so.

1. Sometimes the stochastic complements $\mathbf{S}_i$ are first estimated, and then the distributions of these estimates are computed to provide approximate censored distributions, which in turn leads to an approximate aggregated transition matrix that is used to produce an approximation to $\boldsymbol{\pi}^T$ by employing Theorem 5.1.
2. The other approach is to bypass the stochastic complements altogether and somehow estimate the censored distributions $\mathbf{s}_i^T$ directly. This is the essence of the approximation scheme described in section 4.

To understand the application of the second approach given above, consider the updated transition matrix $\mathbf{P}$ given in (6) to be partitioned into $g + 1$ levels in which the first g diagonal blocks are just 1×1 , and the lower right-hand block is the $(n - g) \times (n - g)$ matrix $\mathbf{P}_{22}$ associated with the states in $\overline{G}$. In other words, to fit the context of the Theorem 5.1, the partition in (6) is viewed as

$$(14) \quad \mathbf{P} = \begin{matrix} & \begin{matrix} G & \overline{G} \end{matrix} \\ \begin{matrix} G \\ \overline{G} \end{matrix} & \begin{pmatrix} \mathbf{P}_{11} & \mathbf{P}_{12} \\ \mathbf{P}_{21} & \mathbf{P}_{22} \end{pmatrix} \end{matrix} = \left(\begin{array}{c|c|c|c} p_{11} & \cdots & p_{1g} & \mathbf{P}_{1\star} \\ \hline \vdots & \ddots & \vdots & \vdots \\ \hline p_{g1} & \cdots & p_{gg} & \mathbf{P}_{g\star} \\ \hline \mathbf{P}_{\star 1} & \cdots & \mathbf{P}_{\star g} & \mathbf{P}_{22} \end{array} \right),$$

where

$$\mathbf{P}_{11} = \begin{pmatrix} p_{11} & \cdots & p_{1g} \\ \vdots & \ddots & \vdots \\ p_{g1} & \cdots & p_{gg} \end{pmatrix}, \quad \mathbf{P}_{12} = \begin{pmatrix} \mathbf{P}_{1\star} \\ \vdots \\ \mathbf{P}_{g\star} \end{pmatrix}, \quad \text{and} \quad \mathbf{P}_{21} = (\mathbf{P}_{\star 1} \cdots \mathbf{P}_{\star g}).$$

Since the first g diagonal blocks in the partition (14) have size 1×1 (they are the scalars p_{ii}), it is evident that their corresponding stochastic complements are $\mathbf{S}_i = 1$ (because they are 1×1 stochastic matrices), and thus the censored distributions are $\mathbf{s}_i^T = 1$ for $1 \leq i \leq g$. This means that the *exact* aggregated transition matrix (13) associated with the partition (14) is

$$(15) \quad \mathbf{A} = \begin{pmatrix} p_{11} & \cdots & p_{1g} & \mathbf{P}_{1\star}\mathbf{e} \\ \vdots & \ddots & \vdots & \vdots \\ p_{g1} & \cdots & p_{gg} & \mathbf{P}_{g\star}\mathbf{e} \\ \mathbf{s}^T\mathbf{P}_{\star 1} & \cdots & \mathbf{s}^T\mathbf{P}_{\star g} & \mathbf{s}^T\mathbf{P}_{22}\mathbf{e} \end{pmatrix}_{(g+1) \times (g+1)}$$

$$= \begin{pmatrix} \mathbf{P}_{11} & \mathbf{P}_{12}\mathbf{e} \\ \mathbf{s}^T\mathbf{P}_{21} & \mathbf{s}^T\mathbf{P}_{22}\mathbf{e} \end{pmatrix} = \begin{pmatrix} \mathbf{P}_{11} & \mathbf{P}_{12}\mathbf{e} \\ \mathbf{s}^T\mathbf{P}_{21} & 1 - \mathbf{s}^T\mathbf{P}_{21}\mathbf{e} \end{pmatrix},$$

where $\mathbf{s}^T$ is the censored distribution derived from the only significant stochastic complement

$$\mathbf{S} = \mathbf{P}_{22} + \mathbf{P}_{21}(\mathbf{I} - \mathbf{P}_{11})^{-1}\mathbf{P}_{12}.$$

If the stationary distribution for $\mathbf{A}$ is

$$\boldsymbol{\alpha}^T = (\alpha_1, \dots, \alpha_g, \alpha_{g+1}),$$

then exact aggregation (Theorem 5.1) ensures that the *exact* stationary distribution for $\mathbf{P}$ is

$$(16) \quad \boldsymbol{\pi}^T = (\pi_1, \dots, \pi_g \mid \pi_{g+1}, \dots, \pi_n) = (\pi_1, \dots, \pi_g \mid \bar{\boldsymbol{\pi}}^T) = (\alpha_1, \dots, \alpha_g \mid \alpha_{g+1}\mathbf{s}^T).$$

It is a fundamental issue to describe just how well the estimate $\tilde{\boldsymbol{\pi}}^T$ given in (8) approximates the exact distribution $\boldsymbol{\pi}^T$ given in (16). Obviously, the degree to which $\tilde{\pi}_i \approx \pi_i$ for $i > g$ (i.e., the degree to which $\bar{\boldsymbol{\phi}}^T \approx \bar{\boldsymbol{\pi}}^T$) depends on the degree to which the partition $\mathcal{S} = G \cup \bar{G}$ can be adequately constructed. While it is somewhat intuitive that this should also affect the degree to which $\tilde{\pi}_i$ approximates π_i for $i \leq g$, it is not clear, at least on the surface, just how good this latter approximation is expected to be. The analysis is as follows.

Instead of using the exact censored distribution $\mathbf{s}^T$ to build the exact aggregated matrix $\mathbf{A}$ in (15), the vector $\tilde{\mathbf{s}}^T = \bar{\boldsymbol{\phi}}^T / \bar{\boldsymbol{\phi}}^T \mathbf{e}$ is used to approximate $\mathbf{s}^T$ in order to construct the approximate aggregate $\tilde{\mathbf{A}}$ given in (7). The magnitude of

$$\boldsymbol{\delta}^T = \mathbf{s}^T - \tilde{\mathbf{s}}^T = \frac{\bar{\boldsymbol{\pi}}^T}{\bar{\boldsymbol{\pi}}^T \mathbf{e}} - \frac{\bar{\boldsymbol{\phi}}^T}{\bar{\boldsymbol{\phi}}^T \mathbf{e}}$$

and the magnitude of

$$(17) \quad \mathbf{E} = \mathbf{A} - \tilde{\mathbf{A}} = \begin{pmatrix} \mathbf{0} & \mathbf{0} \\ \boldsymbol{\delta}^T \mathbf{P}_{21} & -\boldsymbol{\delta}^T \mathbf{P}_{21} \mathbf{e} \end{pmatrix} = \begin{pmatrix} \mathbf{0} \\ \boldsymbol{\delta}^T \end{pmatrix} \mathbf{P}_{21} (\mathbf{I} \mid -\mathbf{e})$$

are clearly of the same order. This suggests that if the partition $\mathcal{S} = G \cup \overline{G}$ can be adequately constructed so as to ensure that the magnitude of δ^T is small, then $\tilde{\mathbf{A}}$ is close to $\mathbf{A}$, so their respective stationary distributions $\tilde{\alpha}^T$ and α^T should be close, thus ensuring that $\tilde{\pi}_i$ and π_i are close for $i \leq g$.

However, Markov chains can exhibit sensitivities to small perturbations when a subdominant eigenvalue of the transition probability matrix is close to 1 [28, 30] or when there are large mean first passage times in the chain [8], and several measures of “condition” have been developed to gauge these situations [9, 11, 14, 18, 24].

The point being made here is that unless the degree to which $\mathbf{A}$ is well conditioned is established, the degree of the approximation in (8) is in doubt regardless of how close $\overline{\phi}^T$ is to $\overline{\pi}^T$. This may seem to be a criticism of the idea behind the approximation (8), but, to the contrary, the purpose of this article is to argue that this is a good idea because it can be viewed as the first step in an *iterative* aggregation scheme that performs remarkably well. The following sections are dedicated to developing an iterative aggregation approach to updating stationary probabilities.

7. Updating with iterative aggregation. Iterative aggregation is an algorithm for solving nearly uncoupled (sometimes called nearly completely decomposable) Markov chains, and it is discussed in detail in [38]. Iterative aggregation is not a general-purpose technique, and it usually does not work for chains that are not nearly uncoupled. However, the ideas can be adapted to the updating problem, and these variations work extremely well, even when applied to Markov chains that are not nearly uncoupled. This is in part due to the fact that the approximate aggregation matrix (7) differs from the exact aggregation matrix (15) in only one row. Our iterative aggregation updating algorithm is described below.

Assume that the stationary distribution

$$\phi^T = (\phi_1, \phi_2, \dots, \phi_m)$$

for some irreducible Markov chain $\mathcal{C}$ is already known, perhaps from prior computations, and suppose that $\mathcal{C}$ needs to be updated. As in earlier sections, let the transition probability matrix and stationary distribution for the updated chain be denoted by $\mathbf{P}$ and

$$\pi^T = (\pi_1, \pi_2, \dots, \pi_n),$$

respectively. The updated matrix $\mathbf{P}$ is assumed to be irreducible. In applications such as computing Google’s PageRank [3], irreducibility is guaranteed because small positive values are added to each entry resulting in $\mathbf{P} > \mathbf{0}$. It is important to note that m is not necessarily equal to n because the updating process allows for the addition or deletion of states as well as the alteration of transition probabilities.

THE ITERATIVE AGGREGATION UPDATING ALGORITHM

Initialization

- i. Partition the states of the updated chain as $\mathcal{S} = G \cup \overline{G}$ and reorder $\mathbf{P}$ as described in (6)
- ii. $\overline{\phi}^T \leftarrow$ the components from ϕ^T that correspond to the states in $\overline{G}$
- iii. $\mathbf{s}^T \leftarrow \overline{\phi}^T / (\overline{\phi}^T \mathbf{e})$ (an initial approximate censored distribution)

Iterate until convergence

1. $\mathbf{A} \leftarrow \begin{pmatrix} \mathbf{P}_{11} & \mathbf{P}_{12}\mathbf{e} \\ \mathbf{s}^T\mathbf{P}_{21} & 1 - \mathbf{s}^T\mathbf{P}_{21}\mathbf{e} \end{pmatrix}_{(g+1) \times (g+1)} \quad (g = |G|)$
2. $\boldsymbol{\alpha}^T \leftarrow (\alpha_1, \alpha_2, \dots, \alpha_g, \alpha_{g+1})$ (the stationary distribution of $\mathbf{A}$)
3. $\boldsymbol{\chi}^T \leftarrow (\alpha_1, \alpha_2, \dots, \alpha_g \mid \alpha_{g+1}\mathbf{s}^T)$
4. $\boldsymbol{\psi}^T \leftarrow \boldsymbol{\chi}^T\mathbf{P}$ (see note following the algorithm)
5. If $\|\boldsymbol{\psi}^T - \boldsymbol{\chi}^T\| < \tau$ for a given tolerance τ , then quit—else $\mathbf{s}^T \leftarrow \boldsymbol{\psi}^T / \boldsymbol{\psi}^T\mathbf{e}$ and go to step 1

Note concerning step 4. Step 4 is necessary because the vector $\boldsymbol{\chi}^T$ generated in step 3 is a fixed point in the sense that if step 4 is omitted and the process is restarted using $\boldsymbol{\chi}^T$ instead of $\boldsymbol{\psi}^T$, then the same $\boldsymbol{\chi}^T$ is simply reproduced at step 3 on each subsequent iteration. Step 4 has two purposes—it moves the iterate off the fixed point while simultaneously contributing to the convergence process. That is, the $\boldsymbol{\psi}^T$ resulting from step 4 can be used to restart the algorithm as well as produce a better approximation because applying a power step makes small progress toward the stationary solution. In the past, some authors [38] have used Gauss–Seidel in place of the power method at step 4.

While precise rates of convergence for general iterative aggregation algorithms are difficult to articulate, the specialized nature of our iterative aggregation updating algorithm allows us to easily establish its rate of convergence. The following theorem shows that this rate is directly dependent on how fast the powers of the one significant stochastic complement $\mathbf{S} = \mathbf{P}_{22} + \mathbf{P}_{21}(\mathbf{I} - \mathbf{P}_{11})^{-1}\mathbf{P}_{12}$ converge. In other words, since $\mathbf{S}$ is an irreducible stochastic matrix, the rate of convergence is completely dictated by the magnitude and Jordan structure of the largest subdominant eigenvalue of $\mathbf{S}$.

THEOREM 7.1. *The iterative aggregation updating algorithm defined above converges to the stationary distribution $\boldsymbol{\pi}^T$ of $\mathbf{P}$ for all partitions $S = G \cup \bar{G}$. The rate at which the iterates converge to $\boldsymbol{\pi}^T$ is exactly the rate at which the powers $\mathbf{S}^n$ converge, which is governed by the magnitude and Jordan structure of largest subdominant eigenvalue $\lambda_2(\mathbf{S})$ of $\mathbf{S}$. If $\lambda_2(\mathbf{S})$ is real and simple, then the asymptotic rate of convergence is $R = -\log_{10} |\lambda_2(\mathbf{S})|$.*

Proof. For any initial probability vector $\mathbf{s}^T(0)$, let $\mathbf{A}(n)$, $\boldsymbol{\alpha}^T(n)$, $\boldsymbol{\chi}^T(n)$, $\boldsymbol{\psi}^T(n)$, and $\mathbf{s}^T(n)$ denote the respective results from steps 1–5 after n iterations of the iterative aggregation updating algorithm. A few straightforward calculations reveal that

$$\begin{aligned} \mathbf{A}(n+1) &= \begin{pmatrix} \mathbf{P}_{11} & \mathbf{P}_{12}\mathbf{e} \\ \mathbf{s}^T(n)\mathbf{P}_{21} & 1 - \mathbf{s}^T(n)\mathbf{P}_{21}\mathbf{e} \end{pmatrix}, \\ \boldsymbol{\alpha}^T(n+1) &= \beta_{n+1} (\mathbf{s}^T(n)\mathbf{P}_{21}(\mathbf{I} - \mathbf{P}_{11})^{-1} \mid 1), \\ &\quad \text{where } \beta_{n+1} = (1 + \mathbf{s}^T(n)\mathbf{P}_{21}(\mathbf{I} - \mathbf{P}_{11})^{-1}\mathbf{e})^{-1} \quad [23, \text{p. 458}], \\ \boldsymbol{\chi}^T(n+1) &= \beta_{n+1} (\mathbf{s}^T(n)\mathbf{P}_{21}(\mathbf{I} - \mathbf{P}_{11})^{-1} \mid \mathbf{s}^T(n)), \\ \boldsymbol{\psi}^T(n+1) &= \beta_{n+1} (\mathbf{s}^T(n)\mathbf{P}_{21}(\mathbf{I} - \mathbf{P}_{11})^{-1} \mid \mathbf{s}^T(n)\mathbf{S}), \\ \mathbf{s}^T(n+1) &= \mathbf{s}^T(n)\mathbf{S} = \mathbf{s}^T(0)\mathbf{S}^n. \end{aligned}$$

This makes it clear that $\mathbf{s}^T(n) \rightarrow \mathbf{s}^T$ (the censored distribution associated with $\mathbf{S}$) independent of the initial value $\mathbf{s}^T(0)$ and that the rate of convergence is exactly the rate at which $\mathbf{S}^n \rightarrow \mathbf{e}\mathbf{s}^T$. As $\mathbf{s}^T(n) \rightarrow \mathbf{s}^T$, we have

$$\beta_n \rightarrow \beta = (1 + \mathbf{s}^T\mathbf{P}_{21}(\mathbf{I} - \mathbf{P}_{11})^{-1}\mathbf{e})^{-1}$$

and

$$\psi^T(n) \rightarrow \beta(\mathbf{s}^T \mathbf{P}_{21}(\mathbf{I} - \mathbf{P}_{11})^{-1} | \mathbf{s}^T) = \boldsymbol{\pi}^T. \quad \square$$

8. Determining the partition. The iterative aggregation updating algorithm is globally convergent, and it never requires more iterations than the power method to attain a given level of convergence [17]. However, iterative aggregation clearly requires more work per iteration than the power method. One iteration of iterative aggregation requires forming the aggregation matrix, solving for its stationary vector, and executing one power iteration. The key to realizing an improvement in iterative aggregation over the power method rests in properly choosing the partition $\mathcal{S} = G \cup \overline{G}$. As Theorem 7.1 shows, good partitions are precisely those that yield a stochastic complement $\mathbf{S} = \mathbf{P}_{22} + \mathbf{P}_{21}(\mathbf{I} - \mathbf{P}_{11})^{-1} \mathbf{P}_{12}$ whose subdominant eigenvalue $\lambda_2(\mathbf{S})$ is small in magnitude.

Experience indicates that as $|G| = g$ (the size of $\mathbf{P}_{11}$) becomes larger, iterative aggregation tends to converge in fewer iterations. But as g becomes larger, each iteration requires more work, so the trick is to strike an acceptable balance. A small g that significantly reduces $|\lambda_2(\mathbf{S})|$ is the ideal situation.

Even for moderately sized problems there is an extremely large number of possible partitions, but there are some useful heuristics that can help guide the choice of G that will produce reasonably good results. For example, a relatively simple approach is to take G to be the set of all states “near” the updates, where “near” might be measured in a graph theoretic sense or else by transient flow (i.e., using the magnitude of entries of $\mathbf{x}_{j+1}^T = \mathbf{x}_j^T \mathbf{P}$ after j iterations, where j is small, say 5 or 10). In the absence of any other information, this naive strategy is at least a good place to start. However, there are usually additional options that lead to even better “G-sets,” and some of these are described below.

8.1. Partitioning by differing time scales. In most applications involving irreducible aperiodic Markov chains the components of the n th step distribution vector do not converge at a uniform rate, and consequently most iterative techniques, including the power method, often spend the majority of the time in resolving a small number of components—the slow ones. The slow converging components can be isolated either by monitoring the process for a few iterations or by theoretical means such as those described in section 2.1. For the PageRank problem, Kamvar et al. [19] have shown experimentally that a trend is set in the first few iterations, so that one can classify a state as slow converging or fast converging after just 10 iterations. If the states corresponding to the slower converging components are placed in G while the faster converging states are lumped into $\overline{G}$, then the iterative aggregation algorithm concentrates its effort on resolving the smaller number of slow converging states.

In loose terms, the effect of steps 1–3 in the iterative aggregation algorithm is essentially to make progress toward achieving an equilibrium (or steady state) for a smaller chain consisting of just the “slow states” in G together with one additional lumped state that accounts for all “fast states” in $\overline{G}$. The power iteration in step 4 moves the entire process ahead on a global basis, so if the slow states in G are substantially resolved by the relatively cheaper steps 1–3, then not many of the more costly global power steps are required to push the entire chain toward its global equilibrium. This is the essence of the original Simon–Ando idea first proposed in 1961 and explained and analyzed in [26, 37]. As $g = |G|$ becomes smaller relative to n , steps 1–3 become significantly cheaper to execute, and the process converges

quite rapidly in both iteration count and wall-clock time. Examples and reports on experiments are given in section 8.3.

In some applications the slow states are particularly easy to identify because they are the ones having the larger stationary probabilities. This is a particularly nice state of affairs for the updating problem because we have the stationary probabilities from the prior period at our disposal, and thus all we have to do to construct a good G -set is to include the states with prior large stationary probabilities and throw in the states that were added or updated along with a few of their nearest neighbors. Clearly, this is an advantage only when there are just a few “large” states. However, it turns out that this is a characteristic feature of Google’s PageRank application and other scale-free networks with power-law distributions. This is explained in the next section.

8.2. Scale-free networks and Google’s PageRank. As discussed in [1, 2, 5, 10], the link structure of the World Wide Web constitutes a *scale-free network*. This means that the number of nodes $n(l)$ having l edges (possibly directed) is proportional to l^{-k} , where k is a constant that does not change as the network expands (hence the term “scale-free”). In other words, the distribution of nodal degrees obeys a *power-law distribution* in the sense that

$$P[\deg(N) = d] \propto \frac{1}{d^k} \quad \text{for some } k > 1 \quad (\propto \text{ means “proportional to”}).$$

For example, studies [1, 2, 5, 10] have shown that for the World Wide Web the parameter for the indegree power-law distribution is $k \approx 2.1$, while the outdegree distribution has $k \approx 2.7$. The growth rate of the World Wide Web is tremendous, and the rapidly accumulating wealth of information contained therein is staggering, so the scale-free nature of the Web becomes important if the Web is to be harnessed.

The vast amount of knowledge would, for the most part, be inaccessible if it were not for Web search engines, and Google is the reigning champion in the search engine business. At the heart of Google is its innovative PageRank concept [3, 4], which is a process for assigning to each Web page a value that determines the order of presentation in reply to a query. PageRanks, in their purest form, are simply stationary probabilities for a particular Markov chain (described below). However, in practice, Google tweaks the mathematical PageRanks with proprietary “metrics” to create the final values that are used when matching a user’s query. Notwithstanding the tweaking, the mathematics is the fundamental component of PageRank, and this is what we focus on.

In the remainder of this article, we consider PageRank to be the stationary probability distribution vector π^T for an irreducible aperiodic Markov chain whose transition probability matrix has the form

$$\mathbf{P}_{n \times n} = \alpha \mathbf{S} + (1 - \alpha) \mathbf{e} \mathbf{v}^T,$$

in which $\mathbf{S} = \mathbf{H} + \mathbf{E}$ is a stochastic matrix involving the Web’s raw hyperlink matrix $\mathbf{H}$ defined by

$$h_{ij} = \begin{cases} 1/(\text{total \# outlinks from page } \mathcal{P}_i) & \text{if } \mathcal{P}_i \text{ contains a link to } \mathcal{P}_j, \\ 0 & \text{otherwise} \end{cases}$$

and a modification matrix $\mathbf{E}$ that accounts for “dangling nodes” (pages with no outlinks). Vector $\mathbf{e}$ is a column of ones, $\mathbf{v}^T$ is a “personalization” probability vector

that gives Google flexibility to perform customization, and $0 < \alpha < 1$ is Google's parameter that models a Web surfer's propensity to deviate from the underlying link structure; i.e., with probability $(1 - \alpha)$ a Web surfer requests a "random" Web page. For more details concerning these and other features of PageRank, see [21, 22].

Of course, the scale is enormous—currently $n = O(10^9)$ —and a recent MATLAB publication [33] characterized PageRank as "the world's largest matrix computation" with execution times measured in days. In building a search engine for a linked database in which the link structure is static (or nearly so), a large computational cost to determine PageRank might be tolerable because once π^T is determined the search engine can use it repeatedly. However, the World Wide Web is a dynamic network in which pages and links between them are being added and deleted almost continuously, so the problem of updating π^T is important and substantial. At the 2002 national SIAM meeting in Philadelphia (which is the last public disclosure that the authors are aware of) Google's Director of Technology Craig Silverstein went on record as saying that at that time it had no more effective way to deal with the global updating of PageRank other than by starting from scratch every three or four weeks. Furthermore, the PageRank vectors from prior periods were not used to determine PageRank for a current period. Google has, no doubt, made progress on this problem since 2002, but they are not talking. While local updating for popular sites seems to be more frequent, observation suggests that the time between complete global updates may still be an issue.

8.3. Experiments with power-law distributions. The scale-free nature of the Web translates into a power-law distribution of PageRanks—experiments described in [10, 34] indicate that PageRank has a power-law distribution whose parameter is $k \approx 2.1$. In other words, there are relatively very few pages that have a significant PageRank, while the overwhelming majority of pages have a nearly negligible PageRank.

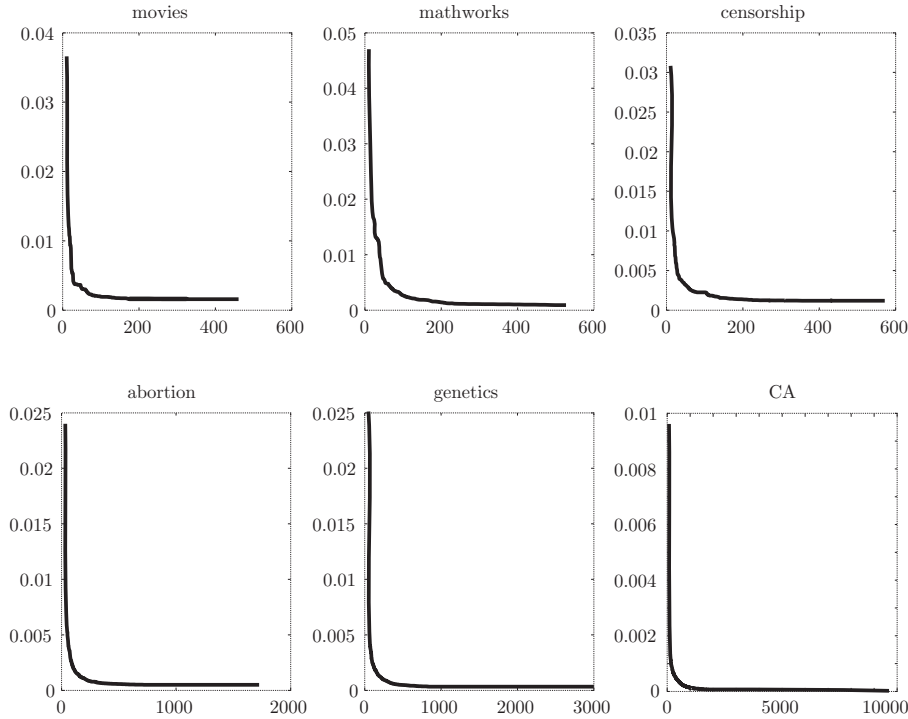
Consequently, when PageRanks are plotted in order of decreasing magnitude, the resulting graph has a pronounced "L-shape" with an extremely sharp bend. It is this characteristic "L-shape" of PageRank distributions that reveals a near optimal partition $\mathcal{S} = G \cup \bar{G}$ for the iterative aggregation updating algorithm described in section 7, hereafter referred to as IAD.

To illustrate this point, we report on experiments derived from Web crawls pertaining to six specialized topics:

Topic	#Pages	#Links
Movies	451	713
MathWorks	517	13,531
Censorship	562	736
Abortion	1,693	4,325
Genetics	2,952	6,485
California	9,664	16,150

When the PageRanks for these datasets are plotted in order of decreasing magnitude, the characteristic L-shapes are apparent. A horizontal axis in Figure 1 represents components of a PageRank vector in order of decreasing magnitude, and on the vertical axis are the corresponding magnitudes of the PageRank.

In an attempt to discern characteristics of good partitions, several experiments were performed on these six datasets. After the initial PageRanks were computed,


 FIG. 1. *PageRanks for six datasets.*

each dataset was updated by adding some new pages and deleting some old ones, in addition to adding some new links and deleting some old links. Except for the larger California dataset, thirty new pages were added and twenty old ones were deleted, while fifty new links were created and twenty old ones were removed. For California, thirty new pages were added, three hundred old ones were removed, two hundred new links were added, and fifty old links were removed.¹ For each dataset, several partitions $\mathcal{S} = G \cup \overline{G}$ were constructed by first placing new states and the states that were altered (along with their nearest neighbors) into G , and then additional states were successively added to G in order of the magnitude of prior PageRanks. The IAD of section 7 was executed in MATLAB for each trial partition. Termination was always when the residual 1-norm dropped below 10^{-10} . The experimental results concerning iteration counts as well as total execution times (in seconds) are reported in Tables 1 and 2.

For the purpose of baseline comparisons, the iteration counts and execution times for the power method are given at the bottom of each table along with the relative improvement afforded by our IAD when the best observed G -partition was employed. A star ($\star$) in a table indicates that there was no experiment at the indicated size $g = |G|$ because g exceeds the number of pages in the dataset.

8.4. Experimental conclusions. While these experiments are small when compared to the scale of the entire World Wide Web, they nevertheless reveal some interesting patterns that are summarized below.

¹Different values here had little effect on the performance of the algorithm.

TABLE 1
Experimental results for Movies, Censorship, and MathWorks.

Movies ($g_{opt} \approx 50$)			MathWorks ($g_{opt} \approx 50$)		Censorship ($g_{opt} \approx 100$)	
$g = G $	Iterations	Time	Iterations	Time	Iterations	Time
5	23	.018	66	.362	41	.062
10	13	.017	64	.329	41	.046
15	13	.016	64	.332	40	.045
20	13	.015	50	.295	20	.024
25	12	.016	46	.291	20	.025
50	12	.015*	19	.207*	13	.019
100	10	.016	18	.224	9	.017*
200	9	.018	16	.284	9	.019
300	9	.021	11	.265	9	.022
400	7	.021	8	.278	9	.026
Power method	22	.017	69	.255	42	.031
*Improvement		11.8%		18.8%		45.2%

TABLE 2
Experimental results for Abortion, Genetics, and California.

Abortion ($g_{opt} \approx 250$)			Genetics ($g_{opt} \approx 250$)		California ($g_{opt} \approx 2000$)	
$g = G $	Iterations	Time	Iterations	Time	Iterations	Time
10	165	.773	163	2.16	170	7.75
50	58	.256	19	.483	75	3.56
100	14	.159	19	.456	57	3.75
250	13	.140*	17	.276*	51	2.59
500	7	.199	9	.313	34	2.01
1000	7	.194	8	.319	19	1.03
2000	*	*	6	.393	10	.997*
3000	*	*	*	*	7	1.17
4000	*	*	*	*	7	1.22
5000	*	*	*	*	7	1.56
Power method	168	.449	165	1.45	176	5.87
*Improvement		68.8%		81%		83%

- G-sets can always exist for which the iterative aggregation technique provides a significant improvement over the power method.
- The improvements become more pronounced as the size of the datasets increases.
- Good G -sets can be constructed by including states affected by updated information along with a few states that are associated with the largest stationary probabilities from the preupdated distribution.
- As $g = |G|$ increases, the performance of IAD (as measured by execution time) improves up to some point, but increasing g beyond this point degrades the

performance of IAD. How to determine a priori a nearly optimal size for G is discussed in section 8.5.

- Finally, when IAD is used as an updating technique, the fact that updates might change the problem size is of little or no consequence. This is an extremely important feature because dealing with issues caused by adding or deleting states is generally a major problem for most updating applications.

8.5. Near optimal G -sets and L -curves. As observed above, IAD performs well by using a G -set that includes states affected by updated information along with states that are associated with the largest stationary probabilities from the preupdated distribution. But since the performance of IAD is dependent on the size of such a G , it is important to have a mechanism that gauges how many of the largest states from the preupdated distribution should be included in G . By examining the approximate values of g_{opt} given in Tables 1 and 2, it is clear that there is somewhat of a pattern that relates g_{opt} to the shape of the L -curve for the power-law distribution. To see this, superimpose the respective values of g_{opt} given in Tables 1 and 2 on the L -curves given in Figure 1. The results are shown in Figure 2.

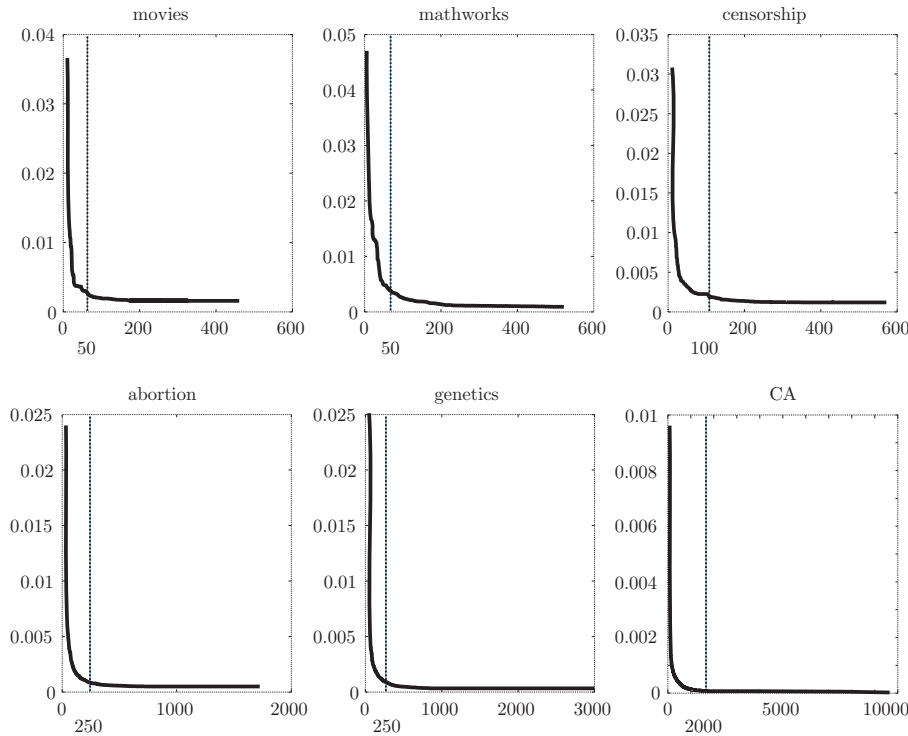


FIG. 2. Location of g_{opt} .

These graphs suggest that when the stationary probabilities of a Markov chain have a power-law distribution, the size of the G -set used in IAD is nearly optimal around a point that is just to the right of the pronounced bend in the L -curve. In other words, an apparent method for constructing a reasonably good partition $\mathcal{S} = G \cup \overline{G}$ for IAD updating is as follows.

1. First put all new states and states with altered links (perhaps along with some nearest neighbors) into G .

2. Add other states that remain after the update in order of the magnitude of their prior stationary probabilities up to the point where these stationary probabilities level off.

Of course, there is some subjectiveness to this strategy. However, the leveling-off point is relatively easy to discern in distributions having a sharply defined bend in the L-curve, and only distributions that gradually die away or do not conform to a power-law distribution are problematic. For example, there is more uncertainty in choosing the leveling-off point in the three smaller test cases (movies, censorship, and MathWorks) than in the three larger ones (abortion, genetics, and California), but being somewhere in the ballpark is generally good enough even when the bend is not so sharply defined. The results in Tables 1 and 2 indicate how much variation around g_{opt} can be tolerated without seriously affecting IAD performance, and in all cases there is a fair amount of leeway.

If, when ordered by magnitude, the stationary probabilities

$$\pi(1) \geq \pi(2) \geq \cdots \geq \pi(n)$$

for an irreducible Markov chain conform to a power-law distribution so that there are constants $\alpha > 0$ and $k > 0$ such that $\pi(i) \approx \alpha i^{-k}$, then the “leveling-off point” i_{level} can be taken to be the smallest value for which $|d\pi(i)/di| \approx \epsilon$ for some user-defined tolerance ϵ . That is, $i_{level} \approx (k\alpha/\epsilon)^{1/k+1}$. This provides a rough estimate of g_{opt} , but empirical evidence suggests that better estimates require a scaling factor $\sigma(n)$ that accounts for the size of the chain; i.e.,

$$g_{opt} \approx \sigma(n) \left(\frac{k\alpha}{\epsilon} \right)^{1/k+1} \approx \sigma(n) \left(\frac{k\pi(1)}{\epsilon} \right)^{1/k+1}.$$

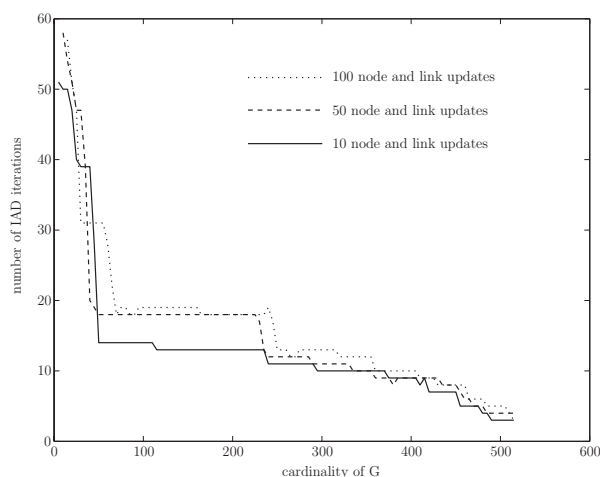
If this is the case, and if the observations from [10, 34] are correct in the sense that PageRanks for the entire World Wide Web conform to a power-law distribution with parameter $k = 2.109$, then we should have that

$$g_{opt} \approx \sigma(n) \left[\frac{2.109\pi(1)}{\epsilon} \right]^{1/3.109}.$$

However, validating this conclusion with a specific scaling factor is beyond the scope of our data and computational resources.

8.6. The drop-off point. Our experiments also indicate that the number of IAD iterations required is a nonlinear function of the size of the G -set. As $g = |G|$ increases, as described in section 8.3, there is initially a sharp drop in the number of IAD iterations required. After the sharp drop there is a more moderate and steady decrease in the iteration count as a function of g . But an even more interesting feature of this phenomenon is that *the sharp drop in iteration count occurs at a point that is more or less independent of the number of nodes or links that are updated.*

For example, the graphs in Figure 3 plot the number of IAD iterations against $g = |G|$ for the MathWorks dataset used in section 8.3 when the updating involves 10, 50, and 100 nodes and links. Regardless of the number of updates, all graphs in Figure 3 exhibit roughly the same shape, with a drop-off point around 40 or 50, which is also about the same value for g_{opt} that was determined from the experiments in Table 1. This phenomenon was generally observed in our other experiments as well.


 FIG. 3. *Drop-off points.*

Consequently, this suggests that the value of g_{opt} can alternately be characterized by saying that g_{opt} is approximately equal to the drop-off point in the iteration count.

Theorem 7.1 established that the asymptotic rate of convergence of IAD is governed by the subdominant eigenvalue λ_2 of the one significant stochastic complement $\mathbf{S}$. Therefore, the drop-off point in the iteration count should be explainable in terms of a drop-off in the magnitude of λ_2 as $g = |G|$ is increased. Indeed, this is corroborated by our experiments. For example, by again using the MathWorks data as a typical case, it is seen in Table 3 that as g increases, the value of $|\lambda_2|$ decreases rapidly until g is somewhere around 40 or 50, after which there is only a slow decrease. And this more or less agrees with the value of g_{opt} that was observed from the data in Table 1.

TABLE 3
 $|\lambda_2|$ as a function of $g = |G|$.

g	10	20	30	35	38	40	45	50	100
$ \lambda_2 $	.7206	.6891	.6610	.6054	.4431	.4018	.4012	.4005	.3857

9. Conclusions and future work. An algorithm for updating the stationary vector of a Markov chain subject to changes in the number of states as well as changes to the transition probabilities has been introduced and analyzed. This IAD updating algorithm exploits the old stationary vector to create the new stationary vector, and numerical experiments suggest it is quite effective. It is superior to the power method in terms of both iteration count as well as total execution time. The IAD approach offers room for even greater improvements. For example, the extrapolation technique introduced in [19] can be employed in conjunction with the IAD introduced in this article to further accelerate the updating process. Preliminary experiments indicate that marrying IAD to extrapolation has remarkable promise—results will be reported in a separate article. Finally, we have demonstrated the applicability of IAD to updating Google’s PageRank vector.

Acknowledgments. We thank Ronny Lempel for generously supplying the three datasets on movies, abortion, and genetics, and we thank Cleve Moler for sharing his MathWorks dataset as well as other Web-crawling M-files.

REFERENCES

- [1] A.-L. BARABASI, *Linked: The New Science of Networks*, Plume, New York, 2003.
- [2] A.-L. BARABASI, R. ALBERT, AND H. JEONG, *Scale-free characteristics of random networks: The topology of the World-Wide Web*, Phys. A, 281 (2000), pp. 69–77.
- [3] S. BRIN AND L. PAGE, *The anatomy of a large-scale hypertextual web search engine*, Comput. Networks ISDN Systems, 33 (1998), pp. 107–117.
- [4] S. BRIN, L. PAGE, R. MOTWAMI, AND T. WINOGRAD, *The PageRank Citation Ranking: Bringing Order to the Web*, Technical report, Computer Science Department, Stanford University, Stanford, CA, 1998.
- [5] A. BRODER, R. KUMAR, F. MAGHOUL, P. RAGHAVAN, S. RAJAGOPALAN, R. STATA, A. TOMKINS, AND J. WIENER, *Graph structure in the web*, in Proceedings of the Ninth International World Wide Web Conference, 2000; available online at <http://www9.org/w9cdrom/160/160.html>.
- [6] S. CAMPBELL AND C. D. MEYER, *Generalized Inverses of Linear Transformations*, Pitman, San Francisco, 1979.
- [7] S. CHIEN, C. DWORK, R. KUMAR, AND D. SIVAKUMAR, *Towards exploiting link evolution*, in Proceedings of the Workshop on Algorithms and Models for the Web Graph, 2001.
- [8] G. E. CHO AND C. D. MEYER, *Markov chain sensitivity measured by mean first passage times*, Linear Algebra Appl., 313 (2000), pp. 21–28.
- [9] G. E. CHO AND C. D. MEYER, *Comparison of perturbation bounds for the stationary distribution of a Markov chain*, Linear Algebra Appl., 335 (2001), pp. 137–150.
- [10] D. DONATO, L. LAURA, S. LEONARDI, AND S. MILLOZZI, *Large scale properties of the webgraph*, Eur. Phys. J. B, 38 (2004), pp. 239–243.
- [11] R. E. FUNDERLIC AND C. D. MEYER, *Sensitivity of the stationary distribution vector for an ergodic Markov chain*, Linear Algebra Appl., 76 (1986), pp. 1–17.
- [12] R. E. FUNDERLIC AND R. J. PLEMMONS, *Updating LU factorizations for computing stationary distributions*, SIAM J. Alg. Disc. Meth., 7 (1986), pp. 30–42.
- [13] G. H. GOLUB AND C. F. V. LOAN, *Matrix Computations*, The Johns Hopkins University Press, Baltimore, 1996.
- [14] G. H. GOLUB AND C. D. MEYER, *Using the QR factorization and group inversion to compute, differentiate, and estimate the sensitivity of stationary probabilities for Markov chains*, SIAM J. Alg. Disc. Meth., 7 (1986), pp. 273–281.
- [15] R. A. HORN AND C. R. JOHNSON, *Matrix Analysis*, Cambridge University Press, Cambridge, UK, 1990.
- [16] J. J. HUNTER, *Stationary distributions of perturbed Markov chains*, Linear Algebra Appl., 82 (1986), pp. 201–214.
- [17] I. C. F. IPSEN AND S. KIRKLAND, *Convergence analysis of an improved PageRank algorithm*, Technical report CRSC-TR04-02, North Carolina State University, Raleigh, NC, 2004.
- [18] I. C. F. IPSEN AND C. D. MEYER, *Uniform stability of Markov chains*, SIAM J. Matrix Anal. Appl., 15 (1994), pp. 1061–1074.
- [19] S. D. KAMVAR, T. H. HAVELIWALA, C. D. MANNING, AND G. H. GOLUB, *Extrapolation methods for accelerating PageRank computations*, in Proceedings of the Twelfth International World Wide Web Conference, 2003; available online at <http://www2003.org/cdrom/index.html>.
- [20] J. G. KEMENY AND L. J. SNELL, *Finite Markov Chains*, Van Nostrand, New York, 1960.
- [21] A. N. LANGVILLE AND C. D. MEYER, *A survey of eigenvector methods for Web information retrieval*, SIAM Rev., 47 (2005), pp. 135–161.
- [22] A. N. LANGVILLE AND C. D. MEYER, *Google's PageRank and Beyond: The Science of Search Engine Rankings*, Princeton University Press, Princeton, NJ, 2006.
- [23] C. D. MEYER JR., *The role of the group generalized inverse in the theory of finite Markov chains*, SIAM Rev., 17 (1975), pp. 443–464.
- [24] C. D. MEYER JR., *The condition of a finite Markov chain and perturbation bounds for the limiting probabilities*, SIAM J. Alg. Disc. Meth., 1 (1980), pp. 273–283.
- [25] C. D. MEYER, *Analysis of finite Markov chains by group inversion techniques*, in Recent Applications of Generalized Inverses, Res. Notes in Math. 66, S. L. Campbell, ed., Pitman, Boston, London, 1982, pp. 50–81.
- [26] C. D. MEYER, *Stochastic complementation, uncoupling Markov chains, and the theory of nearly reducible systems*, SIAM Rev., 31 (1989), pp. 240–272.
- [27] C. D. MEYER, *The character of a finite Markov chain*, in Linear Algebra, Markov Chains, and Queueing Models, IMA Vol. Math. Appl. 48, C. D. Meyer and R. J. Plemmons, eds., Springer-Verlag, New York, 1993, pp. 47–58.
- [28] C. D. MEYER, *Sensitivity of the stationary distribution of a Markov chain*, SIAM J. Matrix Anal. Appl., 15 (1994), pp. 715–728.

- [29] C. D. MEYER, *Matrix Analysis and Applied Linear Algebra*, SIAM, Philadelphia, 2000.
- [30] C. D. MEYER AND R. J. PLEMMONS, *Linear Algebra, Markov Chains, and Queueing Models*, Springer-Verlag, New York, 1993.
- [31] C. D. MEYER AND J. M. SHOAF, *Updating finite Markov chains by using techniques of group matrix inversion*, J. Statist. Comput. Simulation, 11 (1980), pp. 163–181.
- [32] C. D. MEYER AND G. W. STEWART, *Derivatives and perturbations of eigenvectors*, SIAM J. Numer. Anal., 25 (1988), pp. 679–691.
- [33] C. MOLER, *The world's largest matrix computation*, MATLAB News and Notes, October (2002), pp. 12–13.
- [34] G. PANDURANGAN, P. RAGHAVAN, AND E. UPFAL, *Using PageRank to characterize Web structure*, in Computing and Combinatorics, Lecture Notes in Comput. Sci. 2387, Springer-Verlag, Berlin, 2002.
- [35] E. SENETA, *Nonnegative Matrices and Markov Chains*, Springer-Verlag, New York, 1981.
- [36] E. SENETA, *Sensitivity analysis, ergodicity coefficients, and rank-one updates for finite Markov chains*, in Numerical Solution of Markov Chains, W. J. Stewart, ed., Dekker, New York, 1991, pp. 121–129.
- [37] H. A. SIMON AND A. ANDO, *Aggregation of variables in dynamic systems*, Econometrica, 29 (1961), pp. 111–138.
- [38] W. J. STEWART, *Introduction to the Numerical Solution of Markov Chains*, Princeton University Press, Princeton, NJ, 1994.

SOME PROPERTIES OF GENERALIZED POSITIVE SUBDEFINITE MATRICES*

S. K. NEOGY[†] AND A. K. DAS[‡]

This paper is dedicated to late Professor S. R. Mohan

Abstract. The class of generalized positive subdefinite (GPSBD) matrices is an interesting matrix class introduced by Crouzeix and Komlósi [Appl. Optim. 59, Kluwer, Dordrecht, The Netherlands, 2001, pp. 45–63]. In this paper, we obtain some properties of GPSBD matrices. We show that copositive GPSBD matrices are P_0 and a merely generalized positive subdefinite (MGPSBD) matrix with some additional conditions belongs to the class of row sufficient matrices introduced by Cottle, Pang, and Venkateswari [Linear Algebra Appl., 114/115 (1989), pp. 231–249]. Further, it is shown that for a subclass of GPSBD matrices, the solution set of a linear complementarity problem is same as the set of Karush–Kuhn–Tucker-stationary points of the corresponding quadratic programming problem. We provide a counter example to show that a copositive GPSBD matrix need not be sufficient in general. Finally, we show that if a matrix A can be written as a sum of a copositive-plus MGPSBD matrix with an additional condition and a copositive matrix and if it satisfies a feasibility condition, then Lemke’s algorithm can solve $\text{LCP}(q, A)$. This further extends the applicability of Lemke’s algorithm and a result of Evers.

Key words. generalized positive subdefinite matrix, copositive matrix, Lemke’s algorithm, row sufficient matrix, copositive-plus matrix, copositive-star matrix

AMS subject classifications. 90C33, 90C20

DOI. 10.1137/040613585

1. Introduction. Martos [10] introduced the class of symmetric positive subdefinite (PSBD) matrices in connection with a characterization of a pseudoconvex function. Cottle and Ferland [2] followed the path set by Martos in [10] and among other things obtained converses for some of Martos’s results. Later Crouzeix et al. [6] studied nonsymmetric PSBD matrices in the context of generalized monotonicity and the linear complementarity problem. In [11], Mohan, Neogy, and Das observed that a copositive PSBD matrix of rank ≥ 2 is sufficient. It is also shown that if a matrix A is the sum of a merely positive subdefinite copositive-plus matrix and a copositive matrix and a feasibility condition is satisfied, then Lemke’s algorithm solves $\text{LCP}(q, A)$. The solution set of a linear complementarity problem ($S(q, A)$) can be linked with the set of Karush–Kuhn–Tucker- (KKT-) stationary points ($S''(q, A)$) of the corresponding quadratic programming problem. The row sufficient matrices have been characterized by Cottle, Pang, and Venkateswari [5] as the class for which the solution set of $\text{LCP}(q, A)$ is the same as the solution set of KKT points of the corresponding quadratic program. In [7], Crouzeix and Komlósi showed that the property ($S''(q, A) \subseteq S(q, A)$) always holds for generalized positive subdefinite (GPSBD) matrices. However, this result as stated in [7] is incorrect and we show that ($S''(q, A) \subseteq S(q, A)$) holds for GPSBD matrices under some additional assumptions. We produce a counter example to show that the assumption cannot be relaxed further (see section 3). The *linear complementarity problem* is defined below.

*Received by the editors August 18, 2004; accepted for publication (in revised form) by L. Vandenbergh September 2, 2005; published electronically February 16, 2006.

<http://www.siam.org/journals/simax/27-4/61358.html>

[†]Indian Statistical Institute, New Delhi 110016, India (skn@isid.ac.in).

[‡]Indian Statistical Institute, Kolkata 700108, India (akdas@isical.ac.in).

Given a real square matrix A of order n and a vector $q \in R^n$, the linear complementarity problem is to find $w \in R^n$ and $z \in R^n$ such that

$$(1.1) \quad w - Az = q, \quad w \geq 0, \quad z \geq 0,$$

$$(1.2) \quad w^t z = 0.$$

This problem is denoted as $LCP(q, A)$. It is well studied in the literature on mathematical programming and arises in a number of applications in operations research, mathematical economics, engineering, and game theory. In particular, the problem of computing a Karush–Kuhn–Tucker point of a convex quadratic programming problem with linear constraints on the variables can be formulated as an LCP. For further details and applications on this problem, see Cottle, Pang, and Stone [4] and Murty [12].

In this paper, we study GPSBD matrices. In section 2, we introduce the notation and provide the relevant definitions used in this paper. In section 3, we show that a principal submatrix of a GPSBD matrix is also GPSBD and prove that a copositive GPSBD matrix is a P_0 -matrix. A subclass of P_0 occurs in Markov chain analysis and in the study of global univalence in economic theory. See [1] and [14]. We provide a counterexample to show that a copositive GPSBD matrix need not be sufficient. However, a copositive merely generalized positive subdefinite (MGPSBD) matrix with an additional condition on it is shown to be in the class of row sufficient matrices, a matrix class which is invariant under principal pivot transform. In other words, when one performs a principal pivot operation on a row sufficient matrix, the resulting matrix is again row sufficient. See [4] and [15]. Finally, we show that if a matrix A which can be written as a sum of a copositive-plus MGPSBD matrix with an additional condition and a copositive matrix and if it satisfies a feasibility condition, then Lemke's algorithm can solve $LCP(q, A)$.

A large subclass of GPSBD matrices is identified as row sufficient matrices in this paper. This has practical relevance to the study of quadratic programming and interior point algorithm. Our observations on the properties of GPSBD matrices can motivate further applications in matrix theory. In addition, our result mentioned above on the applicability of Lemke's algorithm extends the class of LCP problems solvable by Lemke's algorithm.

2. Preliminaries. We consider matrices and vectors with real entries. Let R_+^n denote the nonnegative orthant in R^n and $R^{n \times n}$ denote the set of all $n \times n$ real matrices. For any matrix $A \in R^{m \times n}$, a_{ij} denotes its i th row and j th column entry. For any matrix $A \in R^{m \times n}$, let A_i denote its i th row and A_j denote its j th column. For any set $\alpha \subseteq \{1, 2, \dots, n\}$, $\bar{\alpha}$ denotes its complement in $\{1, 2, \dots, n\}$. If A is a matrix of order n , $\alpha \subseteq \{1, 2, \dots, n\}$ and $\beta \subseteq \{1, 2, \dots, n\}$, then $A_{\alpha\beta}$ denotes the submatrix of A consisting of only the rows and columns of A whose indices are in α and β , respectively. Any vector $x \in R^n$ is a column vector unless otherwise specified and x^t denotes the row transpose of x . For any matrix $A \in R^{n \times n}$, A^t denotes its transpose. We say that a vector $y \in R^n$ is *unsigned* if either $y \in R_+^n$ or $-y \in R_+^n$. Given a symmetric matrix $S \in R^{n \times n}$, its *inertia* is the triple $(\nu_+(S), \nu_-(S), \nu_0(S))$, where $\nu_+(S), \nu_-(S), \nu_0(S)$ denote the number of positive, negative, and zero eigenvalues of S , respectively. Let A be a given $n \times n$ matrix, not necessarily symmetric. We say that A is *positive semidefinite* (PSD) if $x^t A x \geq 0 \forall x \in R^n$ and A is *positive definite* (PD) if $x^t A x > 0 \forall 0 \neq x \in R^n$. A is called *positive subdefinite matrix* if for all $x \in R^n$, $x^t A x < 0$ implies $A^t x$ is unsigned. A is said to be *merely positive subdefinite*

(MPSBD) if A is a PSBD matrix but not a PSD matrix. A is said to be a $P(P_0)$ -matrix if all its principal minors are positive (nonnegative). $A \in R^{n \times n}$ is said to be *copositive* (C_0) if $x^t Ax \geq 0 \forall x \geq 0$. A is said to be *copositive-plus* (C_0^+) if $A \in C_0$ and the following implication holds:

$$[x^t Ax = 0, x \geq 0] \Rightarrow (A + A^t)x = 0.$$

We say that $A \in R^{n \times n}$ is *copositive-star* (C_0^*) if $A \in C_0$ and the following implication holds:

$$[x^t Ax = 0, Ax \geq 0, x \geq 0] \Rightarrow A^t x \leq 0.$$

A is said to be *column sufficient* if for all $x \in R^n$ the following implication holds:

$$x_i(Ax)_i \leq 0 \forall i \Rightarrow x_i(Ax)_i = 0 \forall i.$$

A is said to be *row sufficient* if A^t is column sufficient. A is *sufficient* if A and A^t are both column sufficient. For details on sufficient matrices, see [3], [5], and [16]. A matrix $A \in R^{n \times n}$ is called a Q -matrix if for every $q \in R^n$, $\text{LCP}(q, A)$ has a solution. We say that a matrix A is a Q_0 -matrix if for any $q \in R^n$, (1.1) has a solution implies that $\text{LCP}(q, A)$ has a solution. We say that an LCP is *processable by Lemke's algorithm* (see [4] for a detailed discussion on Lemke's algorithm) if Lemke's algorithm can either compute a solution to it if one exists or show that no solution exists.

A matrix $A \in R^{n \times n}$ is called a GPSBD matrix [7] if there exist nonnegative multipliers s_i, t_i with $s_i + t_i = 1, i = 1, 2, \dots, n$, such that

$$(2.1) \quad \forall z \in R^n, \quad z^t Az < 0 \Rightarrow \begin{cases} \text{either} & -s_i z_i + t_i (A^t z)_i \geq 0 \forall i, \\ \text{or} & -s_i z_i + t_i (A^t z)_i \leq 0 \forall i. \end{cases}$$

Let S and T be two nonnegative diagonal matrices with diagonal elements s_i, t_i , where $s_i + t_i = 1$ for $i = 1, \dots, n$. A matrix $A \in R^{n \times n}$ is said to be GPSBD if there exist two nonnegative diagonal matrices S and T with $S + T = I$ such that

$$(2.2) \quad \forall z \in R^n, \quad z^t Az < 0, \Rightarrow \begin{cases} \text{either} & -Sz + TA^t z \geq 0, \\ \text{or} & -Sz + TA^t z \leq 0. \end{cases}$$

Note that GPSBD reduces to PSBD if $S = 0$. A is called *nondegenerate* GPSBD if for all $z \in R^n, z^t Az < 0$ implies $-Sz + TA^t z \neq 0$ and unsigned, i.e., at least one of the inequalities in (2.2) should hold as a strict inequality. A is said to be a MGPSBD matrix if A is a GPSBD matrix but not a PSBD matrix.

The following is a nontrivial example of a GPSBD matrix.

EXAMPLE 2.1. Let

$$A = \begin{bmatrix} 0 & 2 & 0 \\ -1 & 0 & -1 \\ 0 & 1 & 0 \end{bmatrix}.$$

Note that $\nu_-(A + A^t) = 1$. Then for any $z = [z_1 \ z_2 \ z_3]^t$, $z^t Az = z_1 z_2 < 0$ implies z_1 and z_2 are of opposite sign. Clearly, $A^t z = [-z_2 \ 2z_1 + z_3 \ -z_2]^t$ and for $z = [-1 \ 1 \ 5]^t$, $z^t Az < 0$ does not imply $A^t z$ is unsigned. Therefore, A is not a PSBD matrix. However, with the choice $s_1 = 0, s_2 = 1$, and $s_3 = 0$, it is easy to check that A is a GPSBD matrix.

A careful examination of the definition of GPSBD matrix leads to the following result.

PROPOSITION 2.1. *A is a GPSBD matrix if any one of the following conditions holds:*

- (i) $Z_1 \subseteq Z_2$, where $Z_1 = \{z \mid z^t A z < 0\}$ and $Z_2 = \{z \mid z^t(-S + TA^t)z < 0\}$ and $(-S + AT)$ is PSBD.
- (ii) $t_i = k$, $0 \leq k \leq 1$, $\forall i$ and $(-S + AT)$ is PSBD.
- (iii) $\bar{Z} = \{z \mid z^t A z < 0\} \subseteq R_+^n \cup -R_+^n$.
- (iv) $a_{ii} \geq 0 \forall i$ and there exists an $i = i_0$ such that $A_{i_0} \geq 0$ and $a_{ij} = 0 \forall i, j$, $i \neq i_0$, $j \neq i$.
- (v) $a_{ii} \geq 0 \forall i$ and there exists a $j = j_0$ such that $A_{j_0} \neq 0$ and $a_{ij} = 0 \forall i, j$, $i \neq j$, $j \neq j_0$.

Proof. (i) This part follows from the fact that $z^t A z < 0 \Rightarrow z^t(-S + TA^t)z < 0$ and $(-S + AT)$ is PSBD.

(ii) When $t_i = k \forall i$, then $Z_1 \subseteq Z_2$.

(iii) The conclusion follows if we take $s_i = 1 \forall i$.

(iv) and (v) Choose $t_i = \frac{1}{1+a_{ii}}$. $\square$

The following characterization by Murthy and Parthasarathy [13] on nonnegative Q_0 -matrices will be used in the next section.

THEOREM 2.1 (see [13]). *Assume A is nonnegative, where $n \geq 2$. $A \in Q_0$ if and only if for every $i \in \{1, 2, \dots, n\}$, $A_i \neq 0 \Rightarrow a_{ii} > 0$.*

3. Main results on GPSBD matrices.

THEOREM 3.1. *Suppose A is a GPSBD matrix. Then $A_{\alpha\alpha} \in \text{GPSBD}$, where $\alpha \subseteq \{1, \dots, n\}$.*

Proof. Let $x_\alpha \in R^{|\alpha|}$ and

$$A = \begin{bmatrix} A_{\alpha\alpha} & A_{\alpha\bar{\alpha}} \\ A_{\bar{\alpha}\alpha} & A_{\bar{\alpha}\bar{\alpha}} \end{bmatrix},$$

where $\alpha \subseteq \{1, \dots, n\}$.

Suppose that $x_\alpha^t A_{\alpha\alpha} x_\alpha < 0$. Now define $z \in R^n$ by taking $z_\alpha = x_\alpha$ and $z_{\bar{\alpha}} = 0$. Then $z^t A z = x_\alpha^t A_{\alpha\alpha} x_\alpha$. Since A is a GPSBD matrix, $z^t A z = x_\alpha^t A_{\alpha\alpha} x_\alpha < 0 \Rightarrow$ either $(-S + TA^t)z \geq 0$, which implies that $(-S_{\alpha\alpha} + T_{\alpha\alpha} A_{\alpha\alpha}^t) x_\alpha \geq 0$, or $(-S + TA^t)z \leq 0$, which implies $(-S_{\alpha\alpha} + T_{\alpha\alpha} A_{\alpha\alpha}^t) x_\alpha \leq 0$. Hence $A_{\alpha\alpha} \in \text{GPSBD}$. $\square$

This theorem says that GPSBD is a complete class in the sense of [4, 3.9.5]. Crouzeix et al. [6, Theorem 2.1] observed that if A is a PSBD matrix with $\text{rank}(A) \geq 2$, then A^t is also PSBD. However, this is not true for GPSBD matrices in general. This is shown in the following example.

EXAMPLE 3.1. *Let*

$$A = \begin{bmatrix} 1 & -1 & 0 \\ -1 & 1 & 0 \\ 10 & 0 & 1 \end{bmatrix}.$$

Take $z = [1 \ \frac{5}{2} \ -1]^t$. Clearly, A and A^t are not PSBD. Now with the choice $s_1 = 1$, $s_2 = \frac{1}{2}$, $s_3 = \frac{1}{2}$, it is easy to see that A is a GPSBD matrix. Now consider A^t . Note that for $z = [1 \ \frac{5}{2} \ -1]^t$,

$$(-S + TA)z = \begin{bmatrix} (1 - 2s_1)z_1 - (1 - s_1)z_2 \\ -(1 - s_2)z_1 + (1 - 2s_2)z_2 \\ 10(1 - s_3)z_1 + (1 - 2s_3)z_3 \end{bmatrix} = \begin{bmatrix} \frac{1}{2}s_1 - \frac{3}{2} \\ -4s_2 + \frac{3}{2} \\ -8s_3 + 9 \end{bmatrix}.$$

It is easy to see that no s_1, s_2, s_3 exists for which A^t satisfies the definition of a GPSBD matrix.

The following theorem is observed in [7]. For the sake of completeness, we include the proof here.

THEOREM 3.2. *Suppose A is a nondegenerate MGPSBD matrix. Then $\nu_-(A + A^t) = 1$.*

Proof. Let $B = A + A^t$. Note that B has at least one negative eigenvalue since A is not PSD. Suppose for contradiction there exist $\lambda_1, \lambda_2, z_1, z_2$ so that

$$Bz_1 = 2\lambda_1 z_1, \quad Bz_2 = 2\lambda_2 z_2, \quad \|z_1\|^2 = \|z_2\|^2 = 1,$$

$$\lambda_1 \leq \lambda_2 < 0 \quad \text{and} \quad z_1^t z_2 = 0.$$

It is easy to see that $z_1^t A z_1 < 0$ and $z_2^t A z_2 < 0$. Without loss of generality, assume that $\tilde{A}^t z_1 \leq 0$ and $\tilde{A}^t z_2 \geq 0$, where $\tilde{A}^t = -S + T A^t$.

Now for $\mu \in [0, 1]$, we define $z(\mu) = \mu z_1 + (1 - \mu) z_2$. Then $z(\mu)^t B z(\mu) = 2\mu^2 \lambda_1 + 2(1 - \mu)^2 \lambda_2 < 0$. Since A is a nondegenerate GPSBD matrix, it follows that $0 \neq \tilde{A}^t z(\mu) = \mu \tilde{A}^t z_1 + (1 - \mu) \tilde{A}^t z_2 \in R_+^n \cup -R_+^n$. Since $\tilde{A}^t z(0) \geq 0$ and $\tilde{A}^t z(1) \leq 0$, it follows that there exists a $\bar{\mu} \in (0, 1)$ such that $\tilde{A}^t z(\bar{\mu}) = 0$, which contradicts the nondegeneracy assumption. $\square$

THEOREM 3.3. *Suppose A is a nondegenerate MGPSBD matrix where all the diagonal entries of T are positive. Then $(A + A^t)z = 0 \Rightarrow Az = A^t z = 0$.*

Proof. Let $B = A + A^t$. Suppose there exist λ_1, z_1 so that $Bz_1 = 2\lambda_1 z_1$, $\|z_1\|^2 = 1$, where $\lambda_1 < 0$. Let z_0 be such that $Bz_0 = 0$. For $\mu \in R$, define $z(\mu) = z_1 + \mu z_0$. Then it is easy to see that $z(\mu)^t B z(\mu) = 2\lambda_1 < 0$. Without loss of generality, assume that $\tilde{A}^t z_1 \leq 0$, where $\tilde{A}^t = -S + T A^t$. Since A is a nondegenerate MGPSBD matrix, $\tilde{A}^t z(\mu) \neq 0$. Now for all $\mu \in R$, $0 \neq \tilde{A}^t z(\mu) = \tilde{A}^t z_1 - \mu S z_0 + \mu T A^t z_0 \in R_+^n \cup -R_+^n$. Since $0 \neq \tilde{A}^t z(\mu)$ is unsigned, the terms containing the coefficient μ should vanish. Therefore, $A^t z_0 = 0$. Since $Bz_0 = 0$, it follows that $Az_0 = 0$. $\square$

THEOREM 3.4. *Suppose $A \in \text{GPSBD} \cap C_0$. Then $A \in P_0$.*

Proof. To show that $A \in P_0$, it is enough to show that $A^t \in P_0$. In view of a result of Fiedler and Pták [9, Theorem 1.3] it is enough to show that for any nonzero z , $\max_{z_i \neq 0} [z_i (A^t z)_i] \geq 0$. Given $z \in R^n$, let $I_1 = \{i : z_i > 0\}$ and $I_2 = \{i : z_i < 0\}$. We need to consider three cases.

Case I. $I_2 = \emptyset$. Then $z^t A^t z = z^t A z \geq 0$ as $A \in C_0$. Hence $\max_i [z_i (A^t z)_i] \geq 0$.

Case II. $I_1 = \emptyset$. Then $(-z)^t A^t (-z) = z^t A z \geq 0$ as $A \in C_0$. Hence $\max_i [z_i (A^t z)_i] \geq 0$.

Case III. $I_1 \neq \emptyset$ and $I_2 \neq \emptyset$. Suppose that $\max_i [z_i (A^t z)_i] < 0$. Now $z^t A z = z^t A^t z = \sum_i [z_i (A^t z)_i] < 0$. This implies $-s_i z_i + t_i (A^t z)_i \geq 0 \forall i$ or $-s_i z_i + t_i (A^t z)_i \leq 0 \forall i$. Suppose $-s_i z_i + t_i (A^t z)_i \geq 0 \forall i$. Then for all $i \in I_1$, $-s_i z_i^2 + t_i [z_i (A^t z)_i] \geq 0$. But since $\max_i [z_i (A^t z)_i] < 0$, we get $-s_i z_i^2 + t_i [z_i (A^t z)_i] < 0 \forall i \in I_1$. This leads to a contradiction. Therefore, $\max_i [z_i (A^t z)_i] \geq 0$. Similarly, when $-s_i z_i + t_i (A^t z)_i \leq 0 \forall i \in I_2$, then consider $-s_i z_i^2 + t_i [z_i (A^t z)_i] \geq 0$. But since $\max_i [z_i (A^t z)_i] < 0$, we get $-s_i z_i^2 + t_i [z_i (A^t z)_i] < 0 \forall i \in I_2$. This leads to a contradiction. Therefore, $\max_i [z_i (A^t z)_i] \geq 0$. This completes the proof. $\square$

In [11], Mohan, Neogy, and Das have shown that if A is copositive PSBD matrix of rank ≥ 2 , then A is sufficient. However, the following example shows that a GPSBD matrix need not be sufficient.

EXAMPLE 3.2. *Consider the GPSBD matrix A in Example 3.1. Note that A is not PSBD. It is easy to check that A is a row sufficient matrix. However, A is not a column sufficient matrix.*

We prove the following result on row sufficiency.

THEOREM 3.5. *Suppose $A \in \text{MGPSBD} \cap C_0$ with $0 < t_i < 1 \forall i$. Then A is a row sufficient matrix.*

Proof. Suppose $z_i(A^t z)_i \leq 0$ for $i = 1, \dots, n$. Let $I_1 = \{i : z_i > 0\}$ and $I_2 = \{i : z_i < 0\}$. We need to consider three cases.

Case I. $I_2 = \emptyset$. Then $z^t A z = z^t A^t z = \sum_i z_i (A^t z)_i \leq 0$. Since $A \in C_0$, $[z_i (A^t z)_i] = 0 \forall i$.

Case II. $I_1 = \emptyset$. Then $(-z)^t A^t (-z) = z^t A^t z = \sum_i z_i (A^t z)_i \leq 0$. Since $A \in C_0$, $[z_i (A^t z)_i] = 0 \forall i$.

Case III. Suppose there exists a vector z such that $z_i (A^t z)_i \leq 0$ for $i = 1, 2, \dots, n$ and $z_k (A^t z)_k < 0$ for at least one $k \in \{1, 2, \dots, n\}$. Let $I_1 \neq \emptyset$ and $I_2 \neq \emptyset$. $z^t A z = z^t A^t z = \sum_i [z_i (A^t z)_i] < 0$. This implies $-s_i z_i + t_i (A^t z)_i \geq 0 \forall i$ or $-s_i z_i + t_i (A^t z)_i \leq 0 \forall i$.

Without loss of generality, assume $-s_i z_i + t_i (A^t z)_i \geq 0 \forall i$. Then for all $i \in I_1$, $-s_i z_i^2 + t_i z_i (A^t z)_i \geq 0$. This implies $[z_i (A^t z)_i] \geq \frac{s_i}{t_i} z_i^2 > 0 \forall i \in I_1$. Therefore, $\sum_{i \in I_1} [z_i (A^t z)_i] > 0$. Since $z_i (A^t z)_i \leq 0$ for $i = 1, \dots, n$, this leads to a contradiction. Therefore, $[z_i (A^t z)_i] = 0 \forall i$. So A is row sufficient. $\square$

Note that the assumption in the above theorem $0 < t_i < 1 \forall i$ cannot be relaxed.

EXAMPLE 3.3. *Consider the matrix*

$$A = \begin{bmatrix} 1 & 0 & 0 \\ 2 & 1 & 0 \\ 10 & 0 & 0 \end{bmatrix}.$$

Note that A is copositive but it is not row sufficient.

For $z = [-1 \ -1 \ 1]^t$, $z^t A z < 0$. Clearly A is not PSBD. Now for $z = [-1 \ -1 \ 1]^t$,

$$(-S + T A^t)z = \begin{bmatrix} (1 - 2s_1)z_1 + (1 - s_1)2z_2 + (1 - s_1)10z_3 \\ (1 - 2s_2)z_2 \\ -s_3 z_3 \end{bmatrix} = \begin{bmatrix} 7 - 6s_1 \\ 2s_2 - 1 \\ -s_3 \end{bmatrix}.$$

It is easy to see that no s_1, s_2, s_3 exists where $0 < s_i < 1 \forall i$, i.e., no t_1, t_2, t_3 exists where $0 < t_i < 1 \forall i$ for which the definition of GPSBD matrix is satisfied. However, with the choice $s_1 = \frac{1}{2}, s_2 = \frac{1}{2}, s_3 = 0$, A is a MGPSBD matrix.

The following result is a consequence of the characterization of row sufficient matrices observed by Cottle, Pang, and Venkateswari [5].

THEOREM 3.6. *Suppose $A \in \text{MGPSBD} \cap C_0$ with $0 < t_i < 1 \forall i$. For each vector $q \in R^n$, if (z^*, u^*) is a KKT pair of the quadratic program $QP(q, A) : [\min z^t (Az + q); z \geq 0, Az + q \geq 0]$, then z^* solves $LCP(q, A) : [z \geq 0, Az + q \geq 0, z^t (Az + q) = 0]$.*

Proof. From Theorem 3.5 and [5, Theorem 4, p. 238], the result follows. $\square$

Remark 3.1. The above example and Theorem 3.6 show that the statement $S''(q, A) \subseteq S(q, A)$ in [7] holds for GPSBD matrices only with the additional assumptions stated in Theorem 3.5.

THEOREM 3.7. *Assume A is a nonnegative merely generalized positive subdefinite matrix with $0 < t_i < 1 \forall i$. Then $A \in C_0^*$.*

Proof. By Theorem 3.5, A is row sufficient. Therefore, $A \in Q_0$ [5, p. 239]. Now by Theorem 2.1, for any nonnegative Q_0 -matrix $A_i \neq 0 \Rightarrow a_{ii} > 0$. Let $\alpha = \{i \mid a_{ii} > 0\}$. Then $x^t A x = 0$ for $x_\alpha = 0$. Now it is easy to check that for $(0, x_{\bar{\alpha}})$, $Ax \geq 0$ implies $A^t x \leq 0$. Thus $A \in C_0^*$. $\square$

From Theorem 3.5, it follows that $LCP(q, A)$ where A is a copositive MGPSBD matrix with $0 < t_i < 1$ is processable by Lemke's algorithm.

The following theorem extends the result of Evers [8] and the result obtained in [11] for solving $LCP(q, A)$ by Lemke's algorithm when A satisfies certain conditions stated in the following theorem. The proof follows along the similar lines of the proof given in [11].

THEOREM 3.8. *Suppose $A \in R^{n \times n}$ can be written as $M + N$, where $M \in MGPSBD \cap C_0^+$, is nondegenerate with $0 < t_i < 1 \forall i$, and $N \in C_0$. If the system $q + Mx - N^t y \geq 0$, $y \geq 0$, is feasible, then Lemke's algorithm for $LCP(q, A)$ with covering vector $d > 0$ terminates with a solution.*

Proof. Suppose the feasibility condition of the theorem holds so that there exist an $x^* \in R^n$ and a $y^* \in R_+^n$ such that $q + Mx^* - N^t y^* \geq 0$. First we shall prove that for any $x \geq 0$, if $Ax \geq 0$ and $x^t Ax = 0$, then $x^t q \geq 0$. Note that for given $x \geq 0$, $x^t Ax = 0 \Rightarrow x^t(M + N)x = 0$ and since $M, N \in C_0$, this implies that $x^t Mx = 0$. As M is a nondegenerate MGPSBD matrix, by Theorem 3.3 we get $x^t Mx = 0 \Rightarrow x^t(M + M^t)x = 0 \Rightarrow (M + M^t)x = 0 \Rightarrow M^t x = 0 \Rightarrow Mx = 0$. Also, since $Ax \geq 0$, it follows that $Nx \geq 0$ and hence $x^t N^t y^* \geq 0$. Further, since $q + Mx^* - N^t y^* \geq 0$ and $x \geq 0$, it follows that $x^t(q + Mx^* - N^t y^*) \geq 0$. This implies that $x^t q \geq x^t N^t y^* \geq 0$.

Now from Corollary 4.4.12 and Theorem 4.4.13 of [4, p. 277], it follows that Lemke's algorithm for $LCP(q, A)$ with covering vector $d > 0$ terminates with a solution. $\square$

The following example demonstrates that the class $MGPSBD \cap C_0^+$ is nonempty.

EXAMPLE 3.4. *Consider the copositive-plus matrix*

$$A = \begin{bmatrix} 1 & 0 & 0 \\ 2 & 1 & 0 \\ 8 & 0 & 1 \end{bmatrix}.$$

Take $z = [-1 \ -1 \ 1]^t$. It is easy to check that A is not MPSBD. However, with choice $s_i = \frac{1}{2} \forall i$, A is a MGPSBD matrix.

Acknowledgments. The authors wish to thank the unknown referees and S. R. Mohan, Indian Statistical Institute, Delhi Centre, who have patiently gone through this paper and whose suggestions have improved its presentation and readability considerably.

REFERENCES

- [1] A. BERMAN AND R. J. PLEMMONS, *Nonnegative Matrices in the Mathematical Sciences*, Academic Press, New York, 1979.
- [2] R. W. COTTLE AND J. A. FERLAND, *Matrix-theoretic criteria for the quasiconvexity and pseudoconvexity of quadratic functions*, Linear Algebra Appl., 5 (1972), pp. 123–136.
- [3] R. W. COTTLE AND S.-M. GUU, *Two characterizations of sufficient matrices*, Linear Algebra Appl., 170 (1992), pp. 56–74.
- [4] R. W. COTTLE, J. S. PANG, AND R. E. STONE, *The Linear Complementarity Problem*, Academic Press, New York, 1992.
- [5] R. W. COTTLE, J. S. PANG, AND V. VENKATESWARAN, *Sufficient matrices and the linear complementarity problem*, Linear Algebra Appl., 114/115 (1989), pp. 231–249.
- [6] J.-P. CROUZEIX, A. HASSOUNI, A. LAHLOU, AND S. SCHAIBLE, *Positive subdefinite matrices, generalized monotonicity and linear complementarity problems*, SIAM J. Matrix Anal. Appl., 22 (2000), pp. 66–85.
- [7] J.-P. CROUZEIX AND S. KOMLÓSI, *The linear complementarity problem and the class of generalized positive subdefinite matrices*, in Optimization Theory (Matrahaza, 1999), Appl. Optim. 59, Kluwer, Dordrecht, The Netherlands, 2001, pp. 45–63.

- [8] J. J. M. EVERS, *More with the Lemke complementarity algorithm*, Math. Programming, 15 (1978), pp. 214–219.
- [9] M. FIEDLER AND V. PTÁK, *Some generalizations of positive definiteness and monotonicity*, Numer. Math., 9 (1966), pp. 163–172.
- [10] B. MARTOS, *Subdefinite matrices and quadratic forms*, SIAM J. Appl. Math., 17 (1969), pp. 1215–1223.
- [11] S. R. MOHAN, S. K. NEOGY, AND A. K. DAS, *More on positive subdefinite matrices and the linear complementarity problem*, Linear Algebra Appl., 338 (2001), pp. 275–285.
- [12] K. G. MURTY, *Linear Complementarity, Linear and Nonlinear Programming*, Heldermann Verlag, West Berlin, 1988.
- [13] G. S. R. MURTHY AND T. PARTHASARATHY, *Some properties of fully semimonotone matrices*, SIAM J. Matrix Anal. Appl., 16 (1995), pp. 1268–1286.
- [14] H. NIKAIDO, *Convex Structures and Economic Theory*, Academic Press, New York, 1968.
- [15] A. W. TUCKER, *Principal pivotal transforms of square matrices*, SIAM Rev., 5 (1963), pp. 305–305.
- [16] H. VÄLIAHO, *Criteria for sufficient matrices*, Linear Algebra Appl., 233 (1996), pp. 109–129.

SPECIAL SECTION ON PRECONDITIONING TECHNIQUES

At the Third International Conference on Preconditioning Techniques for Large Sparse Matrix Problems in Scientific and Industrial Applications, held in Napa, California, on October 27–29, 2003, Henk A. van der Vorst, Editor-in-Chief of the *SIAM Journal on Matrix Analysis and Applications* (SIMAX), expressed an interest in publishing in SIMAX a collection of articles that would focus on current research on preconditioning techniques for matrix computation. This special section on preconditioning techniques is the result of subsequent discussions with SIAM and SIMAX. Manuscripts were solicited through various channels, particularly from the speakers at the third preconditioning conference.

The term *preconditioning* generally refers to the additional step designed to improve the performance of a matrix algorithm, such as the rate of convergence of a Krylov method. Preconditioning arises in a variety of matrix computation. However, the papers in this special section provide a snapshot of the ongoing research activities on preconditioning techniques specifically for the solution of systems of linear equations. Multilevel and multigrid preconditioning techniques are the focus of the papers by Notay; de Sterk, Yang, and Heys; and Vassilevski and Zikatanov. Bank, Wan, and Qu have considered correction and interpolation techniques for improving the convergence of a multigrid method for solving convection-diffusion equations. Bridson and Greif have used a combination of multiple preconditioners in an attempt to improve the convergence of an iterative method. Giraud and Gratton have employed first-order perturbation theory to derive the first-order perturbations on the eigenvalues of matrices preconditioned by some forms of spectral preconditioners, and thereby obtained some insight into the design of such preconditioners. Lee, Raghavan, and Ng have investigated the problem of improving the quality of incomplete factorization through row and column reordering techniques.

Some of papers are problem specific. The paper by Benzi and Ng considers preconditioning techniques for Toeplitz least squares problems, while that by Botchev and Golub focuses on preconditioning techniques for saddle point problems. Le Borne and Grasedyck have employed sparse hierarchical matrix approximations as preconditioners in iterative methods for solving convection-dominated elliptic partial differential equations. Bardsley and Nagy have investigated preconditioning strategies for solving ill-posed inverse problems arising from image analysis.

The editor-in-chief of this special section would like to thank Henk A. van der Vorst for the encouragement, as well as Mitch Chernoff and his staff in the SIAM office for the advice and support in putting the special section together. Michele Benzi, Edmond Chow, Yousef Saad, Valeria Simoncini, and Wei-Pai Tang served as guest editors; their effort was much appreciated. Last but not the least, the editor-in-chief would like to thank all the authors for submitting their manuscripts to the special section, and all the anonymous referees for their help in reviewing the manuscripts.

Esmond G. Ng
Guest editor-in-chief

AGGREGATION-BASED ALGEBRAIC MULTILEVEL PRECONDITIONING*

YVAN NOTAY†

Abstract. We propose a preconditioning technique that is applicable in a “black box” fashion to linear systems arising from second order scalar elliptic PDEs discretized by finite differences or finite elements with nodal basis functions. This technique is based on an algebraic multilevel scheme with coarsening by aggregation. We introduce a new aggregation method which, for the targeted class of applications, produces semicoarsening effects whenever desirable, while the number of nodes is decreased by a factor of about 4 at each level, regardless of the problem at hand. Moreover, the number of nonzero entries per row in the successive coarse grid matrices remains approximately constant, ensuring small set up cost and modest memory requirements.

This aggregation technique can be used in an algebraic multigrid (AMG)-like framework, but better results are obtained with an algebraic multilevel scheme based on a block approximate factorization of the matrix. In this scheme, the block pivot corresponding to fine grid nodes is approximated by a modified incomplete LU (MILU) factorization. To enhance robustness and avoid any potential breakdown, the coarsening process is refined by recasting as “coarse” fine grid nodes for which the corresponding pivot in this MILU factorization would be negative or too small.

Numerical results display the efficiency, the scalability, and the robustness of the resulting preconditioner on a wide set of discrete scalar PDE problems, ranging from the two-dimensional Poisson equation to three-dimensional convection-diffusion problems with high Reynolds number and strongly varying convection.

Key words. algebraic multigrid, multilevel, aggregation, preconditioning

AMS subject classification. 65F15

DOI. 10.1137/04061129X

1. Introduction. We consider large sparse, possibly nonsymmetric, linear systems

$$(1.1) \quad A\mathbf{u} = \mathbf{b}$$

arising from the discretization of second order scalar elliptic PDEs. To solve such systems, algebraic multigrid (AMG) and algebraic multilevel methods are now quite popular [5, 6, 7, 8, 9, 10, 11, 12, 17, 19, 24, 26, 30, 34]. They follow the paradigm of classical “geometric” multigrid or multilevel methods (e.g., [2, 3, 22, 16, 33]) but often do not require any information besides the matrix structure and the matrix entries. They can thus be applied to problems without any particular structure, and, somewhat sacrificing efficiency on “easy” cases, they aim at achieving robustness on a large class of problems without need for any specific tuning.

Most of these methods are based on the recursive use of a two-level scheme, and basically the same two ingredients enter the definition of these two-level schemes.

The first of these ingredients is a partitioning of the unknowns (or nodes) into fine grid nodes and coarse grid nodes, according to which the system matrix is permuted

*Received by the editors July 8, 2004; accepted for publication (in revised form) by E. Chow May 6, 2005; published electronically March 17, 2006. This research was supported by the “Fonds National de la Recherche Scientifique,” Maître de recherches.

<http://www.siam.org/journals/simax/27-4/61129.html>

†Service de Métrologie Nucléaire (C.P. 165/84), Université Libre de Bruxelles, 50, Av. F.D. Roosevelt, B-1050 Brussels, Belgium (ynotay@ulb.ac.be).

(in general only implicitly) and written in the 2×2 block form

$$(1.2) \quad A = \begin{pmatrix} A_{FF} & A_{FC} \\ A_{CF} & A_{CC} \end{pmatrix}.$$

Here and in the following, F corresponds to the F set, i.e., the set of fine grid nodes, and C to the C set, i.e., the set of coarse grid nodes.

The second ingredient is an “interpolation” matrix J_{FC} that interpolates on the F set a vector defined on the C set. With this matrix, one associates the prolongation

$$(1.3) \quad p = \begin{pmatrix} J_{FC} \\ I \end{pmatrix}$$

and the Galerkin coarse grid matrix

$$(1.4) \quad \hat{A}_C = p^T A p = A_{CC} + A_{CF} J_{FC} + J_{FC}^T A_{FC} + J_{FC}^T A_{FF} J_{FC}.$$

There are several ways to define these ingredients. Among them, aggregation-based schemes play a particular role. On the one hand, they are appealing because they are simple and cheap. Indeed, each fine grid node is then associated with a unique coarse grid node; that is, J_{FC} has only one nonzero entry per row. It is therefore as sparse as possible, as is the associated coarse grid matrix $\hat{A}_C$. Consequently, resources needed to compute and store the preconditioner are guaranteed to be low.

On the other hand, aggregation-based schemes do not in general lead to robust and scalable performances, except when modified in one way or another, in which case they lose their original simplicity [8, 32, 34]. However, this observation is essentially true for schemes that follow the AMG paradigm. In [24], an algebraic multilevel method based on a block approximate factorization process has been found scalable and robust while using coarse grid matrices computed by an aggregation rule.

Consider the exact block factorization of the matrix (1.2):

$$(1.5) \quad A = \begin{pmatrix} I & \\ A_{CF} A_{FF}^{-1} & I \end{pmatrix} \begin{pmatrix} A_{FF} & \\ & S_A \end{pmatrix} \begin{pmatrix} I & A_{FF}^{-1} A_{FC} \\ & I \end{pmatrix},$$

where $S_A = A_{CC} - A_{CF} A_{FF}^{-1} A_{FC}$ is the Schur complement. The latter kind of method, in its basic two-level variant, approximates this factorization with

$$(1.6) \quad B = \begin{pmatrix} I & \\ A_{CF} P_{FF}^{-1} & I \end{pmatrix} \begin{pmatrix} P_{FF} & \\ & S \end{pmatrix} \begin{pmatrix} I & P_{FF}^{-1} A_{FC} \\ & I \end{pmatrix},$$

where P_{FF} is some approximation to A_{FF} , and S some approximation to S_A , for which one may take the (rescaled) coarse grid matrix $\hat{A}_C$.

Here, we present a method based on these principles, introducing two major enhancements to the method from [24].

First, the coarsening process considered in [24] was essentially the one from the standard AMG method [26, 32], with an a posteriori assignment of each fine grid node to a neighboring coarse grid node. This process therefore inherits the drawbacks of that approach: sensitivity to the strong/weak coupling threshold and slow coarsening in case of low connectivity, severe anisotropy, or dominating convection. For instance,

if the (main part of the) matrix originates from a two-dimensional (2D) five point or three-dimensional (3D) nine point stencil, the F/C partitioning is essentially a red-black one, which is not very desirable. Note also that aggressive coarsening, as one may use with standard AMG [32, section A.7.1.2], is not very convenient in the context of an aggregation scheme, since the a posteriori assignment of fine grid nodes is not easy when some of them are not directly connected to any coarse grid node.

Here we develop another approach, in which aggregates are directly formed by grouping the nodes pairwise. Since the resulting coarsening would be too slow, this process is repeated once; that is, we form quadruplets by grouping pairwise the pairs resulting from the first pass; we call this “double pairwise aggregation.” The number of nodes is then decreased by a factor of about 4 at each level, regardless of whether, for instance, the underlying PDE is 2D or 3D, with or without anisotropy. On the other hand, pairs are formed by following the strongest negative coupling. As will be seen below, this makes the process essentially insensitive to the strong/weak coupling threshold.

The second enhancement concerns the approximation P_{FF} to A_{FF} . As in [24], we rely on modified incomplete LU (MILU) factorizations, because they possess all the desirable properties (see below). However, they are not always well defined for general matrices. Most often, the latter fact is harmless because the submatrix A_{FF} resulting from the coarsening process is fairly well conditioned, and (M)ILU factorizations of well-conditioned matrices tend to be breakdown-free. Nevertheless, there is an obvious danger in using this approach without safeguard. Here, we introduce a “dynamic” MILU algorithm that reallocates to the C set the nodes for which the corresponding pivot in the factorization of A_{FF} would be negative or too small (relative to the corresponding diagonal entry in A_{FF}). Note that we are comfortable in doing so since, on account of the fast coarsening resulting from the double pairwise aggregation strategy, there is plenty of room for moving nodes from F to C before one gets in trouble with the coarsening speed or the set-up cost.

To conclude this introduction, we briefly mention some other approaches that can be followed to build a preconditioner based on a two-level block factorization of the form (1.6). First, as shown in [25], basically any (algebraic) multigrid-like scheme can be used to define the F/C partitioning and the coarse grid matrix that approximates the Schur complement. Here we have proposed an aggregation-based scheme because we found it simple, cheap, and nevertheless robust.

Alternatively, the approximate Schur complement can be computed in an ILU-like fashion, that is, formed by incompletely eliminating the fine grid unknowns, using, for instance, some threshold to drop small entries. This idea is followed in, e.g., [28, 30, 31]. In these works, the F/C partitioning is based on finding a (block) independent set in the matrix graph of A , so that A_{FF} is (block) diagonal. Its inverse is therefore sparse, allowing us to compute the approximate Schur complement by applying some dropping rule to S_A . Consequently, the method inherits the versatility and the wide range of applications of ILU-based methods. However, the coarsening is actually much slower than with our approach. Moreover, the number of nonzero entries per row tends to increase in the successive coarse grid matrices, and most often the multilevel reduction has to be stopped before the coarse grid matrix is sufficiently small to be factorized exactly. The resulting method is then likely to be not competitive with multigrid-based schemes on very large PDE problems.

The remainder of this paper is organized as follows. In section 2, we introduce our double pairwise aggregation strategy. Its use in the context of an AMG-like algorithm is briefly discussed in section 3. In section 4, we present the multilevel block

approximate factorization preconditioner with more details, including a complete description and discussion of our dynamic MILU factorization algorithm. The results of numerical experiments are reported in section 5.

2. Double pairwise aggregation.

2.1. Simple pairwise aggregation. There are two ways to define an aggregation algorithm. One way consists of defining an F/C partitioning and next associating each F node with a unique C node, so as to form aggregates. The other way consists of forming these aggregates directly. This is, for instance, the approach followed in [8, 34]. In these works, the coarse level unknowns are treated as purely algebraic entities, and no F or C set is defined explicitly. This is possible, however, only with an AMG-like algorithm. If one wants to use a block two-level preconditioner of the form (1.6), it is mandatory to further select one C node in each aggregate, the other nodes forming the F set.

Our aggregation algorithm given below (Algorithm 2.1) forms aggregates directly by grouping the nodes pairwise. One picks up one unmarked node i at a time, according to some priority rule designed so as to favor a regular covering of the matrix graph (see below). A tentative aggregate is then formed by grouping this node with the unmarked node with which it is most strongly negative coupled, that is, (one of) the unmarked node(s) j for which a_{ij} is minimal. The strong/weak coupling threshold β then acts as a filter. That is, essentially as in AMG [32, p. 473], one defines the set of nodes S_i to which i is strongly negative coupled by

$$S_i = \left\{ j \neq i \mid a_{ij} < -\beta \max_{a_{ik} < 0} |a_{ik}| \right\} ,$$

and the tentative pair is rejected if and only if $j \notin S_i$. In the latter case the node i initially picked up forms a singleton, that is, a C node without any associated F node. Note, however, that this may occur only if all nodes k for which a_{ik} is minimal have already been marked (or do not correspond to a negative coupling because all couplings in row i are positive). For the kind of applications targeted here, this occurs only incidentally, for instance near boundaries. In general, the algorithm progresses smoothly along the direction of strongest negative coupling, and most nodes picked up are effectively associated with (one of) the neighbor(s) with which they are most strongly negative coupled. Hence, the strong/weak coupling threshold has only a slight influence on the coarsening process. This justifies the relatively large size of the proposed default value in Algorithm 2.1, protecting against unsafe associations without dramatic impact on the coarsening speed.

Once a pair has been formed, we always put the node initially picked up in F and its companion in C . This guarantees that each node in F is strongly negative coupled to the associated node in C , whereas the converse is not necessarily true and is even usually not true when the matrix is nonsymmetric.

In Algorithm 2.1, we also include an optional check for strongly diagonally dominant rows, for which the corresponding node is put in F without any companion node in C (that is, the corresponding row in J_{FC} is zero). This guarantees a proper treatment of finite element matrices in which Dirichlet boundary conditions have been imposed by adding a big number to the corresponding diagonal elements. More generally, it is heuristically clear that, when a row is strongly diagonally dominant enough, a fast reduction of the error at the corresponding node can be obtained without multilevel enhancement.

ALGORITHM 2.1 (pairwise aggregation).

Input: $n \times n$ matrix $A = (a_{ij})$;

strong/weak coupling threshold β (default: $\beta = \mathbf{0.75}$);

logical parameter *CheckDD*.

Output: partitioning (F, C) of $[1, n]$;

subset (aggregates) $G_i, i \in C$, such that $G_i \cap C = \{i\}$
and $G_i \cap G_j = \emptyset$ for $i \neq j$.

Definition: for all i : $S_i = \{j \neq i \mid a_{ij} < -\beta \max_{a_{ik} < 0} |a_{ik}|\}$.

Initialization: if (*CheckDD*): $F = \{i \mid a_{ii} > 3 \sum_{j \neq i} |a_{ij}|\}$, otherwise: $F = \emptyset$;

$C = \emptyset$;

$U = [1, n] \setminus F$;

for all i : $m_i = |\{j \in U \mid i \in S_j\}|$.

Algorithm: while $U \neq \emptyset$, do

1. select $i \in U$ with minimal m_i

2. select $j \in U$ such that $a_{ij} = \min_{k \in U} a_{ik}$

3. if $j \in S_i$:

3a. $C = C \cup \{j\}$, $F = F \cup \{i\}$, $G_j = \{i, j\}$, $U = U \setminus \{i, j\}$

3b. update: $m_k = m_k - 1$ for $k \in S_i$, and $m_k = m_k - 1$ for $k \in S_j$

otherwise:

3a'. $C = C \cup \{i\}$, $G_i = \{i\}$, $U = U \setminus \{i\}$

3b'. update: $m_k = m_k - 1$ for $k \in S_i$

Eventually we discuss item 1 of Algorithm 2.1. We give priority to the node(s) i with minimal m_i , where m_i is the number of unmarked nodes that are strongly negative coupled to i (that is, m_i is the number of sets S_j to which i belongs and that correspond to an unmarked node j). As stated above, this favors a regular covering of the matrix graph. Probably other rules could work as well. We consider this one because we observe that in some nonsymmetric examples, at any stage of the process, there is at least one node j for which $m_j = 0$. Then, if this node is not selected, it has many chances to become a singleton, since no unmarked node is strongly negative coupled to it.

As output of the algorithm, we have, besides the F/C partitioning, also the aggregates $G_j, j \in C$, that is, the set of F nodes associated with j , plus j itself. The interpolation matrix J_{FC} may then formally be defined by

$$(2.1) \quad \forall i \in F, j \in C : (J_{FC})_{ij} = \begin{cases} 1 & \text{if } i \in G_j, \\ 0 & \text{otherwise,} \end{cases}$$

whereas the prolongation p satisfies

$$(2.2) \quad \forall i \in [1, n], j \in C : (p)_{ij} = \begin{cases} 1 & \text{if } i \in G_j, \\ 0 & \text{otherwise.} \end{cases}$$

However, there is no need to form these matrices explicitly, and the Galerkin coarse grid matrix (1.4) is more easily computed with

$$(2.3) \quad \forall i, j \in C : (\hat{A}_C)_{ij} = \sum_{k \in G_i} \sum_{\ell \in G_j} a_{k\ell}.$$

That is, the entries in the coarse grid matrix are obtained by summing the entries in A that connect the different aggregates.

2.2. Forming pairs of pairs. Coarsening by simple pairwise aggregation is relatively slow, which does not favor optimal performance of multilevel methods. A faster coarsening can be obtained by repeating the process, defining aggregates by forming pairs of pairs—more precisely, by forming pairs of aggregates from the first pass, some of which are singletons.

Forming pairs of pairs has already been considered in [8], where a strategy is proposed that is based on the connectivity pattern in the resulting aggregates. This favors a coarsening that mimics the geometric coarsening well. However, as is well known, geometric coarsening is not desirable in all cases, and good algebraic coarsening algorithms should, for instance, automatically provide a semicoarsening like partitioning in the presence of strong anisotropy or dominating convection.

We therefore propose another strategy, based on the repeated use of our pairwise aggregation algorithm. That is, we compute the coarse grid matrix resulting from a first application of Algorithm 2.1, and apply this algorithm again to the latter matrix, forming pairs of C nodes from the first pass, which implicitly define pairs of pairs. Details are given in Algorithm 2.2 below. For the class of applications considered in this paper, the resulting aggregates are mostly quadruplets, with some triplets, pairs, and singletons left. Consequently, the coarsening ratio n/n_c (where $n_c = |C|$ is the number of C nodes) is slightly less than 4, regardless of the problem at hand.

On 2D model problems, the procedure mimics geometric coarsening well if the coefficients are isotropic, whereas strong anisotropy induces a nonstandard semicoarsening $h \rightarrow 4h$ in the direction of strong coupling. We apply the same procedure in 3D cases as well. On model anisotropic problems, this induces a standard planewise semicoarsening or an $h \rightarrow 4h$ linewise semicoarsening, according to the case at hand. It is more difficult to figure out what comes out in isotropic cases, but the numerical experiments show that the approach is robust (see section 5 below).

ALGORITHM 2.2 (double pairwise aggregation).

Input: $n \times n$ matrix $A = (a_{ij})$;
strong/weak coupling threshold β (default: $\beta = \mathbf{0.75}$);

Output: partitioning (F, C) of $[1, n]$;
subset (aggregates) $G_i, i \in C$, such that $G_i \cap C = \{i\}$
and $G_i \cap G_j = \emptyset$ for $i \neq j$.

Algorithm:

1. Apply Algorithm 2.1 to A with threshold β and $CheckDD = True$.
Output: (F_1, C_1) and $G_i^{(1)}, i \in C_1$.
2. Compute the auxiliary matrix $A_1 = (a_{ij}^{(1)}), i, j \in C_1$, with

$$a_{ij}^{(1)} = \sum_{k \in G_i^{(1)}} \sum_{\ell \in G_j^{(1)}} a_{k\ell}.$$

3. Apply Algorithm 2.1 to A_1 with threshold β and $CheckDD = False$.
Output: (F_2, C_2) and $G_i^{(2)}, i \in C_2$.
4. $C = C_2, F = F_1 \cup F_2, G_i = \cup_{j \in G_i^{(2)}} G_j^{(1)}, i \in C$.

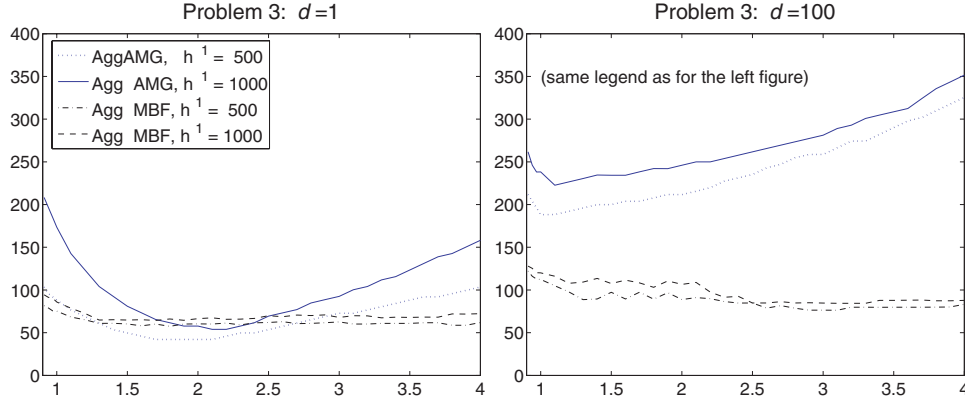


FIG. 1. *Relative solution cost (total number of flops divided by the number of flops for 1 unpreconditioned CG iteration) as a function of the inverse of the scaling parameter.*

3. Aggregation-based AMG. With the ingredients defined in the preceding section, it is not difficult to set up an AMG scheme. One needs only to choose a smoother and a cycling strategy. Since this approach is not central in our work, we refer to, e.g., [32] for more details.

Indeed, the combined use of AMG with simple aggregation is not advised in general, as explained in some detail in [32, section A.9]. To summarize in a few words, consider a model 2D example for which our strategy mimics well the geometric coarsening. Despite this similarity, two major differences appear between a geometric and an aggregation-based multigrid scheme. Firstly, the interpolation defined by (2.1) is zero order accurate, whereas standard geometric multigrid uses (at least) first order accurate interpolation, as required by the theory [16, sections 3.5 and 6.3.2]. Next, the coarse grid matrices defined by (2.3) resemble their geometric counterparts but appear to be badly scaled.

This latter point is best illustrated by the following numerical experiment. We consider the AMG scheme resulting from the algorithms given in section 2, except that the coarse grid matrices are rescaled. We use V-cycle and symmetric Gauss–Seidel smoothing with one pre- and one postsmoothing step at each level; the coarse grid matrix at the coarsest level is factorized exactly, and the number of levels is fixed as indicated in section 4.3; the resulting AMG scheme is used as a preconditioner for the conjugate gradient (CG) method. We consider Problem 3 that is fully described in section 5. This is a five point approximation of a 2D Laplace-like equation, with some Neumann boundary conditions and anisotropies and discontinuities of order d , where d is a parameter; for $d = 1$, we thus have a model isotropic constant coefficient problem.

Figure 1 displays the relative solution cost against the inverse of the scaling parameter, that is, the number by which all coefficients in the successive coarse grid matrices have been divided. One sees that for the model problem ($d = 1$), dividing all coefficients by 2 delivers near-optimal results and scalable performances. This was already observed in [8]. However, performances deteriorate as d increases, and it seems better not to scale the coarse grid matrices then. Generally speaking, results appear sensitive to the scaling parameter, and it is not obvious how to find a rule that could be applied uniformly.

Figure 1 also displays the results obtained with our multilevel block approximate factorization (MBF) preconditioner, considering the method as described in the next section, except that all coarse grid matrices are uniformly scaled according to the indicated parameter (instead of the more sophisticated default rule (4.4) that is justified below). On the model problem, this approach is roughly equivalent to aggregation-based AMG with appropriate scaling. But it appears fairly insensitive to the scaling parameter and much more robust with respect to anisotropies and discontinuities in the coefficients of the underlying PDE.

4. Aggregation-based multilevel block factorization. As already stated, this approach consists of using as a building block a two-level preconditioner of the form (1.6) with

$$(4.1) \quad S = \alpha \hat{A}_C,$$

where $\hat{A}_C$ is the coarse grid matrix computed from (2.3) and α some scaling factor.

There are several ways to explain why this may be efficient despite the mitigated results obtained with AMG. First, applying AMG requires effectively using the interpolation (2.1) to transfer information from the coarse grid to the fine grid. With the block approximate factorization approach, the interpolation is used as only a tool to build the coarse grid matrix, and plays otherwise no role in the algorithm. Hence, it does not matter if this interpolation resembles a zero order one, as long as the resulting coarse grid matrix mimics well those obtained with more sophisticated schemes. More generally, one may observe that if the F/C partitioning resembles a geometric one and if the coarse grid matrix mimics well a discretization on the coarse grid, then the multilevel block approximate factorization preconditioner is close to its “geometric” counterpart as defined, for instance, in [22], where optimal convergence properties are proved for the symmetric case (see also [23] for the nonsymmetric case).

Next, as we saw in the preceding section, the main problem with aggregation-based coarse grid matrices is their bad scaling. We also saw on an example that this issue is much less critical with multilevel block factorization preconditioning. To understand why, observe that the eigenvalues of $B^{-1}A$ with B defined by (1.6) are the same as those of $\tilde{B}^{-1}\tilde{A}$, where

$$\tilde{A} = \begin{pmatrix} I & \\ -A_{CF} P_{FF}^{-1} & I \end{pmatrix} \begin{pmatrix} A_{FF} & A_{FC} \\ A_{CF} & A_{CC} \end{pmatrix} \begin{pmatrix} I & -P_{FF}^{-1} A_{FC} \\ & I \end{pmatrix}$$

and

$$\tilde{B} = \begin{pmatrix} P_{FF} & \\ & S \end{pmatrix}.$$

The scaling of S thus plays exactly the same role as the scaling of one of the diagonal blocks in a 2×2 block-diagonal preconditioning. From a theoretical point of view, this scaling is near-optimal when, considering both matrices $P_{FF}^{-1}\tilde{A}_{FF}$ and $S^{-1}\tilde{A}_{CC}$, the spectrum of the better conditioned one is contained in the spectrum of the worse conditioned one. In practice, one does not have to expect large variations in the performances as long as this condition approximately holds. This explains the observations from Figure 1: on the one hand, it is worthwhile to rescale the coarse grid matrices, but, on the other hand, results are about the same with any scaling factor whose inverse lies between 2.5 and 3.5.

Now, to complete the description of our multilevel block factorization preconditioner, we need to consider the following points: (1) the choice of the approximation P_{FF} to A_{FF} and the associated refinement of the F/C partitioning; (2) the scaling factor α for which a rule has still to be provided; (3) the cycling strategy used to turn the two-level preconditioner based on (1.6) into a multilevel scheme. These points are discussed in the next three subsections.

4.1. Definition of P_{FF} and refinement of the F/C partitioning. In general, A_{FF} as resulting from our coarsening algorithm is fairly well conditioned, and it is therefore not difficult to set up close easy-to-invert approximations P_{FF} . However, from the theoretical developments and numerical results in [20, 25], one knows that P_{FF} may *not* be any approximation. Roughly speaking, it has to be such that

$$(4.2) \quad \tilde{J}_{FC} = -P_{FF}^{-1} A_{FC}$$

also defines a “correct” interpolation for (algebraically) smooth vectors. Without going into the details, this requirement may be understood from the relation

$$\begin{aligned} B^{-1} &= \begin{pmatrix} I & -P_{FF}^{-1} A_{FC} \\ & I \end{pmatrix} \begin{pmatrix} P_{FF}^{-1} & \\ & S^{-1} \end{pmatrix} \begin{pmatrix} I & \\ -A_{CF} P_{FF}^{-1} & I \end{pmatrix} \\ &= \begin{pmatrix} I \\ 0 \end{pmatrix} P_{FF}^{-1} \begin{pmatrix} I & 0 \end{pmatrix} + \begin{pmatrix} -P_{FF}^{-1} A_{FC} \\ I \end{pmatrix} S^{-1} \begin{pmatrix} -A_{CF} P_{FF}^{-1} & I \end{pmatrix}. \end{aligned}$$

This preconditioner acts therefore as an additive two-level scheme, with prolongation from the coarse variables space defined by

$$\tilde{p} = \begin{pmatrix} -P_{FF}^{-1} A_{FC} \\ I \end{pmatrix}.$$

If A is a symmetric M -matrix with nonnegative row-sum, theoretical and numerical results show that MILU factorizations of A_{FF} possess all the desired properties [20, 22, 25]. These factorizations are computed so as to satisfy the row-sum criterion

$$P_{FF} \mathbf{e}_F = A_{FF} \mathbf{e}_F,$$

where $\mathbf{e}_F = (1 \ \dots \ 1)^T$. Note that if $A_{FF} \mathbf{e}_F + A_{FC} \mathbf{e}_C \approx 0$, this rule ensures that the interpolation (4.2) is a good approximation for the constant vector. This is precisely the main requirement for a “correct” interpolation for the class of applications targeted in this paper (that is, second order scalar elliptic PDEs discretized by finite differences or finite elements with nodal basis functions). We therefore decided to use MILU factorizations as well for non- M - or nonsymmetric matrices. We stress, however, that this approach is likely to be insufficient for other classes of problems like, for instance, coupled systems of PDEs, which are the subject of further investigations.

Now, MILU factorizations are not necessarily well defined if A_{FF} is not an M -matrix with nonnegative row-sum. Of course, instabilities (small or negative pivots) are much less likely than when factorizing the whole matrix, since the restriction to F nodes strengthens the relative weight of the diagonal elements. Nevertheless, as is, the method is not breakdown-free. More generally, there is no guarantee that the conditioning of A_{FF} is good enough so that a cheap MILU factorization will deliver a nice approximation P_{FF} .

This leads us to consider a “dynamic” factorization algorithm that reallocates to the C set the nodes for which the corresponding pivot would be too small relative

ALGORITHM 4.1 (adaptive MILU factorization of A_{FF}).

Input: $n \times n$ matrix $A = (a_{ij})$;
 stability threshold γ (default: $\gamma = \frac{3}{5}$);
 partitioning (F, C) of $[1, n]$ (tentative F/C partitioning),
 associated aggregates $G_i, i \in C$;
 Boolean matrix (η_{ij}) indicating where fill-in is allowed.

Output: partitioning (F, C) of $[1, n]$ (final F/C partitioning),
 associated aggregates $G_i, i \in C$;
 MILU factorization $P_{FF} = L_{FF} Q_{FF}^{-1} U_{FF}$ of A_{FF}
 (where $\text{diag}(L_{FF}) = \text{diag}(U_{FF}) = Q_{FF}$) such that $Q_{FF} \geq \gamma \text{diag}(A_{FF})(Q_{FF} = (q_{ij}), L_{FF} = (l_{ij}), U_{FF} = (u_{ij}))$.

Initialization: $\text{Repeat} = \text{True}$.

Algorithm:

1. $\text{Repeat} = \text{False}$.
2. (re)initialize:
 $Q_{FF} = \text{diag}(A_{FF}), L_{FF} = \text{lower}(A_{FF}), U_{FF} = \text{upper}(A_{FF})$.
3. for $k = 1, \dots, n, k \in F$:
 if $q_{kk} \geq \gamma a_{kk}$:
 (eliminate row and column k in A_{FF} according to the MILU algorithm)
 $\forall i > k, i \in F$:

$$q_{ii} = q_{ii} - \frac{l_{ik} u_{ki}}{q_{kk}}$$

$$\forall j > k, j \in F, j \neq i$$
 if $\eta_{ij} = 0$: $q_{ii} = q_{ii} - \frac{l_{ik} u_{kj}}{q_{kk}}$
 if $\eta_{ij} = 1$ and $j > i$: $u_{ij} = u_{ij} - \frac{l_{ik} u_{kj}}{q_{kk}}$
 if $\eta_{ij} = 1$ and $j < i$: $l_{ij} = l_{ij} - \frac{l_{ik} u_{kj}}{q_{kk}}$
 otherwise: $F = F \setminus \{k\}, C = C \cup \{k\}$;
 for $i \in C$ such that $k \in G_i$: $G_i = G_i \setminus \{k\}$;
 $G_k = \{k\}$;
 $\text{Repeat} = \text{True}$.
4. If (Repeat) , GoTo 1, possibly decreasing the value of γ .

to the corresponding diagonal element in A_{FF} . The detailed procedure is given in Algorithm 4.1. In this algorithm, γ is the threshold for this reallocation. That is, letting $P_{FF} = L_{FF} Q_{FF}^{-1} U_{FF}$ with $\text{diag}(L_{FF}) = \text{diag}(U_{FF}) = Q_{FF}$ be the tentative MILU factorization of A_{FF} , a node i is reallocated to the C set if and only if $(Q_{FF})_{ii} < \gamma(A_{FF})_{ii}$.

We iterate on the factorization because removing a posteriori a row and a column disturbs the relation $(P_{FF} \mathbf{e}_F)_i = (A_{FF} \mathbf{e}_F)_i$ on previously computed rows i . Hence, if some nodes are moved to F , the factorization has to be recomputed in a second pass. In principle, during this refactorization, some more nodes may be moved, in which case a third factorization would be necessary, and so on. To maintain low set-up cost, it is mandatory to keep small the number of such refactorizations. That is why we suggest decreasing the threshold γ from one pass to the next (see step 4 of Algorithm 4.1). There is, of course, some tradeoff here between set-up cost and stability issues. We could, however, hardly recommend any particular strategy because, in the numerical

experiments we have performed so far, when some nodes were moved to C during the first pass, *the second factorization always satisfied the stability criterion* $Q_{FF} \geq \gamma \operatorname{diag}(A_{FF})$ *for the value of the threshold γ used during the first pass.* Hence, any possible strategy would anyway accept the F/C partitioning resulting from the first pass.

We now discuss the proposed default value $\gamma = \frac{3}{5}$. This may look relatively severe, but remember that one of the goals is to guarantee the good conditioning of $P_{FF}^{-1}A_{FF}$. In [1, Lemma 10.2] it is proven that, in the symmetric case, if

$$(4.3) \quad \lambda_{\min} \left(Q_{FF} - \xi (A_{FF} - (L_{FF} - Q_{FF}) - (U_{FF} - Q_{FF})) \right) \geq 0$$

holds for some $\xi > \frac{1}{2}$, then

$$\lambda_{\max} (P_{FF}^{-1}A_{FF}) \leq \frac{1}{2 - \xi^{-1}},$$

whereas $\lambda_{\min} (P_{FF}^{-1}A_{FF}) \geq 1$ holds if A_{FF} is an M -matrix.¹ If one discards all fills (that is, if one uses Algorithm 4.1 with $\eta_{ij} = 0$ for all i, j), one has $A_{FF} - (L_{FF} - Q_{FF}) - (U_{FF} - Q_{FF}) = \operatorname{diag}(A_{FF})$, and condition (4.3) reduces to $Q_{FF} \geq \xi \operatorname{diag}(A_{FF})$, which is precisely the one targeted by the algorithm with $\xi = \gamma$. The default threshold then ensures $\lambda_{\max} (P_{FF}^{-1}A_{FF}) \leq 3$.

In our experiments we do not allow fill-in, but we accept the update of entries already nonzero in A_{FF} (that is, we set $\eta_{ij} = 1 \iff a_{ij} \neq 0$). Then, it is difficult to know whether (4.3) still holds for some $\xi > \frac{1}{2}$, and what could be the relationship between ξ and γ . We nevertheless observe that the proposed default threshold is efficient to improve the conditioning of $P_{FF}^{-1}A_{FF}$, especially in highly nonsymmetric cases for which there is no supporting theory.

In most cases, this improvement is obtained nearly for free, only a fairly small amount of nodes being moved from F to C . The effect is more dramatic on some difficult highly nonsymmetric problems, for which n_c was increased by up to 20% at some coarser levels. Note that we are anyway comfortable in this respect, since, on account of the fast coarsening resulting from our double pairwise aggregation strategy, there is plenty of room for moving nodes from F to C before recovering a prohibitively slow coarsening.

4.2. The scaling factor α . We already saw that coarse grid matrices computed by the aggregation rule (2.3) are somewhat overweight. We also saw that it is not necessary to finely tune the scaling parameter α in (4.1). For instance, in light of Figure 1, taking uniformly $\alpha = \frac{1}{3}$ seems enough.

The main goal of this subsection is therefore to report our observation that in a few cases results tend to be (slightly) improved by taking somewhat larger values for α . These cases are more precisely those where a significant number of nodes are moved from F to C by Algorithm 4.1. One should not be surprised at this observation: considering the formula (2.3), clearly, fewer nodes on average in the aggregates implies less overweighting. Therefore, instead of applying uniformly $\alpha = \frac{1}{3}$, we propose the default rule $\alpha = \frac{4n_C}{3n}$, that is

$$(4.4) \quad S = \frac{4n_C}{3n} \hat{A}_C$$

(observe that $\alpha = \frac{1}{3}$ when $\frac{n}{n_C} = 4$).

¹ $\lambda_{\min}(C)$ and $\lambda_{\max}(C)$ stand for, respectively, the smallest and the largest eigenvalues of C .

4.3. From two- to multilevel. To apply the two-level preconditioner (1.6), one needs to solve a system

$$(4.5) \quad B \mathbf{v} = \mathbf{g}$$

at each iteration. This is performed with the following steps:

1. $\mathbf{y}_F = P_{FF}^{-1} \mathbf{g}_F$,
2. $\mathbf{y}_C = \mathbf{g}_C - A_{CF} \mathbf{y}_F$,
3. solve $S \mathbf{v}_C = \mathbf{y}_C$,
4. $\mathbf{v}_F = P_{FF}^{-1} (\mathbf{g}_F - A_{FC} \mathbf{v}_C)$.

A multilevel scheme is obtained when the solution to

$$(4.6) \quad S \mathbf{v}_C = \mathbf{y}_C$$

at step 3 is computed only approximately, using a preconditioner B_S for S that is defined with the same algorithms and rules as those used to define B . This principle is then applied recursively until the coarse grid matrix is sufficiently small so that it is more advantageous to factorize it exactly with some standard software.

The simplest approach consists of using $\mathbf{v}_C = B_S^{-1} \mathbf{y}_C$, leading to a so-called V-cycle. This is unfortunately not optimal with this kind of preconditioner. We therefore consider a more sophisticated approach, in which an approximate solution to (4.6) is computed by running a few steps of a Krylov subspace method with B_S as preconditioner. This idea has already been followed in [24] and traces back to [4]. It is important to recall that the approach is stable, in the sense that a small error in the solution to (4.6) cannot be amplified and entails a large error in the solution to (4.5). More precisely, as shown in [24], the residual in (4.5) is the residual in (4.6) padded with zero for the F entries; in the symmetric case, one has further that the energy norm of the error is the same in (4.5) and in (4.6).

Concerning the practical details, we made essentially the same choices as in [24]. We review them below, with a few comments.

- Except at the coarsest level, the so defined preconditioner is slightly variable from step to step. We therefore select Krylov subspace methods explicitly designed to accommodate variable or “inexact” preconditioning, namely the flexible CG method [15, 21] in the symmetric case and FGMRES [27, 29] in the nonsymmetric one. In all cases, the initial approximation is the zero vector.
- Inner iterations are exited when the residual $\mathbf{r}_C$ in (4.6) satisfies $\|\mathbf{r}_C\| \leq 0.35\|\mathbf{y}_C\|$, or when the number of iterations reaches its maximal value, set equal to $\text{int}[nz(A)/nz(S)]$, where $\text{int}[\cdot]$ and $nz(\cdot)$ mean the integer part of and the number of nonzero entries in, respectively. As discussed in [24], the latter criterion ensures that the global cost of one application of the preconditioner is bounded independently of the number of levels; with the choices made for the coarsening process, this maximal number of iterations is most often equal to 3.
- The coarse grid matrix is factorized exactly when the estimated cost for this exact factorization represents less than a fraction of one unpreconditioned CG iteration for the whole system. This fraction is set equal to 1 in the symmetric case and 0.2 in the nonsymmetric one (for which the real cost may be larger than the estimated one, because of the need for pivoting).
- We use the CERFACS implementation of FGMRES [14], whereas the coarse grid matrices at the coarsest level are factorized using routines of the MA27 and MA28 suites from the HSL archive [18].

5. Numerical results. We consider the following test problems.

PROBLEM 1 (model 2D anisotropic problem). *Linear system resulting from the five point finite difference approximation of*

$$-a_x \frac{\partial^2 u}{\partial x^2} - a_y \frac{\partial^2 u}{\partial y^2} = 1 \quad \text{in } \Omega = (0, 1) \times (0, 1)$$

with constant coefficients a_x, a_y and boundary conditions

$$\begin{cases} u = 0 & \text{on } x = 1, 0 \leq y \leq 1, \\ \frac{\partial u}{\partial n} = 0 & \text{elsewhere on } \partial\Omega. \end{cases}$$

We use a uniform mesh with constant mesh size h in both directions.

PROBLEM 2 (model 2D anisotropic problem, bilinear finite elements). *Linear system resulting from the bilinear finite element approximation of*

$$-a_x \frac{\partial^2 u}{\partial x^2} - a_y \frac{\partial^2 u}{\partial y^2} = 1 \quad \text{in } \Omega = (0, 1) \times (0, 1)$$

with constant coefficients a_x, a_y and homogeneous Dirichlet boundary conditions everywhere on $\partial\Omega$. We use a uniform mesh with constant mesh size h in both directions.

PROBLEM 3 (2D problem with anisotropy and discontinuity). *Linear system resulting from the five point finite difference approximation of*

$$-\frac{\partial}{\partial x} \left(a_x \frac{\partial u}{\partial x} \right) - \frac{\partial}{\partial y} \left(a_y \frac{\partial u}{\partial y} \right) = f \quad \text{in } \Omega = (0, 1) \times (0, 1)$$

with boundary conditions

$$\begin{cases} u = 0 & \text{on } y = 1, 0 \leq x \leq 1, \\ \frac{\partial u}{\partial n} = 0 & \text{elsewhere on } \partial\Omega \end{cases}$$

and coefficients given by

$$\begin{cases} a_x = 1, & a_y = d, & f = 0 & \text{in } (0.65, 0.95) \times (0.05, 0.65), \\ a_x = d, & a_y = 1, & f = 0 & \text{in } (0.25, 0.45) \times (0.25, 0.45), \\ a_x = d, & a_y = d, & f = 1 & \text{in } (0.05, 0.25) \times (0.65, 0.95), \\ a_x = 1, & a_y = 1, & f = 0 & \text{elsewhere,} \end{cases}$$

where d is a parameter. We use a uniform mesh with constant mesh size h in both directions.

PROBLEM 4 (model 3D anisotropic problem). *Linear system resulting from the seven point finite difference approximation of*

$$-a_x \frac{\partial^2 u}{\partial x^2} - a_y \frac{\partial^2 u}{\partial y^2} - a_z \frac{\partial^2 u}{\partial z^2} = 1 \quad \text{in } \Omega = (0, 1) \times (0, 1) \times (0, 1)$$

with constant coefficients a_x, a_y, a_z and boundary conditions

$$\begin{cases} u = 0 & \text{on } x = 1, 0 \leq y, z \leq 1, \\ \frac{\partial u}{\partial n} = 0 & \text{elsewhere on } \partial\Omega. \end{cases}$$

We use a uniform mesh with constant mesh size h in all directions.

PROBLEM 5 (3D problem with discontinuity and local refinement). *Linear system resulting from the seven point finite difference approximation of*

$$-\frac{\partial}{\partial x} \left(a_x \frac{\partial u}{\partial x} \right) - \frac{\partial}{\partial y} \left(a_y \frac{\partial u}{\partial y} \right) - \frac{\partial}{\partial z} \left(a_z \frac{\partial u}{\partial z} \right) = f \quad \text{in } \Omega = (0, 1) \times (0, 1) \times (0, 1)$$

with boundary conditions

$$\begin{cases} u = 0 & \text{on } z = 1, 0 \leq x, y \leq 1, \\ \frac{\partial u}{\partial n} = 0 & \text{elsewhere on } \partial\Omega \end{cases}$$

and coefficients given by

$$\begin{cases} a_x = a_y = a_z = d, & f = 1 & \text{in } (\frac{1}{4}, \frac{3}{4}) \times (\frac{1}{4}, \frac{3}{4}) \times (\frac{1}{4}, \frac{3}{4}), \\ a_x = a_y = a_z = 1, & f = 0 & \text{elsewhere,} \end{cases}$$

where d is a parameter. We use either a uniform mesh with constant mesh size h in all directions or a nonuniform mesh, structurally regular (Cartesian) but refined near material interfaces, with mesh sizes $h_x = h(x)$, $h_y = h(y)$, $h_z = h(z)$ varying according to

$$h(t) = \begin{cases} \frac{1}{400} & \text{if } 0.2 \leq t \leq 0.3 \text{ or } 0.7 \leq t \leq 0.8, \\ \frac{1}{25} & \text{otherwise} \end{cases}$$

to define a $101 \times 101 \times 101$ grid, or half these values to define a $201 \times 201 \times 201$ grid.

PROBLEM 6 (2D convection-diffusion equation). Linear system resulting from the five point finite difference approximation (upwind scheme) of

$$-\nu \Delta u + \bar{v} \bar{\nabla} u = 0 \quad \text{in } \Omega = (0, 1) \times (0, 1)$$

with boundary conditions

$$\begin{cases} u = 1 & \text{on } y = 1, 0 \leq x \leq 1, \\ u = 0 & \text{elsewhere on } \partial\Omega \end{cases}$$

and convective flow given by

$$\bar{v}(x, y) = \begin{pmatrix} x(1-x)(2y-1) \\ -(2x-1)y(1-y) \end{pmatrix};$$

$\nu = \infty$ corresponds to the Laplace equation (no convection). We use either a uniform mesh with constant mesh size h in both directions or a stretched mesh, structurally regular (Cartesian) but refined in the neighborhood of the boundaries, in such a way that the ratio of maximum mesh size to minimum mesh size is equal to 200, the ratio of subsequent mesh sizes being kept constant.

PROBLEM 7 (3D convection-diffusion equation). Linear system resulting from the seven point finite difference approximation (upwind scheme) of

$$-\nu \Delta u + \bar{v} \bar{\nabla} u = 0 \quad \text{in } \Omega = (0, 1) \times (0, 1) \times (0, 1)$$

with boundary conditions

$$\begin{cases} u = 1 & \text{on } z = 1, 0 \leq x, y \leq 1, \\ u = 0 & \text{elsewhere on } \partial\Omega \end{cases}$$

and convective flow given by

$$\bar{v}(x, y, z) = \begin{pmatrix} 2x(1-x)(2y-1)z \\ -(2x-1)y(1-y) \\ -(2x-1)(2y-1)z(1-z) \end{pmatrix};$$

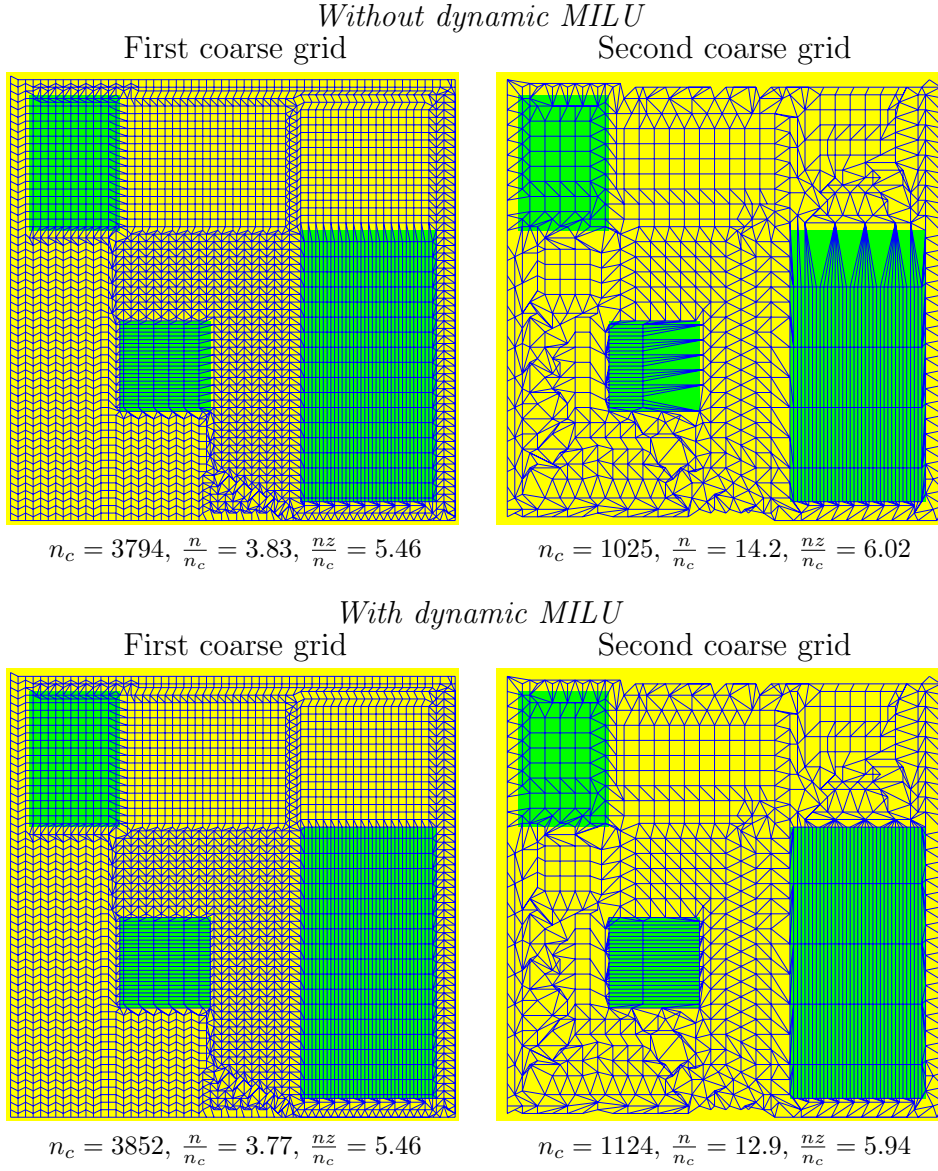


FIG. 2. Coarsening for Problem 3 with $d = 100$ and $h^{-1} = 120$ ($n = 14520$; $\frac{nz}{n} = 4.96$, where nz stands for the number of nonzero entries).

$\nu = \infty$ corresponds to the Laplace equation (no convection). We use either a uniform mesh with constant mesh size h in all directions or a stretched mesh, structurally regular (Cartesian) but refined in the neighborhood of the boundaries, in such a way that the ratio of maximum mesh size to minimum mesh size is equal to 200, the ratio of subsequent mesh sizes being kept constant.

We first illustrate the behavior of our aggregation scheme. In this respect, the most interesting symmetric problem is perhaps Problem 3. For $d = 100$ and $h^{-1} = 120$, the rules given in section 4.3 entail that there are only three levels, the second coarse grid being already sufficiently small to be factorized exactly. Figure 2 displays these first two coarse grids (displaying the fine grid would be uninteresting

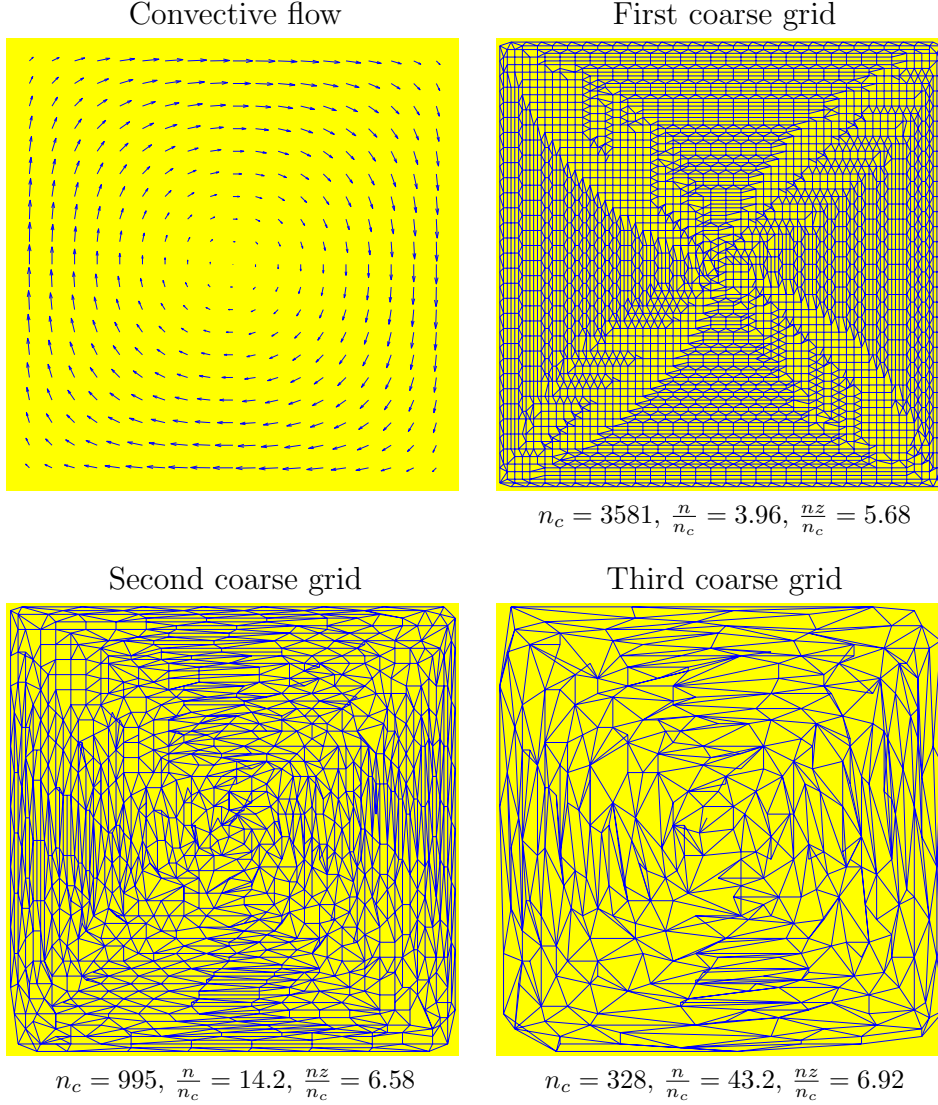


FIG. 3. *Coarsening with dynamic MILU for Problem 6 with $\nu = 10^{-4}$ and a uniform 121×121 grid ($n = 14161$; $\frac{nz}{n} = 4.96$, where nz stands for the number of nonzero entries).*

because it is uniform). On the top we illustrate our coarsening without dynamic MILU (Algorithm 2.2 alone), whereas the bottom figures correspond to the combined use of Algorithms 2.2 and 4.1. For clarity, the zones in which either $a_x \neq 1$ or $a_y \neq 1$ or both have been slightly shadowed (in green with color display). One sees the semi-coarsening $h \rightarrow 4h$ in the anisotropic zones, and one may check that the refinement of the F/C partitioning by dynamic MILU has only a marginal impact on the number of nodes in the coarse grids. Nevertheless, it not only guarantees the quality of the MILU factorization of A_{FF} , but seems also to improve the regularity of the coarsening. This suggests that it could be useful in other contexts as well, offering, for instance, a cheap alternative to the iterative improvement of the coarsening by so-called compatible relaxation [9, 13].

TABLE 1
Results for Problem 1.

a_x	a_y	$h^{-1} = 600$				$h^{-1} = 1200$			
		$\frac{n}{n_c}$	inner	iter.	sol.	$\frac{n}{n_c}$	inner	iter.	sol.
Geometric multigrid									
1	1	3.99	1.00	9	28.3	4.00	1.00	9	28.5
Aggregation-based algebraic multilevel									
1	1	3.99	1.94	18	63.3	4.00	2.00	19	70.3
1	2	3.98	1.95	19	63.0	3.99	1.95	20	68.4
1	4	3.97	2.10	20	70.6	3.98	2.10	21	76.2
1	10	3.95	2.05	22	72.1	3.98	2.10	21	70.7
1	10^2	3.95	2.00	22	69.0	3.98	2.06	18	58.9
1	10^4	3.95	1.94	18	54.9	3.98	1.89	18	55.4

TABLE 2
Results for Problem 2.

a_x	a_y	$h^{-1} = 600$				$h^{-1} = 1200$			
		$\frac{n}{n_c}$	inner	iter.	sol.	$\frac{n}{n_c}$	inner	iter.	sol.
1	1	3.99	1.94	17	55.2	3.99	1.94	17	56.4
1	2	3.99	2.20	15	54.0	4.00	2.20	15	55.6
1	4	3.99	2.11	19	68.9	4.00	2.05	20	72.2
1	10	3.99	2.00	20	68.5	4.00	2.05	20	74.2
1	10^2	3.99	1.95	20	69.1	4.00	1.75	24	80.0
1	10^4	3.99	1.75	20	65.5	4.00	1.83	18	60.3

We also illustrate the coarsening on the unsymmetric Problem 6 with $\nu = 10^{-4}$ and a uniform 121×121 grid. Here, four levels and thus three coarse grids are necessary. (As already written in section 4.3, we have to use a more strict rule in the nonsymmetric case because of the need for pivoting.) We show these coarse grids in Figure 3, displaying only those obtained with dynamic MILU because there is nothing new to learn from those obtained without this enhancement. We also graphically represent the magnitude and the direction of the convective flow $\bar{v}$. This allows us to see that semicoarsening effects appear in the regions where the flow is largest in magnitude and aligned with the grid lines.

We now assess the performances of our multilevel preconditioner. All problems were solved by the flexible CG method [15, 21] (Problems 1–5) or FGMRES [27, 29] with restart parameter set equal to 10 (Problems 6–7). The initial approximation was the zero vector, and the stopping criterion was a decrease of the relative residual error by 10^{-6} . In all cases, we used the algebraic multilevel preconditioner, as described in section 4, based on the aggregation procedure given in section 2. All parameters were set to their default value.

The results are reported in Tables 1–7. In these tables, n/n_C is the coarsening ratio at the first level, *inner* the mean number of inner iterations to (approximately) solve the coarse grid system on this first level, *iter.* the number of (outer) iterations to solve the system, and *sol.* the solution cost (number of floating point operations), relative to the cost of 1 unpreconditioned CG iteration. To have a more precise idea of this relative efficiency, we also solved the model Problem 1 with $a_x = a_y = 1$ using a standard geometric multigrid scheme accelerated by CG (bilinear interpolation, one pre- and one postdamped Jacobi smoothing step with $\omega = 0.5$, V-cycle).

TABLE 3
Results for Problem 3.

d	$h^{-1} = 600$				$h^{-1} = 1200$			
	$\frac{n}{n_c}$	inner	iter.	sol.	$\frac{n}{n_c}$	inner	iter.	sol.
1	3.99	1.76	21	67.1	4.00	1.76	21	68.8
2	3.97	2.00	20	69.8	3.99	2.00	23	82.9
4	3.96	2.04	24	84.1	3.98	2.00	25	90.3
10	3.95	2.04	24	84.2	3.98	2.04	26	94.0
10^2	3.95	2.04	24	82.5	3.98	2.00	26	90.6
10^4	3.95	1.96	26	88.0	3.98	2.04	27	95.6
10^6	3.95	2.15	26	92.2	3.98	2.00	31	107.9

TABLE 4
Results for Problem 4.

a_x	a_y	a_z	$h^{-1} = 100$				$h^{-1} = 200$			
			$\frac{n}{n_c}$	inner	iter.	sol.	$\frac{n}{n_c}$	inner	iter.	sol.
1	1	1	4.00	2.00	19	91.0	4.00	2.00	19	93.8
1	4	1	4.00	2.00	19	91.0	4.00	2.00	19	93.8
1	4	2	3.90	2.16	19	88.8	3.95	2.37	19	95.1
1	4	4	3.93	2.06	17	73.0	3.96	2.06	18	79.0
1	16	1	3.74	2.05	20	80.6	3.87	2.05	20	91.3
1	16	4	3.81	2.16	19	85.5	3.90	2.26	19	93.7
1	16	16	3.92	2.13	16	72.6	3.96	2.18	17	78.7
1	10^2	1	3.74	2.05	19	74.2	3.87	1.78	27	110.2
1	10^2	10	3.74	2.11	19	87.3	3.87	2.10	21	88.2
1	10^2	10^2	3.92	2.12	17	74.0	3.96	2.19	16	72.8
1	10^4	1	3.74	2.00	15	59.0	3.87	2.29	14	62.2
1	10^4	10^2	3.74	2.06	17	69.2	3.87	1.95	21	89.6
1	10^4	10^4	3.92	2.06	17	72.1	3.96	2.06	17	74.2

TABLE 5
Results for Problem 5.

d	101 $\times$ 101 $\times$ 101 grid				201 $\times$ 201 $\times$ 201 grid			
	$\frac{n}{n_c}$	inner	iter.	sol.	$\frac{n}{n_c}$	inner	iter.	sol.
Uniform mesh								
1	4.00	2.05	19	85.0	4.00	2.10	20	94.4
4	4.00	2.05	21	96.8	4.00	2.00	22	105.8
10	4.00	2.05	21	97.1	4.00	2.05	21	104.5
10^2	4.00	2.05	22	104.8	4.00	2.05	22	108.3
10^4	4.00	2.09	22	104.9	4.00	2.05	22	109.3
10^6	4.00	2.09	22	106.5	4.00	2.04	23	113.7
Nonuniform mesh								
1	3.93	2.05	22	100.3	3.97	2.04	24	105.2
2	3.89	2.04	23	107.4	3.95	1.96	26	115.0
4	3.90	2.05	22	103.1	3.95	2.04	24	111.4
10	3.89	2.05	22	103.8	3.95	2.04	24	111.7
10^2	3.89	2.04	23	106.0	3.95	2.04	24	110.4
10^4	3.89	2.04	23	108.1	3.95	2.08	25	118.4
10^6	3.89	2.04	23	108.3	3.95	2.13	24	112.9

TABLE 6
Results for Problem 6.

ν	601 $\times$ 601 grid				1201 $\times$ 1201 grid			
	$\frac{n}{n_c}$	inner	iter.	sol.	$\frac{n}{n_c}$	inner	iter.	sol.
Uniform mesh								
∞	3.99	1.85	13	61.7	4.00	1.85	13	61.9
1	3.96	2.00	15	71.6	3.85	2.00	17	87.0
10^{-1}	3.99	2.00	15	69.8	3.99	1.87	15	67.1
10^{-2}	3.98	2.00	17	87.4	3.96	2.00	18	89.0
10^{-4}	3.94	2.81	21	149.7	3.93	2.90	21	176.3
10^{-6}	3.99	2.83	23	158.3	4.00	2.87	23	178.0
Stretched mesh								
∞	3.98	1.89	18	78.6	3.98	1.89	18	79.9
1	3.98	1.88	16	68.4	3.98	1.89	18	79.9
10^{-1}	3.98	1.94	17	74.0	3.98	1.89	18	79.8
10^{-2}	3.97	2.00	17	76.2	3.99	1.89	18	80.4
10^{-4}	3.79	2.40	25	130.0	3.83	2.71	24	138.4
10^{-6}	3.41	1.97	29	157.8	3.61	2.85	27	180.8

TABLE 7
Results for Problem 7.

ν	101 $\times$ 101 $\times$ 101 grid				201 $\times$ 201 $\times$ 201 grid			
	$\frac{n}{n_c}$	inner	iter.	sol.	$\frac{n}{n_c}$	inner	iter.	sol.
Uniform mesh								
∞	4.00	2.00	15	79.6	4.00	2.00	15	82.0
1	3.76	2.00	17	109.7	3.81	2.00	17	108.3
10^{-1}	3.80	2.00	16	99.4	3.81	2.00	17	111.2
10^{-2}	3.75	2.00	18	119.7	3.84	1.94	18	111.7
10^{-4}	3.93	2.00	21	136.9	3.93	2.00	21	136.8
10^{-6}	3.93	2.00	26	167.2	3.96	2.00	30	203.0
Stretched mesh								
∞	3.91	1.94	16	79.3	3.94	1.88	17	85.8
1	3.91	1.94	16	80.0	3.94	1.88	17	85.8
10^{-1}	3.91	2.00	16	79.2	3.94	1.88	17	85.7
10^{-2}	3.92	1.65	20	95.5	3.95	1.94	17	85.3
10^{-4}	3.46	1.81	21	117.1	3.64	1.87	23	125.8
10^{-6}	3.64	2.00	27	171.0	3.28	2.00	27	186.3

One sees that our method is about 2.5 times slower than a well-tuned geometric multigrid method on a model 2D problem. This is, however, compensated by a great robustness. The different test problems feature several difficulties: strong anisotropies, sometimes with varying direction; large jumps in the PDE coefficients; anisotropy induced by nonuniform grids; large positive off-diagonal entries induced by the bilinear finite element approximation of an anisotropic problem (and the associated difficulty in detecting the right direction of strong coupling); highly nonsymmetric matrices without any regular pattern because they originate from a convective flow that is varying both in magnitude and direction. Despite this, the tables look rather similar, with about the same coarsening ratios and numbers of inner or outer iterations, whereas, except for the convection-diffusion problems with extremely low viscosity,

the relative solution cost is never larger than 1.7 times that for the model 2D problem.

Acknowledgments. I thank the referees for their careful reading and helpful suggestions.

REFERENCES

- [1] O. AXELSSON, *Iterative Solution Methods*, Cambridge University Press, Cambridge, UK, 1994.
- [2] O. AXELSSON AND M. NEYTCHIEVA, *Algebraic multilevel iterations for Stieltjes matrices*, Numer. Linear Algebra Appl., 1 (1994), pp. 213–236.
- [3] O. AXELSSON AND P. VASSILEVSKI, *Algebraic multilevel preconditioning methods*, II, SIAM J. Numer. Anal., 27 (1990), pp. 1569–1590.
- [4] O. AXELSSON AND P. S. VASSILEVSKI, *Variable-step multilevel preconditioning methods*. I. *Self-adjoint and positive definite elliptic problems*, Numer. Linear Algebra Appl., 1 (1994), pp. 75–101.
- [5] R. E. BANK AND R. K. SMITH, *An algebraic multilevel multigraph algorithm*, SIAM J. Sci. Comput., 23 (2002), pp. 1572–1592.
- [6] R. E. BANK AND C. WAGNER, *Multilevel ILU decomposition*, Numer. Math., 82 (1999), pp. 543–576.
- [7] E. F. F. BOTTA AND F. W. WUBS, *Matrix renumbering ILU: An effective algebraic multilevel ILU preconditioner for sparse matrices*, SIAM J. Matrix Anal. Appl., 20 (1999), pp. 1007–1026.
- [8] D. BRAESS, *Towards algebraic multigrid for elliptic problems of second order*, Computing, 55 (1995), pp. 379–393.
- [9] A. BRANDT, *General highly accurate algebraic coarsening*, Electron. Trans. Numer. Anal., 10 (2000), pp. 1–20.
- [10] M. BREZINA, A. J. CLEARY, R. D. FALGOUT, V. E. HENSON, J. E. JONES, T. A. MANTEUFFEL, S. F. MCCORMICK, AND J. W. RUGE, *Algebraic multigrid based on element interpolation (AMGe)*, SIAM J. Sci. Comput., 22 (2000), pp. 1570–1592.
- [11] T. CHARTIER, R. D. FALGOUT, V. E. HENSON, J. JONES, T. MANTEUFFEL, S. MCCORMICK, J. RUGE, AND P. S. VASSILEVSKI, *Spectral AMGe (ρ AMGe)*, SIAM J. Sci. Comput., 25 (2003), pp. 1–26.
- [12] A. J. CLEARY, R. D. FALGOUT, V. E. HENSON, J. E. JONES, T. A. MANTEUFFEL, S. F. MCCORMICK, G. N. MIRANDA, AND J. W. RUGE, *Robustness and scalability of algebraic multigrid*, SIAM J. Sci. Comput., 21 (2000), pp. 1886–1908.
- [13] R. D. FALGOUT AND P. S. VASSILEVSKI, *On generalizing the algebraic multigrid framework*, SIAM J. Numer. Anal., 42 (2004), pp. 1669–1693.
- [14] V. FRAYSSÉ, L. GRAUD, AND S. GRATTON, *Algorithm 842: A set of GMRES routines for real and complex arithmetics on high performance computers*, ACM Trans. Math. Software, 31 (2005), pp. 228–238.
- [15] G. H. GOLUB AND Q. YE, *Inexact preconditioned conjugate gradient method with inner-outer iterations*, SIAM J. Sci. Comput., 21 (1999), pp. 1305–1320.
- [16] W. HACKBUSCH, *Multi-grid Methods and Applications*, Springer-Verlag, Berlin, 1985.
- [17] V. E. HENSON AND P. S. VASSILEVSKI, *Element-free AMGe: General algorithms for computing interpolation weights in AMG*, SIAM J. Sci. Comput., 23 (2001), pp. 629–650.
- [18] *HSL archive*, available via <http://hsl.rl.ac.uk/archive/hslarchive>.
- [19] J. E. JONES AND P. S. VASSILEVSKI, *AMGE based on element agglomeration*, SIAM J. Sci. Comput., 23 (2001), pp. 109–133.
- [20] Y. NOTAY, *Using approximate inverses in algebraic multilevel methods*, Numer. Math., 80 (1998), pp. 397–417.
- [21] Y. NOTAY, *Flexible conjugate gradients*, SIAM J. Sci. Comput., 22 (2000), pp. 1444–1460.
- [22] Y. NOTAY, *Optimal order preconditioning of finite difference matrices*, SIAM J. Sci. Comput., 21 (2000), pp. 1991–2007.
- [23] Y. NOTAY, *A robust algebraic multilevel preconditioner for non-symmetric M-matrices*, Numer. Linear Algebra Appl., 7 (2000), pp. 243–267.
- [24] Y. NOTAY, *Robust parameter free algebraic multilevel preconditioning*, Numer. Linear Algebra Appl., 9 (2002), pp. 409–428.
- [25] Y. NOTAY, *Algebraic multigrid and algebraic multilevel methods: A theoretical comparison*, Numer. Linear Algebra Appl., 12 (2005), pp. 419–451.
- [26] J. W. RUGE AND K. STÜBEN, *Algebraic multigrid (AMG)*, in Multigrid Methods, S. F. McCormick, ed., Frontiers in Appl. Math. 3, SIAM, Philadelphia, 1987, pp. 73–130.

- [27] Y. SAAD, *A flexible inner-outer preconditioned GMRES algorithm*, SIAM J. Sci. Comput., 14 (1993), pp. 461–469.
- [28] Y. SAAD, *ILUM: A multi-elimination ILU preconditioner for general sparse matrices*, SIAM J. Sci. Comput., 17 (1996), pp. 830–847.
- [29] Y. SAAD, *Iterative Methods for Sparse Linear Systems*, 2nd ed., SIAM, Philadelphia, 2003.
- [30] Y. SAAD AND B. SUCHOMEL, *ARMS: An algebraic recursive multilevel solver for general sparse linear systems*, Numer. Linear Algebra Appl., 9 (2002), pp. 359–378.
- [31] Y. SAAD AND J. ZHANG, *BILUM: Block versions of multielimination and multilevel ILU preconditioner for general sparse linear systems*, SIAM J. Sci. Comput., 20 (1999), pp. 2103–2121.
- [32] K. STÜBEN, *An introduction to algebraic multigrid*, in Multigrid, U. Trottenberg, C. W. Oosterlee, and A. Schüller, Academic Press, London, 2001, Appendix A, pp. 413–532.
- [33] U. TROTTEMBERG, C. W. OOSTERLEE, AND A. SCHÜLLER, *Multigrid*, Academic Press, London, 2001.
- [34] P. VANĚK, J. MANDEL, AND M. BREZINA, *Algebraic multigrid based on smoothed aggregation for second and fourth order problems*, Computing, 56 (1996), pp. 179–196.

REDUCING COMPLEXITY IN PARALLEL ALGEBRAIC MULTIGRID PRECONDITIONERS*

HANS DE STERCK[†], ULRIKE MEIER YANG[‡], AND JEFFREY J. HEYS[§]

Abstract. Algebraic multigrid (AMG) is a very efficient iterative solver and preconditioner for large unstructured sparse linear systems. Traditional coarsening schemes for AMG can, however, lead to computational complexity growth as problem size increases, resulting in increased memory use and execution time, and diminished scalability. Two new parallel AMG coarsening schemes are proposed that are based solely on enforcing a maximum independent set property, resulting in sparser coarse grids. The new coarsening techniques remedy memory and execution time complexity growth for various large three-dimensional (3D) problems. If used within AMG as a preconditioner for Krylov subspace methods, the resulting iterative methods tend to converge fast. This paper discusses complexity issues that can arise in AMG, describes the new coarsening schemes, and examines the performance of the new preconditioners for various large 3D problems.

Key words. parallel coarsening algorithms, algebraic multigrid, complexities, preconditioners

AMS subject classifications. 65F10, 65N55, 65Y05

DOI. 10.1137/040615729

1. Introduction. The algebraic multigrid (AMG) algorithm [1, 13, 14, 2] is one of the most efficient algorithms for solving large unstructured sparse linear systems that arise in a wide range of science and engineering applications. One of AMG's most desirable properties, especially in the context of large scale problems and massively parallel computing, is its potential for scalability. We consider a method scalable if the time required to produce the solution does not increase as the problem size and the number of processors increase, while the problem size per processor is constant. While iterative methods are generally scalable per iteration step, often the number of iterations depends on the problem size. However, the convergence of AMG has been shown to be independent of the problem size for model problems [5, 7].

Familiarity with the basic AMG algorithm is assumed in the remainder of this introductory section. A brief overview of the basic AMG algorithm is given in section 2. For real-life problems, the AMG algorithm has been shown to be a robust and efficient algorithm for a wide range of applications [5, 14]. Various parallel versions of AMG have been developed [4, 16, 7, 11, 10, 6]. For an overview, see [18].

Traditional coarsening schemes for AMG that are based on the coarsening heuristics originally proposed by Ruge and Stueben (RS) [13], tend to work well for problems that arise from the discretization of elliptic partial differential equations (PDEs) in

*Received by the editors September 23, 2004; accepted for publication (in revised form) by W.-P. Tang November 21, 2005; published electronically March 17, 2006. This work was performed under the auspices of the U.S. Department of Energy by University of California Lawrence Livermore National Laboratory under contract W-7405-Eng-48. It was also sponsored by the Department of Energy under grants DE-FC02-01ER25479 and DE-FG02-03ER25574, Lawrence Livermore National Laboratory under contract B533502, Sandia National Laboratory under contract 15268, and the National Science Foundation under VIGRE grant DMS-9810751 and grant DMS-0410318.

<http://www.siam.org/journals/simax/27-4/61572.html>

[†]Department of Applied Mathematics, University of Waterloo, Waterloo, ON N2L 3G1, Canada (hdesterck@uwaterloo.ca).

[‡]Center for Applied Scientific Computing, Lawrence Livermore National Laboratory, PO Box 808, Livermore, CA 94551 (umyang@llnl.gov).

[§]Chemical and Materials Engineering Department, Box 876006, Arizona State University, Tempe, AZ 85287-6006 (heys@asu.edu).

two spatial dimensions (2D). For many 2D problems, a solver with near-optimal scalability can be obtained, with the number of iterations that is required for convergence independent of the problem size n , and memory use, setup time, and computational complexity linearly proportional to n . However, when traditional AMG algorithms are applied to three-dimensional (3D) problems, numerical tests show that in many cases scalability is lost: while the number of iterations required may remain constant, computational complexities, particularly stencil size, may grow significantly leading to increased execution times and memory usage. This loss of scalability is already an issue on serial machines, but for larger problem sizes on parallel machines the effects are even more severe, with additional degradation due to increased communication. Complexity issues have been considered before. In [14] Stueben suggests the application of aggressive coarsening (which can be implemented as applying standard coarsening twice) to the finer levels, which leads to reduced complexities. Aggressive coarsening is also used in [10]. This approach also requires the use of long-range interpolation, which in combination with interpolation truncation is efficient for sequential computations. It is, however, difficult to implement in parallel, since it requires information that goes beyond immediate neighbors and might be located on processors that are not neighbor processors according to the graph of the original matrix.

We present here two new, simplified parallel coarsening algorithms. The first, which we call the parallel modified independent set (PMIS) algorithm, is a modification of an existing parallel maximal independent set algorithm by Luby [12, 9]. This algorithm can be implemented in such a way that the resulting coarse grid is independent of the number of processors and the distribution of the points across processors. The second, the hybrid modified independent set (HMIS) algorithm, is obtained by combining the PMIS algorithm with a one-pass RS scheme. We find that, using PMIS and HMIS coarsening, complexity growth problems largely disappear. Not surprisingly, we observe that, using regular AMG interpolation combined with PMIS and HMIS coarsening, AMG convergence factors deteriorate as a function of problem size, resulting in a loss of scalability. However, this convergence degradation can be counteracted effectively by using Krylov subspace acceleration of the AMG solvers. We show that the resulting preconditioners, with regard to total execution times and memory use, often significantly outperform preconditioners that use parallel coarsening schemes based on the heuristics used for classical sequential RS coarsening (e.g., the Cleary-Luby-Jones-Plassmann (CLJP) coarsening algorithm [7]). For large problems on parallel machines in the several 1,000 processor class, the newly proposed methods typically use less than half the memory and require less than half the computer time. Also, for the problem sizes and machines tested, the scalability in memory and execution time (especially AMG setup times) is generally much better than for the existing approaches. We investigate whether these improvements hold for a large class of structured problems arising from scalar elliptic 3D PDEs, including anisotropic and convection-diffusion problems, and investigate whether these improvements also hold for 3D PDE systems on unstructured grids, in particular, first-order system least squares (FOSLS) finite element discretizations of Stokes flow.

The paper is organized as follows. In the following section a brief overview is given of the basic AMG algorithm. In section 3 a more detailed discussion of the arising complexity issues is presented. Section 4 describes the new PMIS and HMIS coarsening algorithms. In section 5 the performance of the PMIS and HMIS schemes is compared with that of RS-based parallel coarsening strategies (namely CLJP and its hybrid variant, Falgout coarsening) for a large class of structured and unstructured problems arising from scalar elliptic 3D PDEs. Section 6 describes scaling studies for

a 3D PDE system modeling Stokes flow. Finally, conclusions are formulated in section 7.

2. Algebraic multigrid. In this section we give an outline of the basic principles and techniques that comprise the scalar AMG algorithm, and we define terminology and notation. Detailed explanations may be found in [13, 14, 2]. Consider a problem of the form

$$(2.1) \quad Au = f,$$

where A is a sparse real $n \times n$ matrix with entries a_{ij} . For convenience, the indices are identified with grid points, so that u_i denotes the value of u at point i , and the grid is denoted by $\Omega = \{1, 2, \dots, n\}$. While uniform two-level convergence has strictly been proven only for certain classes of symmetric positive definite matrices such as M-matrices [13], we will not restrict ourselves to such matrices here since in practice AMG has been shown to often converge for other matrices as well [14]. In any multigrid method, the central idea is that “smooth error,” e , that is not eliminated by relaxation must be removed by coarse-grid correction. This is done by solving the residual equation $Ae = r$ on a coarser grid, then interpolating the error back to the fine grid and using it to correct the fine-grid approximation by $u \leftarrow u + e$.

Using superscripts to indicate the level number, where 1 denotes the finest level so that $A^1 = A$ and $\Omega^1 = \Omega$, the components that AMG needs are as follows:

1. “Grids” $\Omega^1 \supset \Omega^2 \supset \dots \supset \Omega^M$.
2. Grid operators $A^1, A^2, \dots, A^M$.
3. Grid transfer operators:
Interpolation $P^k, k = 1, 2, \dots, M-1$,
Restriction $R^k, k = 1, 2, \dots, M-1$.
4. Smoothers $S^k, k = 1, 2, \dots, M-1$.

These components of AMG are determined in a first step, known as the *setup phase*.

AMG Setup Phase:

1. Set $k = 1$.
2. Partition Ω^k into disjoint sets C^k and F^k .
(a) Set $\Omega^{k+1} = C^k$.
(b) Define interpolation P^k .
3. Define R^k (often $R^k = (P^k)^T$).
4. Set $A^{k+1} = R^k A^k P^k$ (Galerkin condition).
5. Set up S^k , if necessary.
6. If Ω^{k+1} is small enough, set $M = k+1$ and stop. Otherwise, set $k = k+1$ and go to step 2.

Once the setup phase is completed, the *solve phase*, a recursively defined cycle, can be performed as follows, after having set $f^1 = f$ and chosen an arbitrary u^1 .

Algorithm: $MGV(A^k, R^k, P^k, S^k, u^k, f^k)$.

If $k = M$, solve $A^M u^M = f^M$ with a direct solver.

Otherwise:

Apply smoother S^k μ_1 times to $A^k u^k = f^k$.

Perform coarse grid correction:

Set $r^k = f^k - A^k u^k$.

Set $r^{k+1} = R^k r^k$.

Set $e^{k+1} = 0$.

Apply $MGV(A^{k+1}, R^{k+1}, P^{k+1}, S^{k+1}, e^{k+1}, r^{k+1})$.

Interpolate $e^k = P^k e^{k+1}$.

Correct the solution by $u^k \leftarrow u^k + e^k$.
 Apply smoother S^k μ_2 times to $A^k u^k = f^k$.

The algorithm above describes a $V(\mu_1, \mu_2)$ -cycle; other more complex cycles, such as W-cycles, are described in [2, 15]. In every V-cycle, the error is reduced by a certain factor, which is called the convergence factor. A sequence of V-cycles is executed until the error is reduced below a specified tolerance.

3. Complexity issues. While computational complexities are often constant for increasing problem sizes for matrices arising from 2D PDE problems (see [7] and section 5), they may grow significantly for many 3D problems. This growth can lead to a large memory requirement and a significantly increased amount of computational work, which may result in a severe loss of scalability that becomes prohibitive when large problem sizes are attempted. The loss of scalability manifests itself in non-scalable execution times for both the setup phase and the solve phase of the AMG algorithm.

There are two types of complexities that need to be considered: the *operator complexity* and the *stencil size*. The operator complexity, C_{op} , is defined as the sum of the number of nonzero matrix elements in the operator matrices on all grid levels, divided by the number of nonzero matrix elements of the fine grid operator matrix, A . It strongly influences memory use and execution time. The stencil size on a given level is defined as the average number of coefficients per matrix row, and significantly affects setup time, as growing stencil sizes substantially increase the number of operations required in the coarsening and interpolation process due to the need of accessing neighbors of neighbors. So, large stencil sizes can lead to large setup times, even when the operator complexity is small. Both C_{op} and the stencil sizes on all levels should ideally be independent of the problem size n . In large 3D tests using existing AMG coarsenings of RS-type [13, 7], however, it can be seen that both the operator complexity, C_{op} , and the stencil size may grow strongly, resulting in severe loss of scalability; see also section 5. For instance, the RS coarsening [13], and its parallel variants, the CLJP and Falgout coarsenings [4, 7], generally give good results in terms of convergence factors for 3D problems, but may result in large operator complexities and severe stencil size growth. It is important to note that complexity growth can be kept under control for some 3D problems. For elasticity problems, for instance, low complexities can be obtained by judiciously choosing the AMG strength parameter, which defines which matrix elements are considered in the coarse point selection process, while retaining good convergence. For some 3D problems, interpolation truncation can be quite effective to decrease operator complexities and stencil sizes.

We consider the classical RS-based coarsening strategies in more detail, in order to gain some more insight into why these complexity growth problems arise. The AMG coarsening process partitions the grid points on a given grid level into disjoint sets of coarse points (C -points), which are taken to the next level, and fine points (F -points), which are interpolated from C -points. Only matrix coefficients that are sufficiently large are considered in the coarsening process: only *strong connections* are considered. We say that a point i *strongly depends on* j or j *strongly influences* i if

$$(3.1) \quad -a_{ij} \geq \alpha \max_{k \neq i} (-a_{ik}),$$

where the *strength threshold* α is a positive constant smaller than one.

In the RS, Falgout, and CLJP coarsenings [13, 4, 7], the following two heuristics are imposed:

- H1: For each point j that strongly influences an F -point i , j is either a C -point, or strongly depends on a C -point k that also strongly influences i .
- H2: The set of C -points needs to form a maximal independent set in the reduced graph of the matrix (with only strong connections retained in the graph).

H1 implies that two strongly connected F -points are interpolated by a common C -point. In RS, Falgout, and CLJP coarsening, H1 is strongly enforced, while H2 is imposed where possible. Heuristic H1 assures that all strong connections can be taken into account adequately when interpolation formulas are constructed, generally resulting in a solver with constant AMG convergence factors bounded away from one. However, H1 is imposed by choosing additional C points, and may thus potentially lead to higher complexities. Heuristic H2 intends to provide enough C -points (maximal set), but not too many (independent set) for efficient and accurate interpolation.

The RS coarsening [13] is a sequential realization of the above described coarsening principles. Each point i is assigned a measure λ_i , which equals the number of points that are strongly influenced by i . Then a point with a maximal λ_i (there are usually several) is selected to be the first C -point. Now all points that strongly depend on i become F -points. For each point j that strongly influences one of these new F -points, λ_j is increased to improve j 's chances of becoming a new C -point. This highly sequential process is repeated until all points are either F - or C -points. Heuristic H1 is enforced in a second pass over the grid by adding C -points. The resulting AMG algorithm works optimally for many highly structured problems, because the structure is preserved on coarse grids due to the systematic marching, and tends to perform nearly optimally for more unstructured problems: scalable complexities and convergence factors are obtained for a wide range of problems [5, 1, 13, 14].

The same heuristics are used as a guideline for the parallel CLJP coarsening [4, 7]. In CLJP coarsening, the sequential marching strategy from RS for choosing candidate C points is avoided by adding a random number between 0 and 1 to each measure, thus making each point distinct. Now the points with local maximal measure can be selected to be C -points leading to a completely parallel process. This algorithm retains much of the algorithmic scalability, while adding good parallel scalability for a large class of problems [7]. Due to the random nature of the coarsening, fine-grid structure is not preserved on coarse grids, and as a result, for highly structured problems, complexity and convergence may degrade compared to RS coarsening. However, the CLJP coarsening has the important advantage of being parallel, and tests show that it may actually perform somewhat better for unstructured problems than parallel RS implementations, which use the RS coarsening in the interior of each processor, but have different ways of handling boundary points. Examples of parallel RS coarsenings are the RS3 coarsening, which deals with processor boundaries via a third pass applied to the processor boundaries, or the Falgout coarsening, which is a hybrid form of RS and CLJP coarsening [7]. It turns out that the Falgout coarsening is often the most efficient in a parallel environment for problems with a regular fine-grid structure. In general, RS, CLJP, and Falgout coarsening all provide good results in terms of convergence factors, but severe complexity growth may arise, especially for 3D problems. As will be shown in numerical experiments below, the growth in C_{op} and stencil size hampers scalability for large problems on machines with large processor numbers. This loss of scalability may already become an issue on serial machines, but for larger problem sizes on parallel machines the effects are more severe, with

additional degradation due to increased communication.

The discussion above has illustrated that, in general, a larger set of C -points tends to lead to more scalable convergence factors because more accurate interpolation formulas can be constructed, but, at the same time, adding C -points can significantly increase the complexity of the method.

In 2D, imposing heuristic H1 is important for obtaining an efficient method with scalable convergence factors and good complexity. However, it appears that in 3D, the balance may be different: imposing heuristic H1 may be too strong a requirement, resulting in prohibitive complexity growth. In this paper, we look for an improved balance between AMG complexity and convergence factors for 3D problems by considering more sparse coarse grids while trying to retain acceptable convergence factors. We present new, simplified coarsening algorithms that are based on enforcing heuristic H2, while mostly ignoring heuristic H1. In fact, we replace heuristic H1 by the following much less stringent requirement [11] for our new coarsening algorithms:

H1': Each F -point needs to strongly depend on at least one C -point.

4. Two new parallel AMG variants. In this section we give a detailed description of the newly proposed PMIS and hybrid HMIS coarsening algorithms, followed by a brief description of the interpolation formula used in our parallel AMG preconditioners.

4.1. The PMIS coarsening algorithm. Given a matrix problem $Au = f$, define V as the set of unknowns (or nodes) of A . Define the auxiliary strength matrix S as follows:

$$(4.1) \quad S_{ij} = \begin{cases} 1 & \text{if } i \neq j \text{ and } -a_{i,j} \geq \alpha \max_{k \neq i} (-a_{ik}), \\ 0 & \text{otherwise} \end{cases},$$

i.e., $S_{ij} = 1$ only if i strongly depends on j . Here α is the strength threshold which is usually chosen between 0.25 and 0.50 [13]. The i th row of S defines S_i , the set of nodes that influence i , while the i th column of S defines S_i^T , the set of nodes that are influenced by i . That is, $S_i = \{j \in V \mid S_{ij} = 1\}$ and $S_i^T = \{j \in V \mid S_{ji} = 1\}$. Define $G = (V, E) = g(S)$ as the undirected graph of S , where V is now the vertex set of the graph, and E the edge set: $E = \{\{i, j\} \in V \times V \mid S_{ij} = 1 \text{ or } S_{ji} = 1\}$. Given an undirected graph $G = (V, E)$, $I \subset V$ is an *independent set* of V in G iff $\forall i, j \in I : \{i, j\} \notin E$. $I \subset V$ is a *maximal independent set* of V in G iff I is an independent set of V in G , and $\forall j \in V \setminus I, I \cup \{j\}$ is not an independent set of V in G . Given an undirected graph $G = (V, E) = g(S)$ of a strength matrix S , $G' = (V', E') = g(V', S)$ is the subgraph of G induced by vertex set $V' \subset V$ iff $E' = \{\{i, j\} \in V' \times V' \mid \{i, j\} \in E\}$.

A formal description of the PMIS algorithm can be given as follows.

The PMIS algorithm:

Given $G = (V, E) = g(S)$, define weights $w(i) \forall i \in V$:

$w(i) = \#S_i^T + \text{Rand}([0, 1])$.

The initial set of F -points $F = \{i \in V \mid \#S_i^T = 0\}$.

The initial set of C -points $C = \emptyset$.

Take the F -points out of the remaining vertex set $V' = V \setminus F$.

The subgraph induced by the remaining vertex set is then $G' = g(V', S)$.

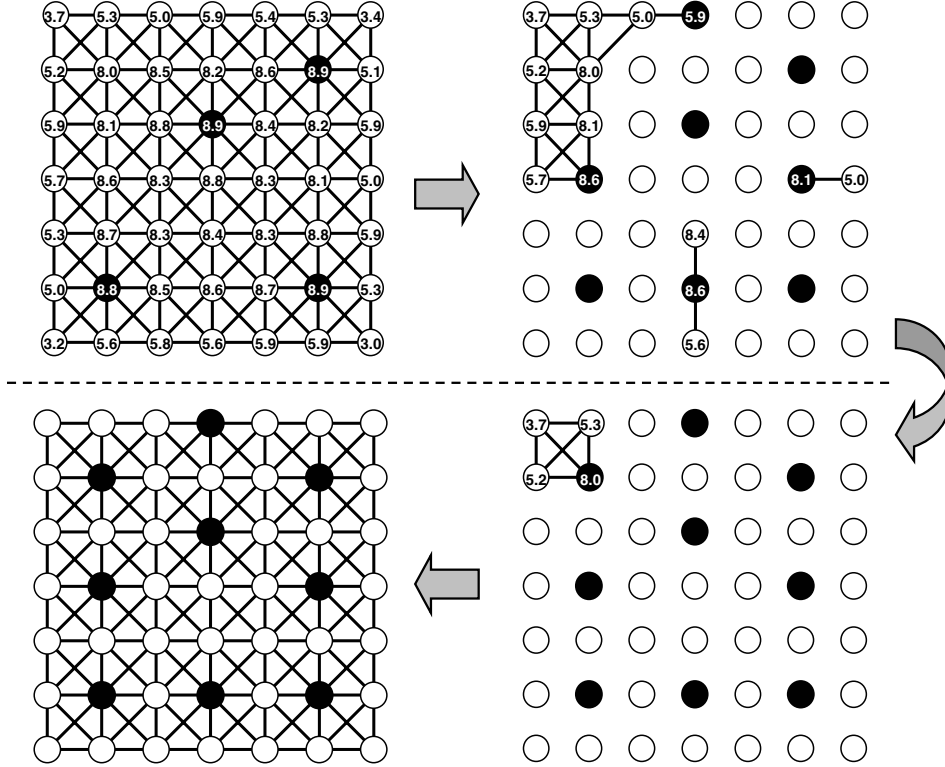


FIG. 4.1. PMIS coarsening performed for a 2D 9-point Laplace operator on a 7×7 grid. (Original grid with measures and first independent set (black points) on top left, application of one PMIS iteration, i.e., elimination of generated C- and F-points from graph combined with generation of next independent set on top right, followed by another PMIS iteration on bottom right, final coarsening on bottom left.)

While $V' \neq \emptyset$ do:

Choose an independent set I of V' in G' :

$i \in I \iff w(i) > w(j) \quad \forall j \text{ with } \{i, j\} \in E'$.

Make all elements of I C-points: $C = C \cup I$.

Make all elements of $V' \setminus I$ that are strongly influenced by a new C-point,

F-points: $F = F \cup F_{new}$, with $F_{new} = \{j \in V' \setminus I \mid \exists i \in I : i \in S_j\}$.

Remove all new C- and F-points from V' : $V' = V' \setminus \{I \cup F_{new}\}$.

The remaining subgraph is then $G' = g(V', S)$.

Enddo.

The PMIS algorithm as described above is basically Luby's parallel maximal independent set algorithm [12], with three small modifications. An illustration of the algorithm applied to a 2D 9-point Laplace operator on a 7×7 grid is given in Figure 4.1. The different steps in the algorithm can briefly be described as follows. First, every point i in V is given a measure, $w(i)$, which is the sum of the number of points that i influences, $\#S_i^T$ (8 for all interior points in the example, 5 on the boundaries and 3 on the corners), and a random number between 0 and 1. The measure $w(i)$ is a measure of point i 's eligibility to be a C-point. Points that influence many other points are likely to be good candidates for becoming C-points, since many strongly

connected points would be able to interpolate from them, and thus they receive a large measure $w(i)$. The random number is added to break ties when the numbers of influenced points of neighboring points are equal. In Luby's original algorithm, only random numbers are used to select the independent set, so adding $\#S_i^T$ is the first modification to Luby's algorithm. In every step of the ensuing procedure, all points in V that have not yet been assigned as C - or F -points become C -points (black points in Figure 4.1) if their measure is larger than those of their unassigned strongly connected neighbors. In Luby's original algorithm, all unassigned points that are strongly connected to new C -points become F -points, and the process repeats with the selection of new C -points as above. After a finite, typically small number of iterations, the grid is fully partitioned into C - and F -points. The resulting set of C -points forms a maximal independent set in the undirected graph of the auxiliary strength matrix S . The set is independent, because for every new C -point, all its neighbors are F -points by construction, and the set is maximal, because turning any F -point into a C -point would violate the independence, as every new F -point is the neighbor of a C -point by construction.

In order to make the Luby algorithm useful for AMG coarsening, two additional small modifications have to be made. First, initially all points i of V that do not influence any other point of V are made F -points because they would not be useful for interpolation as C -points. This also precludes that isolated points stay C -points on all levels. Because of this modification, the set of C -points determined by the PMIS algorithm may not strictly be maximal (because isolated points can be made C -points without violating the independence of the set). Second, directionality of strong connections is taken into account (heuristic H1'): when a new set of C -points is determined, only unassigned neighbors that are strongly influenced by a new C -point are made F -points. This assures that all F -points (except for, possibly, the initially assigned F -points) are influenced by at least one C -point, from which they will be able to interpolate. Because of this modification, the set of C -points determined by the PMIS algorithm may not be strictly independent (because points in C may be strongly connected).

This algorithm is inherently parallel, due to the random nature of C -point selection. Parallelization is straightforward, and only requires communication of boundary information between neighboring processors. Note also that if we guarantee that each point has a fixed random number independent of the number of processors and the distribution of points across processors, the algorithm will generate the same coarsening independent of the number of processors and the distribution of points across processors. This algorithm can also be seen as a simplified version of the CLJP coarsening algorithm [4, 7], which is also derived from Luby's parallel independent set algorithm with additional operations that guarantee heuristic H1 of section 3.

4.2. The HMIS coarsening algorithm. Inspired by the hybridization concept of Falgout coarsening [7], we decided to use the same idea for a new hybrid coarsening algorithm that does not enforce heuristic H1. Falgout coarsening proceeds by first using the classical RS coarsening independently on each processor. Then all the C -points that have been generated in the RS coarsening process and that are located in the interior of each processor are combined in a set, which is used as the first set of C -points (note that this might not be an independent set) for the CLJP-coarsening strategy. The treatment of the boundary points is handled completely by the CLJP coarsening scheme.

Now, since we are pursuing a coarsening strategy that does not enforce heuristic

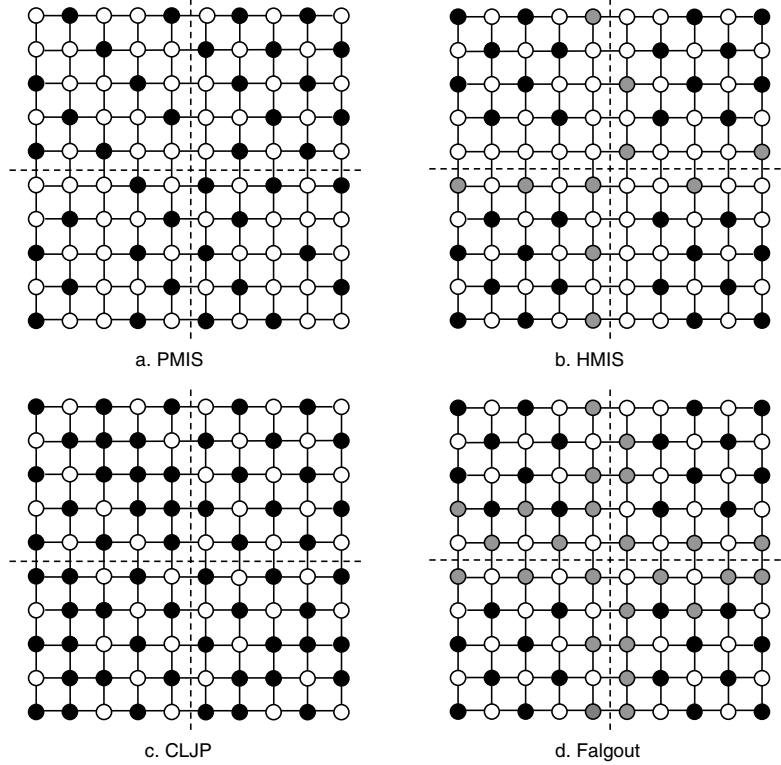


FIG. 4.2. Various coarsenings applied to a 2D 5-point Laplace operator on a 10×10 -grid across 4 processors.

H1, we will use only the first pass of the RS coarsening scheme on each processor independently as our first part of the hybrid algorithm. When this step has been completed, we use all the interior C -points, i.e., those C -points that are not located on the processor boundaries, as the first independent set of C -points to be fed into the PMIS algorithm. We call the resulting new coarsening scheme the HMIS coarsening scheme, since it is a hybrid algorithm which combines one-pass RS coarsening inside the processor interiors with PMIS coarsening near the processor boundaries (each of which ignore heuristic H1, but impose H1'). An example of the HMIS coarsening applied to the 2D 5-point Laplacian on a 10×10 -grid distributed across 4 processors is illustrated in Figure 4.2b, where the black points denote C -points that were generated in the first phase of the HMIS coarsening, when one-pass RS coarsening is applied independently on each processor, and the grey points are C -points that were added in the second phase, when applying PMIS coarsening.

The HMIS coarsening scheme will be compared in the following sections with the CLJP, Falgout, and PMIS coarsenings. Due to previous experiences with CLJP and Falgout coarsenings [7], we anticipate that for problems with regular fine-grid structure, the HMIS algorithm may perform better than the PMIS algorithm, due to the preservation of structure within the processor interiors. Figure 4.2 illustrates the four coarsenings that we will compare applied to a 2D 5-point Laplace operator distributed across 4 processors; white points denote F -points, whereas black and grey points denote C -points. Grey C -points are those that have been generated in the

second phase of the hybrid algorithms. This small example already shows that the new coarsenings will lead to much coarser grids, due to relaxing condition H1.

4.3. Interpolation. In classical AMG, the interpolation of the error at the F -point i takes the form

$$(4.2) \quad e_i = \sum_{j \in C_i} w_{i,j} e_j,$$

where $w_{i,j}$ is an interpolation weight determining the contribution of the value e_j in the final value e_i , and C_i is the subset of C -points whose values will be used to interpolate a value at i . In most classical approaches to AMG interpolation, C_i is a subset of the nearest neighbors of grid point i , and longer-range interpolation is normally not considered.

The points, to which i is connected, comprise three sets: the set C_i , the set D_i^s of points that strongly influence i but are not coarse interpolatory points, and the set D_i^w of points connected to, but not strongly influencing, i . Based on assumptions on small residuals for smooth error [1, 13, 14, 2], the following formula can be derived for the interpolation weights:

$$(4.3) \quad w_{i,j} = -\frac{1}{a_{i,i} + \sum_{k \in D_i^w} a_{i,k}} \left(a_{i,j} + \sum_{k \in D_i^s} \frac{a_{i,k} a_{k,j}}{\sum_{m \in C_i} a_{k,m}} \right).$$

However, because on our coarse grids heuristic H1 is not strongly imposed and two strongly connected F -points i and k may not have a common C -point m , it may happen that some of the terms $\sum_{m \in C_i} a_{k,m}$ in (4.3) vanish. We have modified interpolation formula (4.3) such that in case $\sum_{m \in C_i} a_{k,m} = 0$, $a_{i,k}$ is added to the so-called diagonal term (the term $a_{i,i} + \sum_{k \in D_i^w} a_{i,k}$ in (4.3)), i.e., a strongly influencing neighbor point k of i that is an F -point is treated like a weak connection of i if it does not share a common neighbor that is a C -point with i . We denote by F_i the set of strongly connected neighbors of i that are F -points and do not share a common C -point. Although this does not guarantee that the interpolation formula will be sufficiently accurate (see the numerical convergence results below and the ensuing discussion for an assessment of this question), it does guarantee that the interpolation formula remains well defined, and that constant error is interpolated accurately. Our interpolation is further modified as proposed in [7] to avoid extremely large interpolation weights that can lead to divergence. This leads to the following interpolation formula:

$$(4.4) \quad w_{i,j} = -\frac{1}{a_{i,i} + \sum_{k \in D_i^w \cup F_i} a_{i,k}} \left(a_{i,j} + \sum_{k \in D_i^s \setminus F_i} \frac{a_{i,k} \hat{a}_{k,j}}{\sum_{m \in C_i} \hat{a}_{k,m}} \right),$$

where

$$\hat{a}_{i,j} = \begin{cases} 0 & \text{if } \text{sign}(a_{i,j}) = \text{sign}(a_{i,i}), \\ a_{i,j} & \text{otherwise.} \end{cases}$$

5. Numerical results: Scalar elliptic PDEs. This section contains an extensive numerical scalability study of the PMIS and HMIS coarsenings as compared with the existing CLJP and Falgout coarsenings. These results were obtained on Lawrence Livermore National Laboratory’s supercomputer MCR. MCR is a tightly coupled Intel Linux cluster with 1152 nodes. Each node consists of 2 Xeon processors with a CPU speed of 2.4 GHz and has 4 GBytes of memory that are shared between its two processors. For parallel communication MPI was used. The strength threshold was 0.25 for all PMIS and HMIS runs. For CLJP and Falgout, the strength thresholds used were the values determined to be optimal with regard to total time for the various test cases in earlier work [7], which were in general $\alpha = 0.5$ for 3D test runs and $\alpha = 0.25$ for 2D test runs, unless noted otherwise. A large class of scalar elliptic PDE problems on structured and unstructured grids is considered. For all tests, the AMG solver is accelerated with a GMRES(10) Krylov solver. As a smoother, one sweep of hybrid CF-Gauss–Seidel (first all the C -points, then all the F -points are relaxed) was used on the down cycle and one sweep of hybrid FC-Gauss–Seidel on the up cycle. A hybrid smoother is a smoother that uses a certain method, in this case CF-Gauss–Seidel on each processor, but updates in a Jacobi-like fashion across processors. For more information on hybrid smoothers, see [17]. Also, we set $R^k = (P^k)^T$. We chose a random right-hand side, and $u^1 = 0$. Also iterations were stopped, when the 2-norm of the relative residual was smaller than 10^{-8} . It should be noted that all considered coarsening algorithms behave well as the grids become increasingly coarse; processors can just “shut down” when no points remain to be coarsened on their domains without any extra communication and special treatment. When the coarse grid is smaller than 10 grid points, the remaining processors (at most 9) each receive the final linear system and solve it via Gaussian elimination.

Most test cases were discussed in detail in [7] and will only be described briefly here.

The legend for the tables in all subsections is as follows:

- p : number of processors
- C_{op} : operator complexity
- $\#lev$: total number of levels in the AMG V-cycle
- s_{avg} : maximal average stencil size (average number of nonzeros per matrix row, largest value that occurs on any of the V-cycle levels)
- l_{max} : on which V-cycle level this maximum s_{avg} occurs
- t_{setup} : AMG setup time
- t_{solve} : AMG solve time
- its : number of iterations to convergence
- t_{tot} : total time (AMG setup + solve time)
- dof : total number of degrees of freedom.

5.1. 3D Laplacian problems. In Table 5.1, we show the results of parallel scaling tests for a 7-point standard finite difference discretization of Laplace’s equation

$$(5.1) \quad -\Delta u = f$$

on a cubic regular domain, with problem size per processor $40 \times 40 \times 40$ points. These results immediately illustrate the complexity growth problems that affect the CLJP and, to a somewhat lesser extent, the Falgout coarsening. Operator complexities, total numbers of levels, and stencil sizes are excessively high, and setup times are much larger than the solution phase execution times. Interestingly enough, while CLJP has extremely large operator complexities, its average stencil sizes are significantly smaller

TABLE 5.1

AMG-GMRES(10) with different coarsenings applied to the 7-point Laplacian, 40^3 degrees of freedom (dof) per processor.

Method	p	C_{op}	$\#lev$	s_{avg}	l_{max}	t_{setup}	t_{solve}	its	t_{tot}
CLJP	1	14.39	15	157.9	6	1.88	1.47	6	3.35
	8	15.75	17	206.3	9	6.03	4.23	8	10.26
	64	16.59	20	260.1	11	13.30	7.69	9	20.99
	512	17.02	22	307.0	12	22.33	13.50	10	35.83
	1000	17.15	23	316.3	12	27.16	15.03	10	42.19
	1331	17.19	23	321.4	12	29.57	16.68	10	46.25
Falgout	1	3.61	11	103.6	5	0.77	0.42	5	1.19
	8	4.43	13	282.4	8	3.24	1.33	6	5.57
	64	5.07	15	451.5	9	11.77	3.15	6	14.93
	512	5.45	18	575.4	9	23.19	7.26	7	30.45
	1000	5.51	19	598.4	9	26.64	8.59	7	35.23
	1331	5.55	19	608.3	9	30.57	8.99	7	39.56
PMIS	1	2.32	7	55.4	4	0.41	0.87	13	1.28
	8	2.35	8	63.2	4	1.15	2.56	17	3.71
	64	2.36	9	68.1	4	2.81	4.86	21	7.67
	512	2.37	10	70.7	4	5.04	7.73	25	12.77
	1000	2.37	10	71.1	4	6.63	8.39	26	15.02
	1331	2.37	10	71.3	4	8.28	9.71	28	17.99
HMIS	1	2.79	7	57.1	5	0.73	0.38	5	1.11
	8	2.80	8	76.3	5	1.45	1.74	10	3.19
	64	2.80	9	79.3	5	2.62	3.23	12	5.85
	512	2.80	10	82.2	5	4.89	5.03	15	9.92
	1000	2.80	10	83.6	5	6.18	5.10	15	11.28
	1331	2.80	10	83.4	5	7.56	5.62	16	13.18

than those of the Falgout coarsening. A closer examination of the two coarsenings reveals that Falgout coarsening initially coarsens much faster than CLJP. Stencil sizes grow faster than for CLJP (caused by somewhat larger interpolation operator stencil sizes), peak in the intermediate levels at greatly reduced grid sizes, and decrease fairly fast in the lowest levels. CLJP coarsens fairly slowly at the beginning, and its stencil sizes decrease gradually, stay close to the maximal size over various levels, and decrease slower than for Falgout in the lowest levels.

PMIS and HMIS, however, are two to three times as fast as CLJP and Falgout and have operator complexities of less than half the size and much smaller stencil sizes, leading to significantly smaller setup times. They require more iterations, and additionally the results show that while for CLJP and Falgout the number of iterations are almost constant, the number of iterations are increasing somewhat for HMIS and more significantly for PMIS. The worse convergence for PMIS is not quite unexpected, since it is a more aggressive scheme than HMIS. HMIS, which is based on one pass of the RS algorithm, generates more regular coarsenings than PMIS due to its use of random numbers, which are better suited for this structured problem. However, execution time per iteration is much lower due to a smaller operator complexity, resulting in overall time savings also for the solution phase. HMIS is slightly faster with a slightly larger operator complexity than PMIS for this highly structured problem.

In Table 5.2 we show scaling results for the 7-point Laplacian problem with a large problem size per processor. CLJP results were not obtained because the MCR machine quickly ran out of memory for this coarsening scheme. This is a structured problem, and therefore the Falgout method, which is capable of keeping the structure on coarser grids away from subdomain problems, does not suffer as much from complexity problems at the total scale of the problem considered. The PMIS and HMIS

TABLE 5.2

AMG-GMRES(10) with various coarsenings applied to the 7-point Laplacian, 100³ dof per processor.

Method	p	C_{op}	$\#lev$	s_{avg}	l_{max}	t_{setup}	t_{solve}	its	t_{tot}
Falgout	8	4.23	17	501.6	10	49.56	19.15	6	68.71
	512	4.69	21	750	10	133.03	33.73	8	166.76
	1331	4.74	23	773.9	10	139.48	32.64	7	172.12
PMIS	8	2.59	9	73.2	5	21.33	55.20	22	76.53
	512	2.61	12	79.4	5	36.47	123.00	44	159.47
	1331	2.61	12	79.7	5	41.26	131.21	46	172.47
HMIS	8	2.89	9	141	5	44.48	32.17	11	76.65
	512	2.88	11	131.3	5	103.37	58.01	19	161.38
	1331	2.88	12	130.4	5	114.58	50.26	16	164.84

methods do not have much room to improve the results, and thus the PMIS/HMIS timings are very similar to those of Falgout for this problem with a large ratio of processor interior degrees of freedom (dof) to processor boundary dof. Additionally, convergence is worse. This is particularly notable for PMIS. However, the operator complexities of PMIS and HMIS are only about half as large as the one of Falgout coarsening, which is of benefit on architectures with smaller memory capacities, or when the rest of the application requires a large amount of the memory, which is often the case.

We also investigate AMG scalability for a 3D unstructured grid problem, since previous experiments with CLJP and Falgout have shown that CLJP performs better on unstructured grids [7]. A finite element discretization of the Laplace equation on the unit cube is applied to a grid with approximately 20,000 dof per processor; tetrahedra were used for the discretization of the domain. Table 5.3 shows that PMIS and HMIS perform better than CLJP and Falgout. PMIS/HMIS are more than twice as fast as Falgout and have operator complexities that are half the size as the operator complexity of Falgout. We find that CLJP is slightly faster than Falgout and PMIS is slightly faster than HMIS. This is an unstructured problem for which the RS marching approach is not especially advantageous because there is no fine-grid structure to be kept. Interestingly, the more random nature of coarse point selection of CLJP and PMIS seems to be slightly better for unstructured problems.

TABLE 5.3

AMG-GMRES(10) with various coarsenings applied to a 3D Laplacian on an unstructured grid.

Method	p	C_{op}	s_{avg}	t_{setup}	t_{solve}	its	t_{tot}
CLJP	4	3.84	151.6	1.32	0.89	9	2.21
	32	4.47	207.0	5.07	2.76	10	7.83
	288	4.68	271.0	12.34	9.35	12	21.69
Falgout	4	4.45	165.6	1.64	1.01	9	2.65
	32	5.30	232.4	6.32	2.97	10	9.29
	288	5.53	288.4	14.01	9.55	12	23.56
PMIS	4	1.41	39.9	0.39	0.73	15	1.12
	32	1.45	49.3	0.92	2.11	18	3.03
	288	1.46	53.0	2.41	4.86	24	7.27
HMIS	4	1.48	45.4	0.47	0.75	15	1.22
	32	1.59	56.7	1.31	2.09	18	3.40
	288	1.60	60.8	2.91	4.94	23	7.85

In Figure 5.1 we present the results of parallel scaling tests for CLJP, Falgout, PMIS, and HMIS coarsening applied to a 27-point finite element discretization of

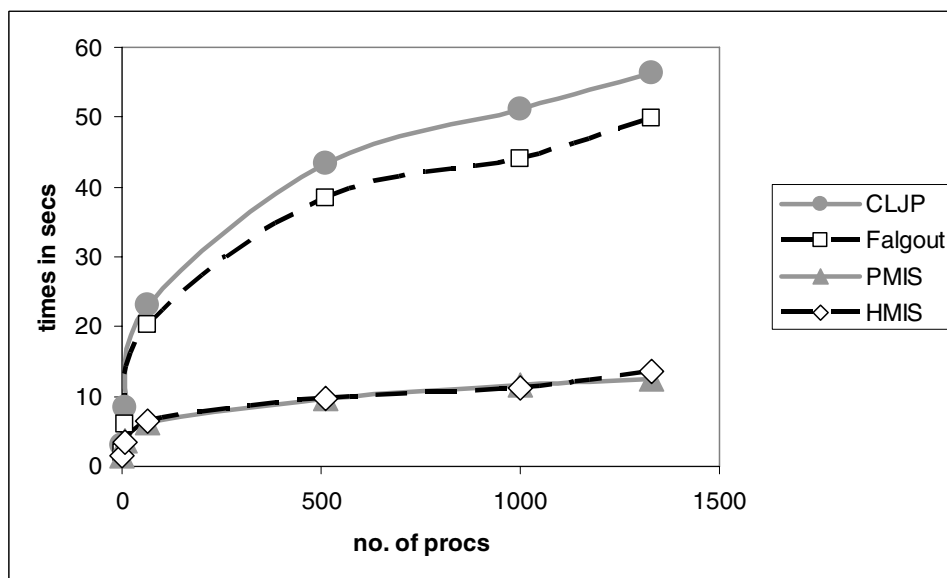


FIG. 5.1. Total times for AMG-GMRES(10) with various coarsenings applied to a 27-point operator, 40^3 degrees of freedom per processor.

Laplace's equation on a cubic regular domain. Problem size per processor is 40^3 degrees of freedom. For this problem CLJP has an operator complexity that is growing from 2.08 on one processor to 2.99 on 1331 processors; Falgout's complexities range from 1.62 to 2.48, whereas PMIS and HMIS have constant complexities of 1.11 and 1.20, respectively. Stencil sizes for CLJP range from 238.4 to 1015, and for Falgout from 123.8 to 849.0, whereas they only slightly vary for PMIS from 36.4 to 51.4 and for HMIS from 85.4 to 106.3. The stencil size, which is about 27 for this problem on the finest grid level, thus grows to an astounding order of a thousand for CLJP and Falgout coarsening. In spite of the fact that Falgout and CLJP require only about half the number of iterations as PMIS and HMIS, overall timings for HMIS and PMIS are up to four times as fast. Just as in the 7-point case, PMIS and HMIS are more than twice as fast with operator complexities of half the size as those of CLJP and Falgout.

TABLE 5.4

AMG-GMRES(10) with various coarsenings applied to a 27-point operator, 90^3 dof per processor.

Method	p	C_{op}	$\#lev$	s_{avg}	l_{max}	t_{setup}	t_{solve}	its	t_{tot}
Falgout	8	2.29	17	769.8	10	35.68	25.54	8	61.22
	512	Out of memory							
	1000	Out of memory							
PMIS	8	1.11	8	42.9	4	12.29	32.43	17	44.72
	512	1.11	10	51.7	5	17.36	55.35	25	72.71
	1000	1.11	11	53.1	6	20.02	59.41	27	79.43
HMIS	8	1.23	9	138.9	5	17.96	24.04	11	42.00
	512	1.23	10	151.3	5	24.24	47.17	20	71.41
	1000	1.23	11	152.6	5	26.58	53.82	22	80.40

In Table 5.4 scaling results for the 27-point operator are presented using $90 \times 90 \times 90$ points per processor. Again, CLJP results were not obtained because the MCR

machine quickly ran out of memory for this problem. Here, the Falgout coarsening also runs out of memory due to growing complexities when using 64 or more processors. Obviously, PMIS and HMIS here are preferable over the other coarsenings.

5.2. Anisotropic Laplacian and convection-diffusion problems. In order to test the robustness and generality of the trends observed in the previous subsection, we have also tested performance and scalability for an anisotropic problem

$$(5.2) \quad -cu_{xx} - u_{yy} - u_{zz} = f$$

with $c=0.001$ (see Table 5.5), and a convection-diffusion problem

$$(5.3) \quad -\Delta u + c(u_x + u_y + u_z) = f$$

with $c = 10q$ and the number of processors $p = q^3$, which leads to a nonsymmetric matrix (see Table 5.6). The matrices for both problems were generated using finite differences leading to a 7-point stencil. For the convection term, standard first-order upwind differences were used. For both problems the number of points per processors was 40^3 . For the anisotropic problem we chose as a strength threshold $\alpha = 0.25$ for the CLJP and Falgout coarsenings, and $\alpha = 0.75$ for the nonsymmetric problem.

TABLE 5.5
Anisotropic Laplacian, 40^3 dof per processor, 1331 processors.

Method	C_{op}	$\#lev$	s_{avg}	l_{max}	t_{setup}	t_{solve}	its	t_{tot}
CLJP	7.07	20	241.3	12	24.32	9.59	8	33.91
Falgout	3.95	19	297.2	10	31.41	7.23	6	38.64
PMIS	2.29	13	31.8	6	7.70	11.59	29	19.29
HMIS	3.60	13	35.3	7	8.90	6.99	16	15.89

For both test cases PMIS and HMIS are faster, have smaller complexities, and are more scalable than CLJP and Falgout very much in the same way as above. Interestingly enough, in Table 5.5, we find that the operator complexity of Falgout is only slightly larger than that of HMIS, but the setup time for HMIS is much lower due to a significantly smaller stencil size.

TABLE 5.6
Convection-diffusion problem, 40^3 dof per processor, 1331 processors.

Method	C_{op}	$\#lev$	s_{avg}	l_{max}	t_{setup}	t_{solve}	its	t_{tot}
CLJP	9.62	25	91.2	10	20.57	16.14	16	36.71
Falgout	5.32	22	107.7	8	17.95	9.35	12	27.30
PMIS	2.34	11	68.2	5	8.43	10.59	29	19.02
HMIS	2.89	12	77.0	5	8.75	7.16	17	15.91

5.3. 2D Laplacian problems. It is interesting to see how PMIS and HMIS perform for 2D problems. As mentioned before, 2D CLJP and Falgout coarsenings typically do not show complexity growth, so we do not expect that PMIS and HMIS can offer much improvement. Table 5.7 shows that for a 9-point 2D Laplacian finite element discretization with 250^2 dof per processor, CLJP and Falgout coarsening indeed provide scalable AMG methods, and, not contrary to expectation, PMIS and HMIS are somewhat slower with only slightly lower complexities; note the very small maximal stencil sizes for PMIS and HMIS coarsening. Scaling tests for the 2D 5-point Laplacian show very similar results and are not shown here.

TABLE 5.7

AMG-GMRES(10) with various coarsenings applied to a 9-point 2D Laplacian, 250² dof per processor.

Method	p	C_{op}	$\#lev$	s_{avg}	l_{max}	t_{setup}	t_{solve}	its	t_{tot}
CLJP	1	1.76	10	28.6	6	0.46	0.54	8	1.00
	64	1.83	14	35.2	7	1.07	1.93	10	3.00
	256	1.83	15	35.6	7	2.02	3.50	10	5.52
	1024	1.83	16	36.2	8	6.00	2.76	10	8.76
Falgout	1	1.33	7	9.0	1	0.26	0.35	7	0.61
	64	1.35	12	25.5	9	0.75	1.17	7	1.92
	256	1.35	14	28.9	9	1.56	1.62	8	3.18
	1024	1.35	15	30.9	10	5.74	2.01	8	7.75
PMIS	1	1.24	7	9.3	4	0.21	1.11	21	1.32
	64	1.24	10	10.5	5	0.46	4.30	37	4.76
	256	1.24	11	10.8	6	1.03	4.97	41	6.00
	1024	1.24	12	10.9	6	4.27	6.06	48	10.33
HMIS	1	1.33	7	9.0	1	0.24	0.35	7	0.59
	64	1.33	10	11.0	7	0.53	1.91	16	2.44
	256	1.33	11	11.4	7	1.08	2.09	17	3.17
	1024	1.33	12	11.6	7	4.34	2.60	20	6.94

We also considered the 2-dimensional rotated anisotropic problem

$$(5.4) \quad -(c^2 + \epsilon s^2)u_{xx} + 2(1 - \epsilon)scu_{xy} - (s^2 + \epsilon c^2)u_{yy} = 1$$

with $s = \sin \gamma$, $c = \cos \gamma$, and $\epsilon = 0.001$. We again used 250² dof per processor, and ran tests for the rotation angles $\gamma = 45^\circ$ and $\gamma = 60^\circ$. Results using 1024 processors are shown in Tables 5.8 and 5.9. For these test problems, the PMIS and HMIS coarsenings perform much worse than CLJP and Falgout, particularly PMIS convergence degrades. Also, note the small stencil sizes when using PMIS and HMIS coarsening. These results show that the interpolation we use is not good enough for these problems.

TABLE 5.8

2D rotated anisotropic Laplacian, $\gamma = 45^\circ$, 250² dof per processor, 1024 processors.

Method	C_{op}	$\#lev$	s_{avg}	l_{max}	t_{setup}	t_{solve}	its	t_{tot}
CLJP	2.96	21	40.8	11	8.02	3.29	9	11.31
Falgout	2.45	21	46.5	10	9.12	3.80	9	12.92
PMIS	1.91	15	11.1	7	5.26	20.47	96	25.73
HMIS	2.26	15	15.7	7	5.40	11.89	55	17.29

TABLE 5.9

2D rotated anisotropic Laplacian, $\gamma = 60^\circ$, 250² dof per processor, 1024 processors.

Method	C_{op}	$\#lev$	s_{avg}	l_{max}	t_{setup}	t_{solve}	its	t_{tot}
CLJP	4.93	20	44.9	9	8.43	11.28	29	19.71
Falgout	3.39	20	38.7	10	7.55	6.09	17	13.64
PMIS	1.75	12	12.8	3	4.21	77.62	550	81.83
HMIS	3.27	14	27.0	3	5.22	10.63	52	15.85

5.4. 3-dimensional PDEs with jumps. A similar observation as in the case of rotated anisotropies can be made for the difficult case of a 3D elliptic PDE with jumps in the coefficients. We solve the partial differential equation

$$(5.5) \quad (au_x)_x + (au_y)_y + (au_z)_z = 1$$

TABLE 5.10
3D elliptic PDE with varying coefficients, 40^3 dof per processor, 1000 processors.

Method	C_{op}	#lev	s_{avg}	l_{max}	t_{setup}	t_{solve}	its	t_{tot}
CLJP	17.00	23	306.3	12	27.47	25.01	17	52.48
Falgout	5.77	19	591.4	9	26.28	14.41	13	40.69
PMIS	2.40	10	69.2	5	6.79	205.00	686	211.79
HMIS	2.82	10	87.3	5	6.27	65.99	202	72.26

TABLE 5.11
3D elliptic PDE with varying coefficients, 60^3 on one processor.

Method	C_{op}	#lev	s_{avg}	l_{max}	t_{setup}	t_{solve}	its	t_{tot}
CLJP	15.20	15	170.2	9	9.02	10.73	11	19.75
RS	5.00	13	185.3	8	3.94	2.77	7	6.71
PMIS	2.55	7	52.9	4	2.26	18.54	70	20.80
PMIS/MP	2.57	7	55.0	4	1.84	6.87	25	8.71

with Dirichlet boundary conditions on a unit cube, where $a(x, y, z) = 1000$ on $0.1 < x, y, z < 0.9$, $a(x, y, z) = 0.01$ on $0 < x, y, z < 0.1$ and the other cubes of size $0.01 \times 0.01 \times 0.01$ that are located at the corners of the domain, and $a(x, y, z) = 1$ elsewhere. The results in Table 5.10 show that for problems with large jumps in coefficients scalability is lost.

The results in Tables 5.8, 5.9, and 5.10 suggest that, for these difficult problems, the stencil sizes resulting from PMIS and HMIS coarsening may be too small for nearest-neighbor interpolation to be sufficiently accurate, and Krylov acceleration is not effective in this case. Additional C -points or long-range interpolation may be required here. The number of iterations needed for convergence can be reduced by increasing the number of relaxation sweeps on each level of a V-cycle, but, for the 3D example, this does not improve total execution times. Good convergence can be obtained by using W-cycles (PMIS converges within 40 iterations and HMIS within 16 iterations for the example in Table 5.10 using this approach), but W-cycles are very expensive and not scalable. Adding C -points in order to better adhere to heuristic H1 would improve interpolation accuracy and convergence, but the resulting complexity growth may be prohibitive for many large applications. Another possible approach would be to consider, for this type of difficult problem, interpolation formulas with an extended range not limited to the nearest neighbors, for instance, techniques along the lines of the multipass interpolation. Multipass interpolation proceeds in several passes, using direct interpolation for points that neighbor C -points and exploiting interpolation formulas at neighboring F -points for the remaining points. For further details, see [14]. This has to be done carefully, because increasing the range of interpolation may result in stencil growth due to the *RAP* Galerkin condition. Preliminary results for a sequential run are given in Table 5.11. Here the 3D problem is solved on a $60 \times 60 \times 60$ cube and CLJP, RS, PMIS, and PMIS/MP (PMIS with multipass interpolation) coarsenings are compared. The results show that PMIS with multipass interpolation converges much faster, and interestingly enough, for this example the complexities of PMIS and PMIS/MP are comparable. For PMIS, at most two passes are needed. Note that, since this is a sequential run, we cannot make any comparison with Falgout or HMIS coarsening, and that we can choose here the “classical” RS-coarsening, which is inherently sequential and in general leads to better complexities than Falgout, since there is no processor boundary treatment. Both PMIS and multipass interpolation are completely parallel algorithms, so we expect convergence

to also be improved in the parallel case. While we originally set out to avoid the parallel implementation of long-range interpolation formulas, it appears that they are necessary to solve some problems. Improved interpolation formulas and their use on grids coarsened with the PMIS and HMIS techniques will be considered in more detail in future research.

6. Numerical results: 3D FOSLS Stokes system. In this section, we present numerical results for 3D Stokes fluid flow simulations using a FOSLS finite element discretization. These results were obtained on the 64-node Linux cluster Hemisphere (each node contains dual 2.4 GHz Intel Xeon processors and 2 GBytes of memory) at the University of Colorado at Boulder, using Jeff Heys' parallel FOSLS code *Parafos*, coupled to *hypr*'s AMG-code *BoomerAMG*.

The Stokes equations for modeling low Reynolds number ($Re \ll 1$) flow can be written as

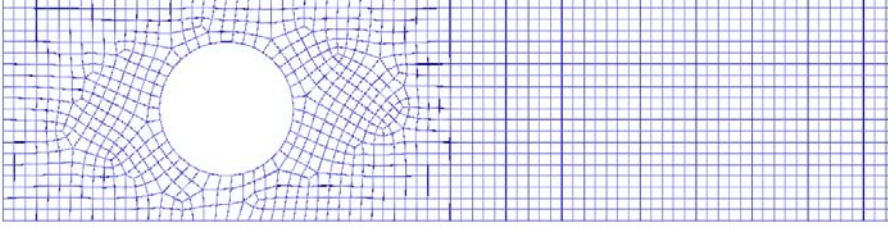
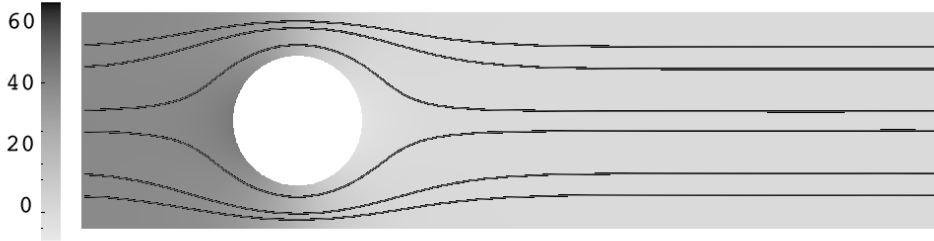
$$(6.1) \quad -\nabla q + \Delta \mathbf{v} = 0 \quad \text{in } \Omega,$$

$$(6.2) \quad \nabla \cdot \mathbf{v} = 0 \quad \text{in } \Omega,$$

where q is pressure and $\mathbf{v} = (v_1, v_2, v_3)$ is velocity. The standard Galerkin finite element method for discretizing (6.1) and (6.2) does not produce an H^1 -elliptic form, and multigrid schemes typically perform poorly. An alternative formulation is based on reformulating the Stokes equation as a first-order system and minimizing the least-squares norm of residual equations in the resulting system—the so-called FOSLS approach [3].

Scaling results are presented below for a 3D Stokes simulation of flow between two plates with a cylindrical obstacle. Figure 6.1 shows the grid and the domain for the simulation. The grid is unstructured. The finite elements used are 8 node hexahedra. The mesh is decomposed among processors using METIS [8]. Figure 6.2 shows the resulting Stokes flow as a simulation result obtained with our codes.

Table 6.1 compares the algorithmic scalability of the Falgout and PMIS coarsenings for this problem. Since the problem to be solved is a system of PDEs, the “unknown” approach was used [13]. This approach is very similar to the algorithm described in section 2; however, coarsening and interpolation are applied independently to variables that belong to the same unknown of the system of PDEs (an unknown is, e.g., velocity, pressure, etc.), and connections between variables of different unknowns are ignored. AMG was used here as a preconditioner to conjugate gradient (CG) instead of GMRES(10) as in the previous section, since FOSLS-matrices have the advantageous property of being symmetric. This also results in some memory savings as compared to GMRES(10) acceleration, because only two previous iteration vectors need to be stored. As a smoother, hybrid symmetric CF-Gauss–Seidel was used, since the use of CG requires a symmetric preconditioner, and a nonsymmetric smoother would destroy the symmetry of the preconditioner. Because this PDE system is highly coupled with 13 degrees of freedom per node, approximately 300 nonzeros are present in every matrix row on the finest grid level on average. Parallel scaling tests were performed with approximately 55,000 degrees of freedom per processor. It was necessary to use a larger strength threshold for Falgout coarsening, since complexity explodes for a small strength threshold and the problem does not fit into the memory. We report the results that were achieved for strength threshold $\alpha = 0.9$, since this led to the best performance for this coarsening. Lower α allows for better convergence, but leads to prohibitively high complexity.

FIG. 6.1. *Unstructured grid section for the 3D Stokes flow test problem.*FIG. 6.2. *3D Stokes flow test problem. Gray scales represent the pressure, and the streamlines of the fluid flow are shown.*TABLE 6.1
AMG-CG with different coarsenings.

Method	p	dof	C_{op}	$\#lev$	s_{avg}	t_{setup}	t_{solve}	its	t_{tot}
Falgout	1	55359	3.06	11	770.4	7.94	50.35	28	58.28
	8	422463	2.73	13	942.7	168.32	132.93	48	301.25
	64	3224991	3.10	16	1079.0	1280.42	689.74	91	1970.16
PMIS	1	55359	1.34	6	485.9	7.11	18.35	21	25.46
	8	422463	1.37	7	598.6	53.95	43.38	34	97.33
	64	3224991	1.40	8	733.1	333.16	125.86	52	459.02

The table confirms, for this PDE system, the results that were shown above for scalar PDEs. The PMIS coarsening has operator complexities half the size of those of Falgout, the stencil size grows substantially less, and both setup times and solve times are about four times faster for PMIS than for Falgout coarsening. The PMIS results are also more scalable than the Falgout results.

7. Conclusions. In this paper, simple coarsening schemes for parallel AMG were proposed, which are based solely on enforcing a maximum independent set property, resulting in coarse grids that are sparser than the grids obtained by adhering to the original RS coarsening heuristics. The new PMIS and HMIS coarsening techniques improve memory and execution time complexity growth for large 3D problems, and,

combined with Krylov acceleration, the resulting AMG-Krylov methods also tend to perform fairly well in terms of the number of iterations required for convergence. Numerical results were presented showing that for an extensive class of large 3D problems on parallel machines in the several 1,000-processor class, the newly proposed methods have significantly smaller complexities and require less than half the computer time of AMG methods that use an RS type coarsening. Also, for the problem sizes and machines tested, scalability with regard to memory and execution time (specifically AMG setup times) is often better than for existing approaches. Numerical results were obtained for various scalar and system problems on both structured and unstructured grids. As could be expected, when complexities are decreased, the number of iterations obtained for the new coarsening schemes were larger than those of the old ones. However, total times improved by factors of two or more for the new coarsening schemes for many 3D problems, which included diffusion problems, anisotropic problems, and nonsymmetric diffusion-convection problems, and complexities were significantly lower.

Nevertheless, for 2D problems, the new schemes did not perform as well as the old in terms of times and convergence. Also, for some difficult problems, including rotated anisotropic problems and problems with large jumps in coefficients, standard AMG interpolation that relies only on nearest neighbors for interpolation is not sufficiently accurate on the coarse grids that result from the new coarsenings, and Krylov acceleration turns out not to be effective in this case. For these cases, convergence might be improved using long-range interpolation formulas such as multipass interpolation proposed by Stueben [14]. The use of improved interpolation formulas on grids coarsened with the PMIS and HMIS techniques are the subject of future research.

REFERENCES

- [1] A. BRANDT, S. F. MCCORMICK, AND J. W. RUGE, *Algebraic Multigrid (AMG) for Automatic Multigrid Solutions with Application to Geodetic Computations*, Report, Inst. for Computational Studies, Fort Collins, CO, 1982.
- [2] W. L. BRIGGS, V. E. HENSON, AND S. F. MCCORMICK, *A Multigrid Tutorial*, 2nd ed., SIAM, Philadelphia, 2000.
- [3] Z. CAI, T. A. MANTEUFFEL, AND S. F. MCCORMICK, *First-order system least squares for second-order partial differential equations: Part II*, SIAM J. Numer. Anal., 34 (1997), pp. 425–454.
- [4] A. J. CLEARY, R. D. FALGOUT, V. E. HENSON, AND J. E. JONES, *Coarse-grid selection for parallel algebraic multigrid*, in Proceedings of the Fifth International Symposium on Solving Irregularly Structured Problems in Parallel, Berkeley, CA, Springer-Verlag, Berlin, 1998.
- [5] A. J. CLEARY, R. D. FALGOUT, V. E. HENSON, J. E. JONES, T. A. MANTEUFFEL, S. F. MCCORMICK, G. N. MIRANDA, AND J. W. RUGE, *Robustness and scalability of algebraic multigrid*, SIAM J. Sci. Comput., 21 (2000), pp. 1886–1908.
- [6] M. GRIEBEL, B. METSCH, D. OELTZ, AND A. SCHWEITZER, *Coarse grid classification: A parallel coarsening scheme for algebraic multigrid methods*, SIAM J. Sci. Comput., submitted.
- [7] V. E. HENSON AND U. M. YANG, *BoomerAMG: A parallel algebraic multigrid solver and preconditioner*, Appl. Numer. Math., 41 (2002), pp. 155–177.
- [8] G. KARYPIS, *METIS*, <http://www-users.cs.umn.edu/~karypis/metis/>.
- [9] M. T. JONES AND P. E. PLASSMAN, *A parallel graph coloring heuristic*, SIAM J. Sci. Comput., 14 (1993), pp. 654–669.
- [10] W. JOUBERT AND J. CULLUM, *Scalable Algebraic Multigrid on 3500 Processors*, Los Alamos National Laboratory Technical report no. LAUR03-568, Electron. Trans. Numer. Anal., submitted.
- [11] A. KRECHEL AND K. STÜBEN, *Parallel algebraic multigrid based on subdomain blocking*, Parallel Comput., 27 (2001), pp. 1009–1031.
- [12] M. LUBY, *A simple parallel algorithm for the maximal independent set problem*, SIAM J. Comput., 15 (1986), pp. 1036–1053.

- [13] J. W. RUGE AND K. STÜBEN, *Algebraic multigrid*, in Multigrid Methods, Frontiers Appl. Math. 3, S. F. McCormick, ed., SIAM, Philadelphia, 1987, pp. 73–130.
- [14] K. STÜBEN, *Algebraic multigrid (AMG): An introduction with applications*, in Multigrid, U. Trottenberg, C. Oosterlee, and A. Schüller, eds., Academic Press, San Diego, 2000.
- [15] U. TROTTEMBERG, C. OOSTERLEE, AND A. SCHÜLLER, *Multigrid*, Academic Press, San Diego, 2001.
- [16] R. TUMINARO AND C. TONG, *Parallel smoothed aggregation multigrid: Aggregation strategies on massively parallel machines*, in Proceedings of the 2000 ACM/IEEE Conference on Supercomputing, Dallas, 2000.
- [17] U. M. YANG, *On the use of relaxation parameters in hybrid smoothers*, Numer. Linear Algebra Appl., 11 (2004), pp. 155–172.
- [18] U. M. YANG, *Parallel algebraic multigrid methods—high performance preconditioners*, in Numerical Solution of Partial Differential Equations on Parallel Computers, A. Bruaset and A. Tveito, eds., Springer-Verlag, New York, 2006.

MULTIPLE VECTOR PRESERVING INTERPOLATION MAPPINGS IN ALGEBRAIC MULTIGRID*

PANAYOT S. VASSILEVSKI[†] AND LUDMIL T. ZIKATANOV[‡]

Abstract. We propose algorithms for the construction of AMG (algebraic multigrid) interpolation mappings such that the resulting coarse space will span (locally and globally) any number of a priori given set of vectors. Specific constructions in the case of element agglomeration AMG methods are given. Some numerical illustrations are also provided.

Key words. multiple vector preserving interpolation, algebraic multigrid, element agglomeration AMG

AMS subject classifications. 65F10, 65N20, 65N30

DOI. 10.1137/040618205

1. Introduction. We are given a symmetric positive definite (spd) sparse matrix $A : \mathbb{R}^n \mapsto \mathbb{R}^n$, which we assume corresponds to a finite element discretization of a second order elliptic PDE. No element (or grid) hierarchy is assumed, so we can handle general unstructured meshes.

Our goal is to construct two-grid, and by recursion, multigrid methods (or preconditioners) with certain vector-preserving interpolation properties that will enable us to more effectively solve systems of equations with the matrix A . A typical two-grid method involves a smoother M (for example, Gauss–Seidel) and a coarse-grid correction. The latter gives rise to a projection matrix $\pi_A = P(P^T A P)^{-1} P^T A$, based on an interpolation matrix $P : \mathbb{R}^{n_c} \mapsto \mathbb{R}^n$, $n_c < n$. Thus, a two-grid error iteration matrix E_{TG} takes the form

$$E_{TG} = (I - M^{-T} A)(I - \pi_A)(I - M^{-1} A).$$

Alternatively, one can define a two-grid preconditioner

$$B_{TG}^{-1} = [I, P] \hat{B}_{TG}^{-1} [I, P]^T$$

based on the block-factored matrix

$$\hat{B}_{TG} = \begin{bmatrix} M & 0 \\ P^T A & I \end{bmatrix} \begin{bmatrix} \hat{D}^{-1} & 0 \\ 0 & P^T A P \end{bmatrix} \begin{bmatrix} M^T & A P \\ 0 & I \end{bmatrix}, \quad \hat{D} = M + M^T - A.$$

By a straightforward computation one can see that $E_{TG} = I - B_{TG}^{-1} A$.

In this paper we will focus on the construction of the interpolation mapping P . In doing that, we will often use the finite element notion of “degrees of freedom” (dofs),

*Received by the editors November 3, 2004; accepted for publication (in revised form) by W.-P. Tang November 22, 2005; published electronically March 17, 2006. This work was performed under the auspices of the U.S. Department of Energy by University of California Lawrence Livermore National Laboratory under contract W-7405-Eng-48.

<http://www.siam.org/journals/simax/27-4/61820.html>

[†]Center for Applied Scientific Computing, University of California Lawrence Livermore National Laboratory, P.O. Box 808, Livermore, CA 94551 (panayot@llnl.gov).

[‡]Department of Mathematics, The Pennsylvania State University, University Park, PA 16802 (ludmil@psu.edu). This author was partially supported by University of California Lawrence Livermore National Laboratory under contract B551021 and by NSF under grant DMS-0511800.

which is equivalent to row and column indices of A . The row indices of P are also referred to as “fine” (or fine-grid) dofs, whereas the column indices are referred to as “coarse” (or coarse-grid) dofs. In most cases (and in the case that we consider here in greater detail) there is a natural embedding of the coarse dofs into the fine-grid dofs, in the sense that the coarse dofs $\mathcal{N}_c$ can be viewed as a subset of the set of fine-grid dofs $\mathcal{N}$. Then a common form of P is given by

$$(1.1) \quad P = \left[\begin{array}{c|c} W & \\ \hline I & \end{array} \right] \begin{array}{l} \} \\ \} \end{array} \begin{array}{l} \mathcal{N} \setminus \mathcal{N}_c \\ \mathcal{N}_c \end{array}.$$

A main example that we will focus on throughout this paper is the case of matrices A , coming from finite element discretization of second order elliptic PDEs. For such matrices a classical choice of P is that P has a row-sum of ones, that is, P interpolates constants exactly. Written in matrix form, this means that $P\mathbf{1}_c = \mathbf{1}$, where $\mathbf{1}_c =$

$$\begin{bmatrix} 1 \\ \vdots \\ 1 \end{bmatrix} \in \mathbb{R}^{n_c} \text{ and } \mathbf{1} = \begin{bmatrix} 1 \\ \vdots \\ 1 \end{bmatrix} \in \mathbb{R}^n.$$

Often, it is desirable to use a generalization of the above minimal requirement (assuming the form (1.1) of P). Such a generalization is as follows: for a given set of vectors $\mathbf{v}_k$, $k = 1, \dots, m$, one has that $\text{Range}(P) \supset \text{Span}\{\mathbf{v}_1, \dots, \mathbf{v}_m\}$.

Let us first point out that this setting of “fitting” exactly several vectors by an interpolation mapping P has some potential difficulties. For example, let Ω be a local set of indices, and let $\{\mathbf{v}_k\}_{k=1}^m$ be a collection of vectors such that their components (restrictions) corresponding to this local set of indices are linearly dependent, that is,

$$\sum_{k=1}^m \alpha_k \mathbf{v}_k|_{\Omega} = 0.$$

We will show that in the case when Ω is the set of coarse dofs, in a generic multilevel method, such local linear dependence *only for the set of coarse dofs* implies that the restrictions of $\{\mathbf{v}_k\}_{k=1}^m$ on larger sets (including fine grid dofs that are close in some sense to Ω) will also be linearly dependent. This leads to the conclusion that “fitting” exactly several vectors might be an impossible task for fixed coarse dofs and fixed places of zero elements in the interpolation matrix P .

As an example of how the aforementioned problem can occur in a generic multilevel method, we consider the AMGe method (see [JV01]). For other variants of AMG such examples can also be easily constructed. Suppose we have a finite element stiffness matrix A that is assembled from a set of local element matrices $\{A_e\}_{e \in \mathcal{T}}$. Here, e stands for an element from a triangulation $\mathcal{T}$ of Ω . Consider now the case of Ω being the set of coarse dofs from a union of elements, $E = \cup e$. Then, one has

$$(1.2) \quad \sum_{k=1}^m \alpha_k \mathbf{v}_k|_{E \cap \mathcal{N}_c} = 0.$$

In the element agglomeration AMGe method sets like E are used to define coarse elements. In order to obtain multilevel hierarchy of spaces, one needs to apply recursion and the coarse element matrices need to be defined appropriately; namely, the constructed P has to satisfy the requirement that it interpolates dofs from E only from coarse dofs also in E . Let P restricted to E be P_E and let A_E be the local matrix corresponding to E , assembled from the element matrices A_e for all elements

e that form E . Thus, based on the above requirement for P , one has that each vector restricted to E is interpolated by the columns of P restricted to E , that is, by the columns of P_E . Therefore, based on (1.2), and the fact that $\mathbf{v}_k|_E \in \text{Range}(P_E)$, or more precisely $\mathbf{v}_k|_E = P_E(\mathbf{v}_k|_{E \cap \mathcal{N}_c})$, one gets that

$$\sum_{k=1}^m \alpha_k \mathbf{v}_k|_E = P_E \left(\sum_{k=1}^m \alpha_k \mathbf{v}_k|_{E \cap \mathcal{N}_c} \right) = 0.$$

The latter equation implies that the set of vectors $\{\mathbf{v}_k\}$ restricted to E (not only to the coarse dofs in E) are linearly dependent. This need not necessarily be true for the given set of vectors $\{\mathbf{v}_k\}$. That is, the task of constructing P to meet the main requirement of the element agglomeration AMGe may not have a solution (in the above setting). One more point in practice is that for relatively small sets E , the vectors $\{\mathbf{v}_k\}$ may be nearly linear dependent locally. This would result in a coarse element matrix $P_E^T A_E P_E$ that has a spurious (near) null eigenvector, due only to our construction. This feature of spurious local (near) null eigenvectors may cause unnecessary difficulties in solving the coarse-grid problem. All difficulties above may happen only if we want to fit several vectors by one and the same P (of the form (1.1)); if only one vector is to be preserved such difficulties do not arise.

On the other hand, the motivation to construct P that preserves a set with more than one vector seems rather natural in some applications, such as elasticity, for example. One needs to interpolate the rigid body vectors then as motivated in [VMB] and [MBV]. In other cases, such as in the recently proposed adaptive algebraic multigrid methods (see [aSA] and [aAMG]) one wants to update a given P such that it keeps in its range a set of previously computed vectors and to fit one more new vector in addition. The new vector is obtained during an adaptation process as explained in these works (see [aSA] and [aAMG]). We would like to point out that the algorithm that we provide in this paper is inherently suitable for applying adaptive procedure, which we view as one of the main contributions of the present paper. Another potential application of interpolation matrices that fit a set of prescribed vectors is in the finite element upscaling methods, in the sense that one can keep on coarse levels coefficient vectors of, for example, linear, quadratic, or any other polynomials, thus ensuring guaranteed good and adequate approximation properties of the coarse upscaled problems. The potential for upscaling is an important feature of the proposed method that is left for more detailed study elsewhere.

With our approach, we first overcome the outlined potential difficulty of local (near) linear dependence, and second, we present an algorithm that constructs P , fitting one vector at the time, while keeping the previous vectors in the range of P (hence inherent adaptivity). More specifically, the algorithm constructs a set of P_k , for $k = 1, \dots, m$, one P_k at a time, such that the resulting P is simply the collection $P = [P_1, P_2, \dots, P_m]$ and every new P_k is a hierarchical complement of the preceding ones, $P_1, \dots, P_{k-1}$. More precisely, we seek P of the form,

$$(1.3) \quad [P_1, P_2, \dots, P_m] = \begin{bmatrix} W_{1,1} & W_{1,2} & \dots & W_{1,m} \\ W_{2,1} & W_{2,2} & \dots & I \\ \vdots & \vdots & \dots & \vdots \\ W_{m,1} & I & \dots & 0 \\ I & 0 & \dots & 0 \end{bmatrix}.$$

By this choice of the structure of P one guarantees that P has full column rank and therefore the Galerkin coarse matrix $P^T A P$ is spd (since A is). If one uses finite

element terminology, one may view the columns of every additional P_k as new basis functions which are typically added in finite element (local) refinement procedures. The newly added basis functions can be seen as hierarchical bubble functions (since they vanish at the preceding coarse set). Thus the proposed construction of P can be seen as an algebraic analogue of finite element refinement.

The actual construction of P is explained in section 2, where the details on constructing the columns of P are presented. In section 3, we describe the choice of the sparsity pattern of the columns of P (the W -blocks in (1.3)) in the setting of the element agglomeration AMGe from [JV01]. We used in our numerical tests the trace minimization construction of the actual entries of each individual P_k using an (inexact) additive Schwarz algorithm based on the construction in [XZ04], which we summarize in section 4. The final section contains some numerical results illustrating the performance of the two-grid and corresponding multilevel AMG methods based on interpolation mappings that interpolate several vectors $\{\mathbf{v}_k\}$. We used in the tests the coefficient vectors of linear functions 1, x , y (the coordinates of the nodes of a given finite element mesh). It appeared that for sufficiently aggressive coarsening, the use of several a priori given vectors to construct P is beneficial for the resulting AMG compared to the more classical approach when only one vector is used to construct P . Alternatively, one may use vectors that are adaptively constructed as described in [aSA]–[aAMG].

2. Construction of multiple vector preserving interpolation. The construction of P that has the form given by (1.3) consists of the following steps. We assume that a coarse grid set $\mathcal{N}_c$ of coarse dofs, which we view as a subset of the fine grid dofs $\mathcal{N} = \{1, 2, \dots, n\}$, is chosen. Then we split $\mathcal{N}_c$ into m groups of nonoverlapping sets, $\mathcal{N}_m^c \cup \mathcal{N}_{m-1}^c \cup \dots \cup \mathcal{N}_1^c = \mathcal{N}_c$. Each P_k admits the form

$$P_k = \begin{bmatrix} \star & & \\ I & & \\ 0 & & \\ \vdots & & \\ 0 & & \end{bmatrix} \begin{array}{l} \} \\ \} \\ \} \\ \vdots \\ \} \end{array} \begin{array}{l} \mathcal{N} \setminus (\cup_{i=1}^k \mathcal{N}_i^c) \\ \mathcal{N}_k^c \\ \mathcal{N}_{k-1}^c \\ \vdots \\ \mathcal{N}_1^c. \end{array}$$

Note that the number of columns in P_k is equal to $|\mathcal{N}_k^c|$ (cardinality of $\mathcal{N}_k^c$). In practice, one starts with a current set $\mathcal{N}_c = \mathcal{N}_1^c$ and at step $k > 1$ one augments it with a newly constructed set $\mathcal{N}_k^c \subset \mathcal{N} \setminus (\cup_{i=1}^{k-1} \mathcal{N}_i^c)$, i.e., $\mathcal{N}_c := \mathcal{N}_c \cup \mathcal{N}_k^c$.

To avoid the potential local (near) linear dependence, after a column P_k for a $k \geq 1$ of P is constructed, such that $P_k \mathbf{v}_k^c = \mathbf{v}_k$, one updates the remaining vectors by computing the residuals

$$\mathbf{v}_l := \mathbf{v}_l^{\text{new}} = \mathbf{v}_l - P_k \mathbf{v}_l^c, \quad l = k+1, \dots, m.$$

Here, $\mathbf{v}_l^c$ are the restrictions of $\mathbf{v}_l$ onto the k th coarse grid $\mathcal{N}_k^c$. The latter corresponds to the row indices of the identity block of P_k . A simple observation is that at every step $k \geq 1$, since $\mathbf{v}_l \in \text{Span}\{\text{Range}(P_k), \mathbf{v}_l^{\text{new}}\}$, $l > k$, one has

$$\begin{aligned} & \text{Span}\{\text{Range}(P_1), \dots, \text{Range}(P_k), \mathbf{v}_{k+1}^{\text{new}}, \dots, \mathbf{v}_m^{\text{new}}\} \\ & \supset \text{Span}\{\mathbf{v}_1, \dots, \mathbf{v}_k, \mathbf{v}_{k+1}, \dots, \mathbf{v}_m\}. \end{aligned}$$

Also, with the above residual modification of the vectors we ensure that the new vectors $\mathbf{v}_l$, for $l > k$, vanish at all preceding coarse dofs, $\cup_{i=1}^k \mathcal{N}_i^c$, $i = 1, \dots, k$. Recall

that we identify the coarse dofs $\mathcal{N}_k^c$ with certain fine dofs, namely, those defined by the row indices of P_k where the identity block of P_k is placed. The fact that the residual vector $\mathbf{v}_{k+1}$ vanishes at $\cup \mathcal{N}_i^c$, $i = 1, 2, \dots, k$, allows us to seek P_{k+1} having zero rows corresponding to the set $\cup \mathcal{N}_i^c$, $i = 1, 2, \dots, k$. Thus, we reduce the problem of constructing a mapping P that interpolates exactly $m \geq 1$ vectors $\mathbf{v}_k$ to m consecutive tasks of constructing a mapping P_k that interpolates only one vector exactly, namely, the residual vector $\mathbf{v}_k := \mathbf{v}_k^{new}$. The actual algorithm that one can utilize in practice depends on the goal one wants to achieve. In AMG, one needs to construct a P that has a certain minimal norm. For example, in [WCS00], and more recently in [XZ04], one constructs a P (in our case, P_k) that minimizes the trace of the coarse matrix $P_k^T A P_k$. This is not the only possible choice (see, e.g., [Ch03]).

We have not specified the choice of the sparsity pattern of the columns P_k of P , i.e., the sparsity of the W -blocks in (1.3). This is done in the following section for one specific AMG method, namely, the element agglomeration AMGe ([JV01]).

3. Selection of sparsity pattern of P in element agglomeration AMGe.

In summary, the element agglomeration AMGe ([JV01]) assumes that in addition to the fine grid matrix A one has access to the fine-grid element matrices $\{A_e\}_{e \in \mathcal{T}}$ and the set of elements $e \subset \mathcal{T}$, the latter meaning that one knows relations describing the topology of the fine-grid elements. For example, one needs the incidence matrices element-dof, element-face, face-dof, etc. Based on an agglomeration algorithm (see [JV01]), one constructs coarse elements E , which are union on (connected to) fine elements, and as described in detail in [V02] one can construct the topology of the coarse elements. In the “element agglomeration AMGe” method one imposes the restriction that the interpolation mapping P has the property that fine dofs that belong to an agglomerated element E are interpolated only from coarse dofs in E . This in particular implies that dofs shared by two or more agglomerated elements should be interpolated only from coarse dofs that are shared by the same agglomerated elements (that share that fine dof). In some instances this implies that such shared dofs must become coarse dofs. This is the case for all vertex dofs. By vertex here we mean (assuming scalar PDE and Lagrangian finite elements) the intersection of neighboring agglomerated elements (AEs) into a single dof. That is, the set of vertices provides a natural minimal coarse grid. This was the choice made in the original paper [JV01]. The sparsity pattern of P is then clear. For every such “vertex” coarse dof “ i ,” there is a column ψ_i in P and the nonzero entries of ψ_i correspond to a subset of all dofs in the union of AEs E that share that vertex. A natural subset would be the dofs in the union Ω_i of the AEs that share vertex “ i ” which are not on the boundary of Ω_i . The “boundary” set is defined as dofs in Ω_i that belong to other AEs (not in Ω_i). Also, one excludes from Ω_i all other coarse dofs (different from i). The resulting local set of fine dofs defines the support of the “ i ”th column of P . By this choice one guarantees that dofs belonging to an agglomerate E are interpolated only from coarse dofs in that agglomerated element. This is the main requirement that allows one to define a coarse element matrix in a natural way; namely, let A_E be the assembled local matrix corresponding to the agglomerated element E . Then, the coarse element matrix, corresponding to the coarse element E , reads $A_E^c = P_E^T A_E P_E$. Here P_E is the restriction of P to the set E , i.e., the rows of P_E correspond to the rows of P restricted to E and the columns of P_E correspond to the columns of P restricted to the coarse dofs from E .

Recall that in the setting of the preceding section, we have $P = [P_1, P_2, \dots, P_m]$. Then the sets $\mathcal{N}_k^c$ are spread over the AEs and the respective column “ i ” of P_k will have

support subjected to the AEs that share coarse dof “ i ,” the boundary dofs excluded, as well as all coarse dofs (different from “ i ”) from the current and preceding sets $\mathcal{N}_j^c$, i.e., for $j \leq k$, also excluded.

4. A trace minimization construction of the interpolation coefficients.

The algorithm that we implemented to construct the actual entries of P_k that fits a single vector was based on the additive Schwarz method as described in [XZ04]. However, for practical purpose, the inversion of the actual Schwarz blocks was done only approximately, by a few Gauss–Seidel iterations. The first vector (the constant one) led to a matrix P_1 which was computed by a little more accurate inversion of the Schwarz blocks, whereas for the remaining vectors the Schwarz blocks were approximated only by one symmetric Gauss–Seidel iteration. To justify this, first note that the constant is in the null-space of the matrix (in the interior of the domain) and it is shown in [XZ04] that the basis functions (giving rise to the columns of P_k) are locally harmonic. This is an important property in order to achieve good two-grid convergence for Laplace-like matrices. Second, note that for $k > 1$, P_k is sought vanishing at a starting set of coarse dofs $\mathcal{N}_1^c$. Therefore one may expect that $P_k^T A P_k$ will be relatively well-conditioned. Thus, the accuracy with which one seeks P_k (for $k > 1$) with optimal min-trace properties should not be that important (at least theoretically and also as observed in our experiments).

The specific algorithm is as follows. Let Ω_i be the support set of the i th column of a given P_k . Let I_i be the extension by zero of a vector defined on Ω_i . Then, the i th Schwarz block is defined as $T_i = I_i (I_i^T A I_i)^{-1} I_i^T$. The Gauss–Seidel iterations are used to evaluate (approximately) the actions of $(I_i^T A I_i)^{-1}$. Then, if one wants to build a $P := P_k$ such that $P \mathbf{v}^c = \mathbf{v}$, where $\mathbf{v}^c = \mathbf{v}|_{\mathcal{N}_c}$, one proceeds as follows. Let $T = \sum_{i \in \mathcal{N}_c} g_i^2 T_i$. Here, $g_i = \mathbf{v}_k(i)$. For a given vector $\mathbf{w}$, $\mathbf{w}(i)$ stands for its i th entry. Then, $\psi_i = g_i T_i T^{-1} \mathbf{v}$ defines the i th column of P . The vector $\mathbf{x} = T^{-1} \mathbf{v}$ is computed by the preconditioned conjugate gradient (PCG) method applied to $T \mathbf{x} = \mathbf{v}$ with a simple diagonal matrix $D = \sum_{i \in \mathcal{N}_c} g_i^2 I_i D_i^{-1} I_i^T$, where $D_i = \text{diag } I_i^T A I_i$, as preconditioner. It is readily seen (by a local analysis) that D is a spectrally equivalent preconditioner to T . Thus, only a few PCG iterations are needed to compute $\mathbf{x} = T^{-1} \mathbf{v}$. By construction then, $P \mathbf{v}_c = \sum_{i \in \mathcal{N}_c} g_i \psi_i = \mathbf{v}$. The fact that the resulting P leads to $P^T A P$ with minimal trace is proved in [XZ04]. There is also a motivation to choose the i th coarse dof as in Algorithm 5.2; namely, that then T gets better conditioned. Note that too small a value of $g_i = \mathbf{v}(i)$ will make T almost singular, or very ill-conditioned.

5. Algorithms and numerical illustration.

Algorithms for augmenting current coarse grid. It may be a typical situation in practice that an initial coarse grid $\mathcal{N}_c := \mathcal{N}_1^c$ is used to construct a $P := P_1$ that fits the first vector $\mathbf{v}_1$. Then, after one has computed the residual vectors $\mathbf{v}_k := \mathbf{v}_k - P_1 \mathbf{v}_k^c$, $k = 2, \dots, m$ with $\mathbf{v}_k^c = \mathbf{v}_k|_{\mathcal{N}_1^c}$, one selects the additional coarse set $\mathcal{N}_2^c \subset \mathcal{N} \setminus \mathcal{N}_c$, sets $\mathcal{N}_c := \mathcal{N}_c \cup \mathcal{N}_2^c$, and proceeds further. To account for (near) linear local dependence, one may want to incorporate a dropping strategy, that is, if some entries of the (residual) vectors are very small one drops them by zeroing them. Then, in the case of element agglomeration AMGe, one may adopt the following strategy. Based on a “minimal intersection set” algorithm (see below), one can partition all AEs into nonoverlapping sets (groups) of dofs in $\mathcal{N} \setminus \mathcal{N}_c$. Here, at step k , $\mathcal{N}_c := \mathcal{N}_1^c \cup \dots \cup \mathcal{N}_{k-1}^c$. These minimal intersection sets, in the case of true geometri-

cal finite elements, are the vertices, edge interior, face interior, and AE-interior dofs. Then, if $\mathbf{v}_k$ is nonzero at a given minimal intersection set, one selects a new coarse dof from that set, for example, by looking at the local maximal absolute value of $\mathbf{v}_k$ restricted to that set. If $\mathbf{v}_k$ is zero (after potential dropping of small entries), one does not have to select any additional coarse dofs from that particular intersection set. All newly selected coarse dofs form $\mathcal{N}_k^c$.

The algorithm for forming the *minimal intersection sets* uses the relation “AE_dof” represented by the incidence matrix $\mathcal{E}$, defined as

$$\mathcal{E}_{ij} = \begin{cases} 1 & \text{if dof } j \text{ is in the agglomerated element } i, \\ 0 & \text{otherwise.} \end{cases}$$

Note that $\mathcal{E} \in \mathbb{R}^{n_E \times n}$, where n_E is the number of AEs and n is the number of degrees of freedom. Consider now $\mathcal{Z} = \mathcal{E}^T \mathcal{E} \in \mathbb{R}^{n \times n}$ and observe that

$$\mathcal{Z}_{ij} = \#\{E \mid i \in E \text{ and } j \in E\},$$

where $\#$ stands for cardinality. In other words, the right-hand side of the above equation equals the number of AEs E such that both dofs i and $j \in E$. We then split the set of degrees of freedom $\{1, \dots, n\}$ in nonoverlapping sets $\{\mathcal{I}_k\}_{k=1}^\ell$ (called *minimal intersection sets*) with the property that i and j are in one and the same set $\mathcal{I}_k$ if and only if $\mathcal{Z}_{ij} = \mathcal{Z}_{ii} = \mathcal{Z}_{jj}$.

A more formal algorithm to determine these sets is given below. As input, the algorithm takes $\mathcal{Z}$ (or equivalently $\mathcal{E}$), and as output, we have the number of minimal intersection sets ℓ and also a vector $\chi = (\chi(i))_{i=1}^n$ such that $\chi(i) = k$ if and only if $i \in \mathcal{I}_k$.

ALGORITHM 5.1 (Minimal intersection sets).

```

Set  $\ell = 0$ ;  $\chi(i) = 0$ ,  $i = 1, \dots, n$ .
For  $i = 1$  to  $n$  do
  If  $\chi(i) = 0$ , then
     $\ell = \ell + 1$ 
    For  $j$  such that  $\chi(j) = 0$  and  $\mathcal{Z}_{ij} = \mathcal{Z}_{ii} = \mathcal{Z}_{jj}$  do
      Set  $\chi(j) = \ell$ 
    end For
  end If
end For

```

Clearly, one can easily derive a version of this algorithm without explicitly forming $\mathcal{Z}$. In either version, the number of operations would be proportional to the number of nonzeros in $\mathcal{E}$.

Next, we present an algorithm on how to select an additional coarse grid. We assume that we are given a vector $\mathbf{v} := \mathbf{v}_k \neq \mathbf{0}$ and denote with $\|\mathbf{v}\|_{\max}$ its ℓ_∞ norm, namely, $\|\mathbf{v}\|_{\max} = \max_i |\mathbf{v}(i)|$. Here, and in the algorithm below, $\mathbf{v}(i)$ stands for the i th entry of $\mathbf{v}$. We also assume that we are also given a set of coarse degrees of freedom $\mathcal{N}_c := \cup_{j=1}^k \mathcal{N}_j^c$, and a drop tolerance $\delta \in [0, 1)$. As the output of the algorithm, we will obtain an additional set of coarse degrees of freedom $\mathcal{N}_c^{k+1}$ (denoted below with $\mathcal{N}_\mathbf{v}^c$) and the range of the interpolation will contain $\mathbf{v}$. Further, let $\mathcal{N} := \mathcal{N} \setminus \mathcal{N}_c$, and let $\{E\}$ be a overlapping partition of the dofs in $\mathcal{N}$ into sets E , which we refer to as AEs. An additional coarse grid $\mathcal{N}_\mathbf{v}^c$ is then obtained as a subset of the current $\mathcal{N}$ in the following steps.

TABLE 1

Two-grid (TG) results with linear vector preserving interpolation; coarsening factor = 128.

# elements	2048	8192	32768	131072
# dofs	1089	4225	16641	66049
# PCG iter	8	9	10	10
# TG iter	20	18	19	19
ϱ_{TG}	0.40	0.357	0.39	0.388

TABLE 2

Multilevel AMG results with linear vector preserving interpolation; first level element coarsening factor = 128.

# elements	2048	8192	32768	131072
# dofs	1089	4225	16641	66049
# PCG iter	8	9	9	9
# AMG iter	20	18	20	23
ϱ_{AMG}	0.40	0.359	0.386	0.41
operator complexity	1.39	1.62	1.79	1.89
# levels	4	6	8	10

ALGORITHM 5.2 (Selecting additional coarse grid dofs in AMGe).

- Drop small entries, i.e., if $|\mathbf{v}(i)| \leq \delta \|\mathbf{v}\|_{\max}$, set $\mathbf{v}(i) = 0$. Put $\mathcal{N}_c^{\mathbf{v}} = \emptyset$.
- Construct the minimal intersection sets based on two given relations “AE_dof” and “dof_AE” represented by the incidence matrices $\mathcal{E}$ and $\mathcal{E}^T$, or as described in Algorithm 5.1, based only on the product matrix $\mathcal{E}^T \mathcal{E}$.

For every minimal intersection set $\mathcal{I}$ **do**

- Set $i_0(\mathcal{I}) = \emptyset$, and let

$$i_{\max}(\mathcal{I}) = \arg \max \{|\mathbf{v}(j)|, j \in \mathcal{I}, \mathbf{v}(j) \neq 0\}.$$

- If $i_{\max}(\mathcal{I}) \neq \emptyset$, then set $i_0(\mathcal{I})$ to be any element j from $i_{\max}(\mathcal{I})$ and augment $\mathcal{N}_c^{\mathbf{v}}$ in the following way

$$\mathcal{N}_c^{\mathbf{v}} := \mathcal{N}_c^{\mathbf{v}} \cup i_0(\mathcal{I})$$

end For

Numerical tests. The test problems were the Poisson equation discretized on a sequence of triangular meshes on the unit square domain with homogeneous Dirichlet boundary conditions using standard Lagrangian piecewise linear elements. We used element agglomeration with a large coarsening factor (at the first coarse level only), that is, to get the number of AEs equal to the number of fine-grid elements divided by the coarsening factor. In our tests the element coarsening factor (at the first coarse level) was either 128 or as close as possible to 128. At the next level the element coarsening factor was much smaller (8, 4, or 2). The initial coarse grid $\mathcal{N}_c = \mathcal{N}_1^c$ was simply the set of vertices of the resulting AEs (i.e., as in the original element agglomeration AMGe method from [JV01]). The sparsity pattern of the interpolation matrices was chosen as described in section 3, i.e., based on the AEs and the current fine grid.

The vectors that we tried to fit were 1, x , and y , i.e., $\mathbf{v}_1 = \mathbf{1} = (1)_{i \in \mathcal{N}}$, $\mathbf{v}_2 = (x_i)_{i \in \mathcal{N}}$, and $\mathbf{v}_3 = (y_i)_{i \in \mathcal{N}}$. Here, for a dof i , (x_i, y_i) stands for its geometrical coordinates.

TABLE 3

Multilevel AMG results with constant vector preserving interpolation; first level element coarsening factor = 128.

# elements	2048	8192	32768	131072
# dofs	1089	4225	16641	66049
# PCG iter	10	16	18	18
# AMG iter	57	74	80	81
ϱ_{AMG}	0.69	0.75	0.76	0.76
# levels	4	6	8	10

TABLE 4

Multilevel and two-level AMG results with linear vector preserving interpolation for anisotropic diffusion problem; first level element coarsening factor = 100.

# elements	400	1600	6400	25600
# dofs	231	861	3321	13041
# PCG iter	7	15	23	38
# AMG iter	18	50	99*	99*
ϱ_{AMG}	0.374	0.71	0.84	0.94
operator complexity	1.15	1.51	1.86	1.96
# levels	2	4	6	8
two level # PCG iter	7	15	24	40
# TG iter	18	52	99*	99*
ϱ_{TG}	0.374	0.71	0.84	0.94

In the construction of all P_k , $k = 1, 2, 3$ we used the construction of P_k with the prescribed sparsity pattern so that the resulting coarse matrix has minimal trace, as summarized in section 4.

We show, for illustration, a complete set of coarse basis “functions” (columns of the interpolation mapping P) in Figures 3, 4, and 5, corresponding to one agglomerated element from the mesh shown in Figure 2. In Figure 6, we show that these basis functions span locally the constant one and the coordinate vectors (x_i) and (y_i) .

In the presented tables, one finds two-level and multilevel convergence results for a $V(1,1)$ -cycle AMG, used as a preconditioner in both the CG method as well as in a stationary iterative process. The stopping tolerance was a 10^{-9} relative reduction of the initial residual norm; the smoother was the overlapping Schwarz method. The Schwarz domains were the elements (at the given level) viewed as sets of dofs. At the initial mesh, these corresponded to three-by-three matrices. In the case of stationary iterations we also show the asymptotic convergence factor ϱ . One can compare the methods (see Tables 2 and 3) using the same AEs but different interpolation matrices P . Namely, Table 2 corresponds to P that preserved the linear vectors, whereas in Table 3 only the constant vector was exactly interpolated.

Also for comparison, in Table 1 the two-level results are shown to demonstrate how close the multilevel results (from Table 2) match the two-level ones.

The tables with multilevel results contain numbers denoted by “operator complexity” which by definition equals the sum of the nonzero entries of the matrices on all levels divided by the number of nonzero entries of the fine-grid matrix. It gives an indication about the storage and the cost of one V-cycle.

Finally, in order to demonstrate that just a simple aggressive element agglomeration (as above) may not always work, we show in Table 4 results of the same nature as above, now for a difficult anisotropic diffusion problem. The anisotropy is gener-

TABLE 5

Multilevel AMG results with vector preserving interpolation for anisotropic diffusion problem; adaptive AMG used to generate 3 vectors; first level element coarsening factor = 32. Last column stands for 5 adaptively constructed vectors and first level element coarsening factor = 64.

# elements	400	1600	6400	25600	25600
# dofs	231	861	3321	13041	13041
# PCG iter	3	7	10	18	14
# AMG iter	3	12	27	78	48
ϱ_{AMG}	0.03	0.38	0.66	0.87	0.78
operator complexity	2.21	3.68	4.84	5.45	5.23
# levels	2	6	8	10	10

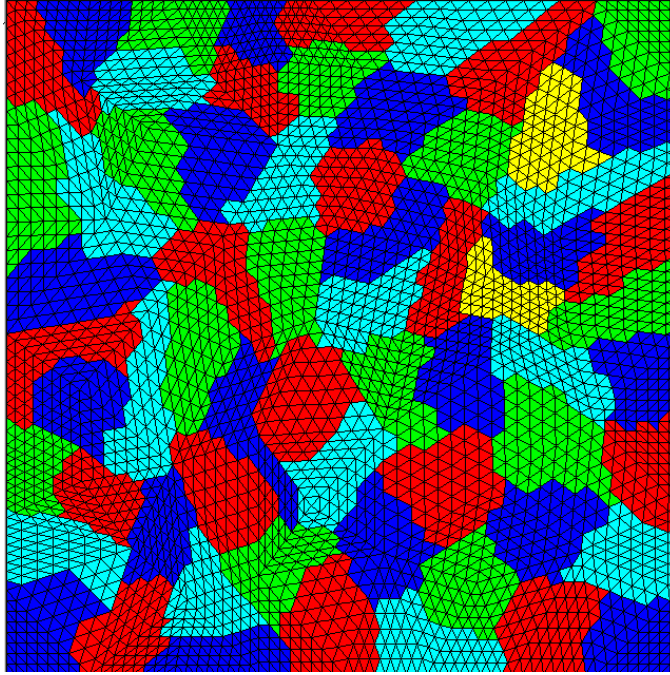


FIG. 1. Agglomerated elements from unstructured fine mesh; element coarsening factor = 100. Initial mesh has 6400 elements; # agglomerated elements = 64.

ally not grid-aligned. One of the meshes, after one level of agglomeration, is shown in Figure 1. It is well known that such problems would require special coarsening, based, for example, on a “compatible relaxation” algorithm, which in short selects coarse dofs, tight to the convergence of the smoother restricted to a grid complementary to the coarse one (a strong theoretical justification for this argument is given in [FVZ04]). The notion of “compatible relaxation” is introduced in [B00]. For more details in this direction we refer to [L04] and [FV04]. Alternative efficient coarsening can be achieved, in the element agglomeration AMGe framework, based on a variant ([Ch05]) of the spectral AMGe method from [Ch03]. The topic of compatible relaxation, however, is not the focus of the present paper. As yet another alternative we show in Table 5 some preliminary results based on adaptively constructed test vectors

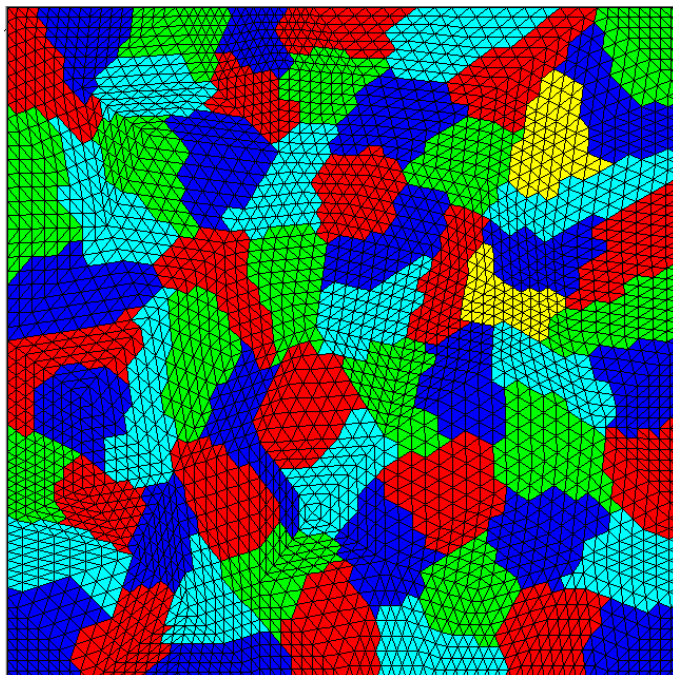


FIG. 2. *Agglomerated elements; element coarsening factor = 128. Initial mesh has 8192 elements; # agglomerated elements = 64.*

as proposed in [aSA] and [aAMG]. The difference from these adaptive AMG methods here is only in the construction of the interpolation matrices, updated one level at a time based on a currently constructed vector. If one compares Tables 5 and 4 the improvement in terms of the number of iterations is noticeable, however, with the expense of higher operator complexity. Further research is needed in order to make the adaptive AMG methods in the element agglomeration setting with better complexity.

Conclusions. In conclusion, utilizing several vectors in the construction of AMG interpolation mappings can be useful if relatively aggressive coarsening (at initial coarse levels) is exploited in the algorithm. If the (near) null-space vectors (or rigid body modes) are not known in advance one may utilize their adaptive construction as proposed in [aSA] and [aAMG] combined with the proposed in the present paper multiple vector preserving interpolations, which is inherently suited for use in adaptive AMG procedures. Other potential applications of such interpolation mappings (or resulting coarse spaces) can be expected in nonlinear AMG methods or in finite element upscaling methods, where one needs more accurate (from an approximation point of view) coarse spaces. These topics are not considered in the present paper.

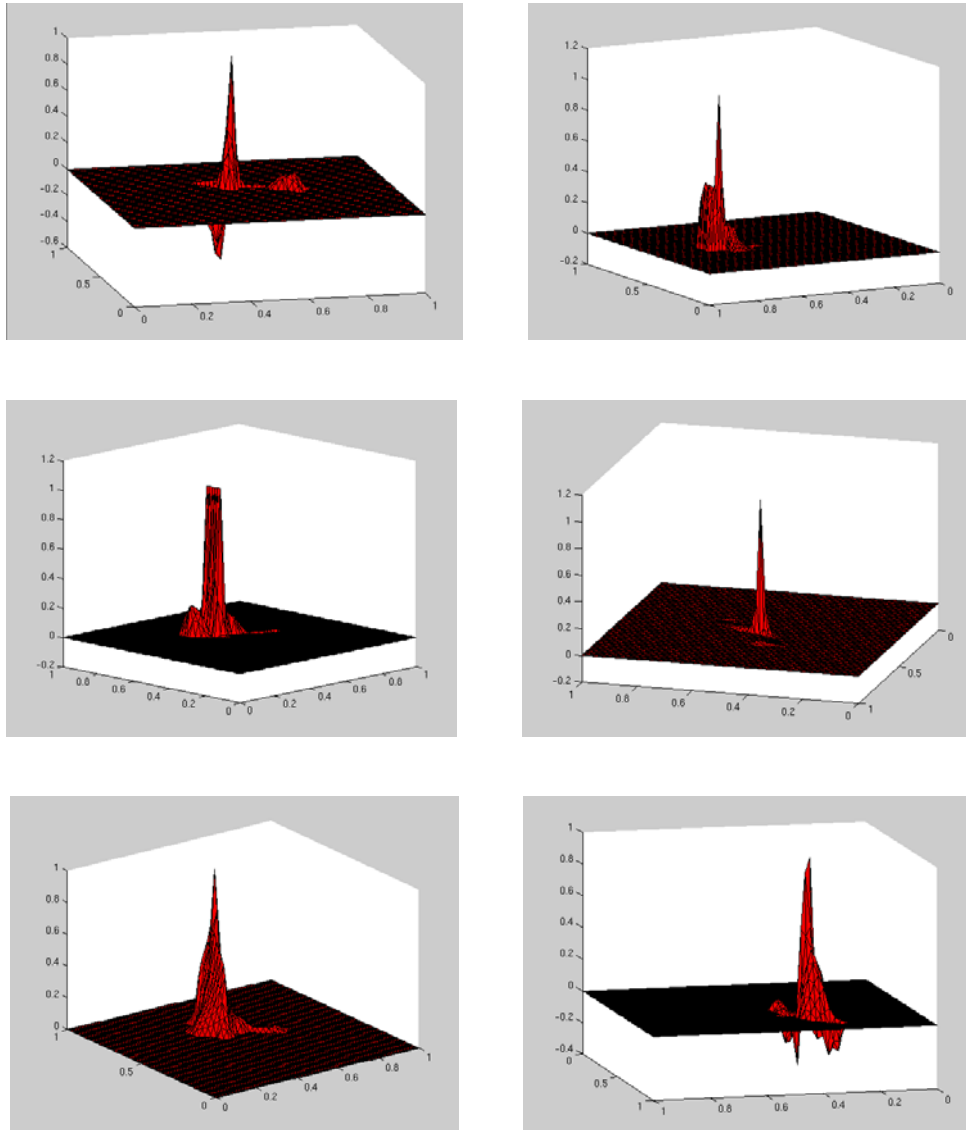


FIG. 3. *Basis functions associated with one agglomerated element; first six.*

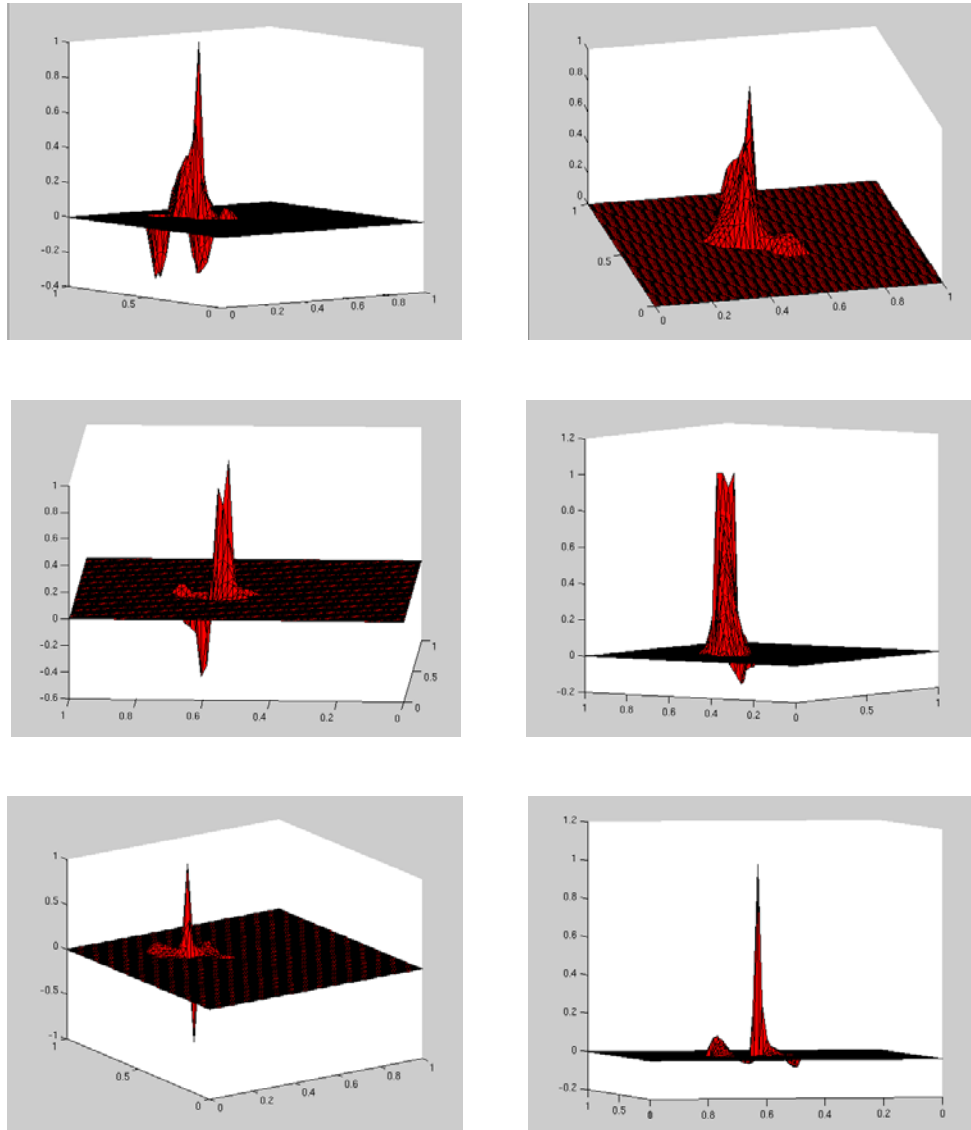


FIG. 4. Basis functions associated with one agglomerated element; seventh to twelfth.

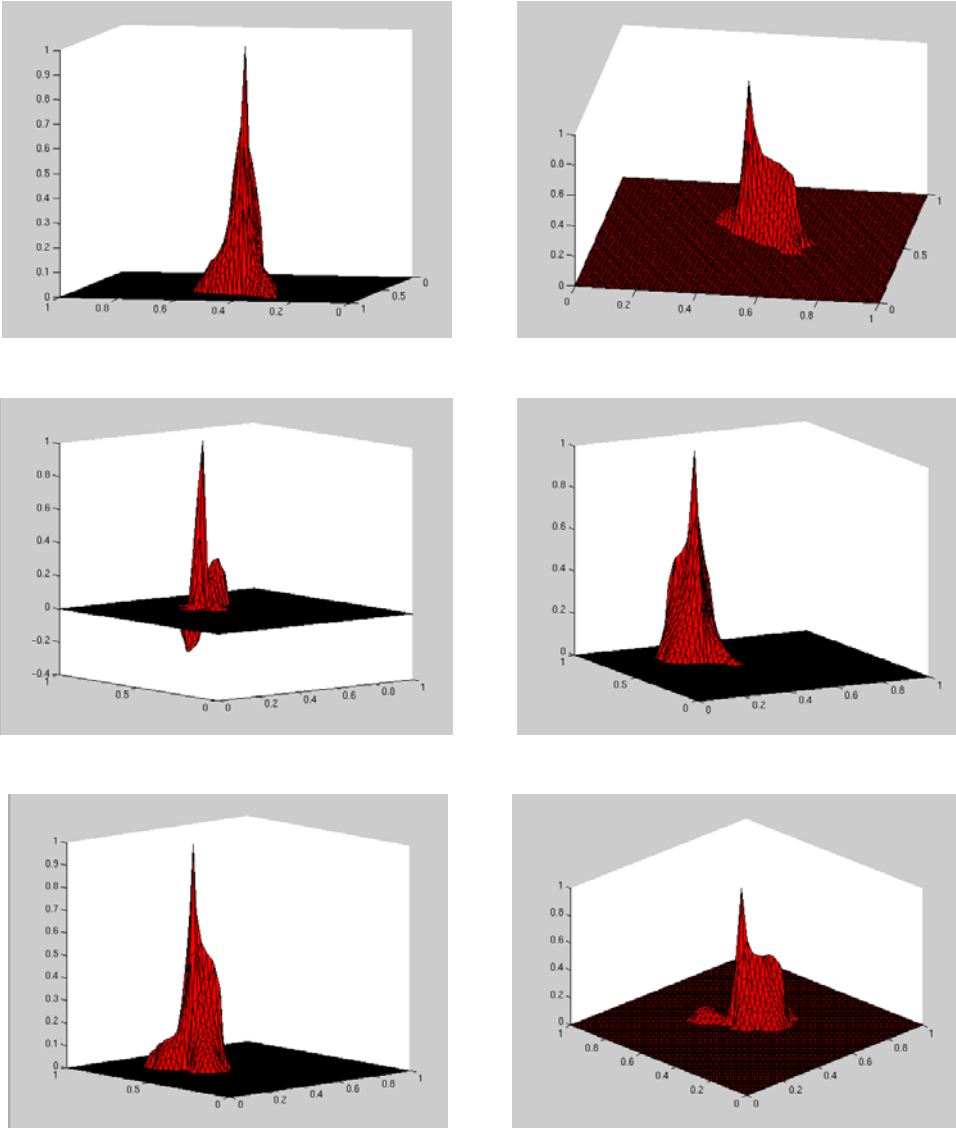


FIG. 5. Basis functions associated with one agglomerated element; 13th to 18th.

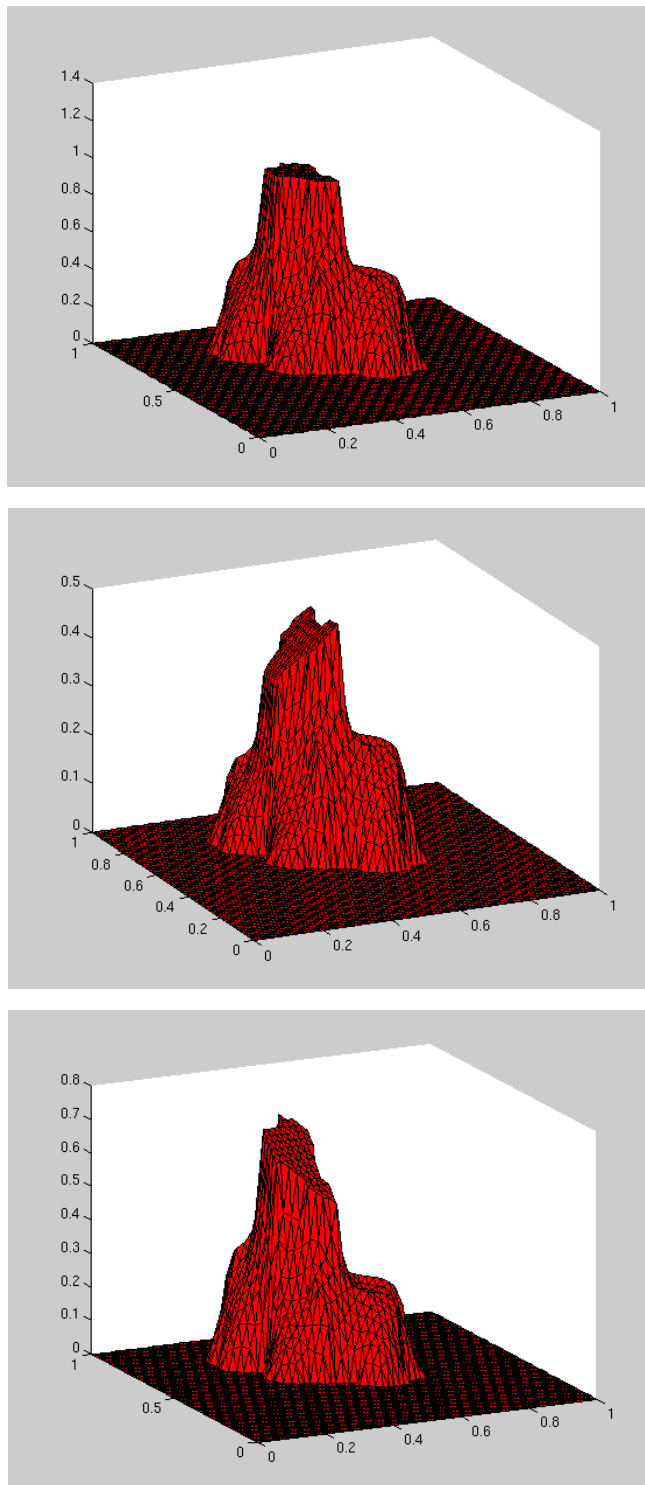


FIG. 6. Local representation of 1, x , and y functions associated with one agglomerated element.

REFERENCES

- [B00] A. BRANDT, *General highly accurate algebraic coarsening*, Electron. Trans. Numer. Anal., 10 (2000), pp. 1–20.
- [aSA] M. BREZINA, R. FALGOUT, S. MACLACHLAN, T. MANTEUFFEL, S. MCCORMICK, AND J. RUGE, *Adaptive smoothed aggregation (aSA)*, SIAM J. Sci. Comput., 25 (2004), pp. 1896–1920.
- [aAMG] M. BREZINA, R. FALGOUT, S. MACLACHLAN, T. MANTEUFFEL, S. MCCORMICK, AND J. RUGE, *Adaptive algebraic multigrid*, SIAM J. Sci. Comput., 27 (2006), pp. 1261–1286.
- [Ch03] T. CHARTIER, R. FALGOUT, V. E. HENSON, J. JONES, T. MANTEUFFEL, S. MCCORMICK, J. RUGE, AND P. S. VASSILEVSKI, *Spectral AMGe (ρ AMGe)*, SIAM J. Sci. Comput., 25 (2003), pp. 1–26.
- [Ch05] T. CHARTIER, R. FALGOUT, V. E. HENSON, J. JONES, T. MANTEUFFEL, S. MCCORMICK, J. RUGE, AND P. S. VASSILEVSKI, *Spectral element agglomerate AMGe*, in Proceedings of the 16th International Conference on Domain Decomposition Methods, New York University, New York, 2005, Lecture Notes in Computational Science and Engineering, Springer Verlag, to appear.
- [FV04] R. D. FALGOUT AND P. S. VASSILEVSKI, *On generalizing the algebraic multigrid framework*, SIAM J. Numer. Anal., 42 (2004), pp. 1669–1693.
- [FVZ04] R. D. FALGOUT, P. S. VASSILEVSKI, AND L. T. ZIKATANOV, *On two-grid convergence estimates*, Numer. Linear Algebra Appl., 12 (2005), pp. 471–494.
- [JV01] J. E. JONES AND P. S. VASSILEVSKI, *AMGe based on element agglomerations*, SIAM J. Sci. Comput., 23 (2001), pp. 109–133.
- [L04] O. E. LIVNE, *Coarsening by compatible relaxation*, Numer. Linear Algebra Appl., 11 (2004), pp. 205–227.
- [MBV] J. MANDEL, M. BREZINA, AND P. VANĚK, *Energy optimization of algebraic multigrid bases*, Computing, 62 (1999), pp. 205–228.
- [VMB] P. VANĚK, J. MANDEL, AND M. BREZINA, *Algebraic multigrid by smoothed aggregation for second and fourth order elliptic problems*, Computing, 56 (1996), pp. 179–196.
- [V02] P. S. VASSILEVSKI, *Sparse matrix element topology with application to AMG and preconditioning*, Numer. Linear Algebra Appl., 9 (2002), pp. 429–444.
- [WCS00] W. L. WAN, T. F. CHAN, AND B. SMITH, *An energy-minimizing interpolation for robust multigrid methods*, SIAM J. Sci. Comput., 21 (2000), pp. 1632–1649.
- [XZ04] J. XU AND L. T. ZIKATANOV, *On an energy minimizing basis in algebraic multigrid methods*, Comput. Vis. Sci., 7 (2004), pp. 121–127.

A MULTIPRECONDITIONED CONJUGATE GRADIENT ALGORITHM*

ROBERT BRIDSON[†] AND CHEN GREIF[†]

Abstract. We propose a generalization of the conjugate gradient method that uses multiple preconditioners, combining them automatically in an optimal way. The algorithm may be useful for domain decomposition techniques and other problems in which the need for more than one preconditioner arises naturally. A short recurrence relation does not in general hold for this new method, but in at least one case such a relation is satisfied: for two symmetric positive definite preconditioners whose sum is the coefficient matrix of the linear system. A truncated version of the method works effectively for a variety of test problems. Similarities and differences between this algorithm and the standard and block conjugate gradient methods are discussed, and numerical examples are provided.

Key words. symmetric positive definite linear systems, conjugate gradients, preconditioning, Krylov subspace solvers

AMS subject classification. 65F10

DOI. 10.1137/040620047

1. Introduction. The conjugate gradient (CG) method celebrated a few years ago a milestone 50th anniversary. In the years since its conception in 1952 [9], the preconditioned version of CG (PCG) has established itself as the method of choice for iteratively solving large sparse symmetric positive definite (SPD) linear systems, $Ax = b$. Throughout the years the method has seen many variants and generalizations, blossoming into a large family of Krylov subspace solvers.

This paper is concerned with a variation of the standard PCG method that employs multiple preconditioners. We call the algorithm MPCG: multipreconditioned conjugate gradient. The motivation is that for preconditioning certain problems there are several alternative approaches with different desirable properties, but it may be difficult to combine them into a single effective preconditioner. A multipreconditioned solver could automatically take advantage of all available preconditioners.

Our new iterative method obtains an energy norm minimization property, while maintaining A -conjugation and orthogonality properties similar to PCG, but with iterates constructed in a *generalized* Krylov space incorporating an arbitrary set of preconditioners.

As with flexible CG [10], our approach cannot generally maintain one of the most attractive features of standard PCG: the famous three-term recurrence relation. However, in practice a truncated version of the method works efficiently in many cases we have tested. Moreover, we are able to show analytically that for the case of two preconditioners whose sum is equal to the coefficient matrix, a short recurrence relation does hold.

Block versions of CG have been proposed in the literature, but none of them considers using multiple preconditioners. O’Leary [11] derived a block CG method

*Received by the editors December 2, 2004; accepted for publication (in revised form) by M. Benzi May 12, 2005; published electronically March 17, 2006. This work was supported in part by the Natural Sciences and Engineering Research Council of Canada.

<http://www.siam.org/journals/simax/27-4/62004.html>

[†]Department of Computer Science, The University of British Columbia, Vancouver, BC V6T 1Z4, Canada (rbridson@cs.ubc.ca, greif@cs.ubc.ca).

designed to handle multiple right-hand sides, which is also capable of accelerating the convergence of linear systems with a single right-hand side. The method uses a single preconditioner and can be classified as a block Krylov method.

Brezinski [3] proposes an effective adaptation of the block CG method for solving with a single right-hand side, partitioning the initial residual into multiple search directions (see also Bantegnies and Brezinski [1]). This partitioning approach makes it possible to group together blocks of components which might have distinct behavior, possibly due to representing different physical properties. A variety of multiparameter methods have been derived using this methodology; see [3] and references therein.

A block method designed for a parallel computing environment is proposed by Gu et al. in [7, 8]. The unknowns are partitioned into disjoint subdomains, taking into consideration the number of parallel processors available, but for any given search direction the conjugation property is weakened by zeroing out components of the search directions in other subdomains, eliminating global communication bottlenecks in a parallel environment at the price of slower convergence.

Related work that should also be mentioned is the family of flexible or inexact methods [5, 10, 12, 14]. In those methods the preconditioner changes throughout the iteration, whereas in the method that we propose, MPCG, the preconditioners are fixed and are determined a priori. (We note, however, that a flexible variant of MPCG can be applied in a straightforward fashion.)

The remainder of this paper is structured as follows. In section 2 we derive a multipreconditioned steepest descent method. In section 3, the main part of the paper, we derive the MPCG algorithm and discuss its properties. Section 4 is devoted to numerical experiments. Finally, in section 5 we draw some conclusions and point out possible directions for future research.

Throughout we will assume without loss of generality that our initial guess is $x_0 = 0$, with initial residual $r_0 = b$; generally quantities computed at the i th iteration of an algorithm will have the subscript i . The matrix A is symmetric positive definite, and the preconditioners are M_j , $j = 1, \dots, k$, where $M_j^{-1} \approx A^{-1}$. This is the one case where the subscript does not indicate the iteration at which the quantity is computed: $M_1, \dots, M_k$ are fixed throughout the algorithm.

2. Multipreconditioned steepest descent (MPSD). Since A is symmetric positive definite, it is possible to employ the notion of an energy norm: $\|e\|_A = \sqrt{e^T A e}$. The basic steepest descent (SD) algorithm for solving $Ax = b$ is to take the negative gradient of the energy norm of the error, i.e., the steepest descent direction, which also happens to be the current residual vector r_i , as the search direction for a step that minimizes the energy norm of the error associated with the new guess:

$$\begin{aligned} p_{i+1} &= r_i, \\ \alpha_{i+1} &= (p_{i+1}^T A p_{i+1})^{-1} (p_{i+1}^T r_i), \\ x_{i+1} &= x_i + \alpha_{i+1} p_{i+1}, \\ r_{i+1} &= r_i - \alpha_{i+1} A p_{i+1}. \end{aligned}$$

Of course, convergence is much faster if the search direction is closer to the actual error $A^{-1}r_i$, so it is natural to precondition this iteration by instead choosing $p_{i+1} = M^{-1}r_i$, where $M^{-1} \approx A^{-1}$, obtaining a *preconditioned* SD algorithm (PSD).

One possible approach to further improve the new residual is to enlarge the search space from one dimension to multiple dimensions: use a set of search directions $p^1, \dots, p^k$. In particular, if multiple preconditioners $M_1, \dots, M_k$ are available, use

$p_{i+1}^j = M_j^{-1} r_{i+1}$. Let $P_i = [p_i^1 | \dots | p_i^k]$. To get the same energy norm minimization, we derive an MPSD algorithm:

$$\begin{aligned} p_{i+1}^j &= M_j^{-1} r_i \quad \text{for } j = 1, \dots, k, \\ \alpha_{i+1} &= (P_{i+1}^T A P_{i+1})^{-1} (P_{i+1}^T r_i), \\ x_{i+1} &= x_i + P_{i+1} \alpha_{i+1}, \\ r_{i+1} &= r_i - A P_{i+1} \alpha_{i+1}. \end{aligned}$$

Note that, as is the case for other block methods, α_{i+1} is now a *vector* of coefficients specifying the linear combination of search directions for updating the guess.

3. MPCG. Although the SD method converges, it is inefficient compared with the CG method. This section establishes the multipreconditioned analogue of CG in a similar fashion to the derivation of the standard PCG, whose first step is an iteration of PSD. The analogy to the derivation of the standard method with a single preconditioner allows us to make the reasonable assumption that MPCG will improve on MPSD in a way similar to the improvement obtained by using CG (or PCG) rather than SD (or PSD).

3.1. Derivation. One way of looking at CG and why it is so much faster than SD is to interpret it as a generalized SD with multiple search directions. At step $i+1$, the search directions are simply $p_1, \dots, p_{i+1}$, i.e., the new search direction plus all the previous ones. Thus we get a global energy norm minimum, not just a local greedy minimization. The clever part about CG is choosing the search directions to be A -conjugate, so that $P_{i+1}^T A P_{i+1}$ is just diagonal and trivial to invert. Furthermore, due to the global minimization, the previous search directions P_i are orthogonal to the most recent residual, and thus $P_{i+1}^T r_i$ is zero except for the last component, making the update even simpler.

We will want to preserve these features in generalizing PCG to have multiple search directions per step (generated from multiple preconditioners). That is, we want the property

$$P_i^T A P_j = 0 \quad \text{for } i \neq j.$$

We begin with one step of MPSD:

$$\begin{aligned} p_1^j &= M_j^{-1} r_0 \quad \text{for } j = 1, \dots, k, \\ \alpha_1 &= (P_1^T A P_1)^{-1} (P_1^T r_0), \\ x_1 &= x_0 + P_1 \alpha_1, \\ r_1 &= r_0 - A P_1 \alpha_1. \end{aligned}$$

Then we generate the preconditioned residuals to increase the dimension of the search space:

$$z_2^j = M_j^{-1} r_1 \quad \text{for } j = 1, \dots, k.$$

Let $Z_i = [z_i^1 | \dots | z_i^k]$. We will want to get P_2 from Z_2 by making it A -conjugate to the previous directions:

$$P_2 = Z_2 - P_1 (P_1^T A P_1)^{-1} P_1^T A Z_2,$$

```


$$\begin{aligned}
& p_1^j = z_1^j = M_j^{-1} r_0 \quad \text{for } j = 1, \dots, k, \\
& P_1 = [p_1^1 | p_1^2 | \dots | p_1^k], \\
& \alpha_1 = (P_1^T A P_1)^{-1} (P_1^T r_0), \\
& x_1 = x_0 + P_1 \alpha_1, \\
& r_1 = r_0 - A P_1 \alpha_1, \\
& \text{Repeat } i = 1, 2, \dots \text{ until convergence} \\
& \quad Z_{i+1} = [M_1^{-1} r_i | M_2^{-1} r_i | \dots | M_k^{-1} r_i], \\
& \quad P_{i+1} = Z_{i+1} - \sum_{j=1}^i P_j (P_j^T A P_j)^{-1} P_j^T A Z_{i+1}, \\
& \quad \alpha_{i+1} = (P_{i+1}^T A P_{i+1})^{-1} (P_{i+1}^T r_i), \\
& \quad x_{i+1} = x_i + P_{i+1} \alpha_{i+1}, \\
& \quad r_{i+1} = r_i - A P_{i+1} \alpha_{i+1}, \\
& \text{end}
\end{aligned}$$


```

FIG. 3.1. *The MPCG algorithm.*

or more generally,

$$P_{i+1} = Z_{i+1} - \sum_{j=1}^i P_j (P_j^T A P_j)^{-1} P_j^T A Z_{i+1}.$$

Now, with the new A -conjugate search direction, we can again seek a global minimum that simplifies to a local computation:

$$\begin{aligned}
\alpha_{i+1} &= (P_{i+1}^T A P_{i+1})^{-1} (P_{i+1}^T r_i), \\
x_{i+1} &= x_i + P_{i+1} \alpha_{i+1}, \\
r_{i+1} &= r_i - A P_{i+1} \alpha_{i+1}.
\end{aligned}$$

The algorithm is given in Figure 3.1. Each iteration involves k preconditioner solves. Vector operations in the original PCG are now replaced by operations on $n \times k$ matrices. The matrices $P_j^T A P_j$ are $k \times k$, hence their inversion (for computing P_{i+1} and α_{i+1}) is computationally negligible. We can say, then, that the cost of each MPCG iteration with k preconditioners is approximately k times more expensive than a single PCG iteration, provided that the computational cost of the preconditioner solves is similar.

3.2. Orthogonality properties. Since we do a global minimization of energy norm at the previous steps, we get the usual orthogonality property

$$(3.1) \quad P_i^T r_j = 0 \quad \text{for } i \leq j.$$

From (3.1) it follows that

$$\begin{aligned}
(3.2) \quad r_j^T Z_i &= r_j^T (P_i + P_{i-1} (P_{i-1}^T A P_{i-1})^{-1} P_{i-1}^T A Z_i) \\
&= r_j^T P_i + r_j^T P_{i-1} (\dots) \\
&= 0 + 0 \quad \text{for } i \leq j.
\end{aligned}$$

Up to this point we have derived properties which are valid for any choice of the Z_i 's, e.g., even with random nonsymmetric preconditioners chosen independently at

each step. However, let us now use the fact that our preconditioners are symmetric and do not change throughout the iteration (as opposed to a flexible method). Then

$$r_j^T z_i^s = r_j^T M_s^{-1} r_{i-1} = r_j^T (M_s^{-1})^T r_{i-1} = (M_s^{-1} r_j)^T r_{i-1} = (z_{j+1}^s)^T r_{i-1}.$$

In combination with (3.2), we conclude that

$$(3.3) \quad r_j^T Z_i = 0 \quad \text{for } j \neq i-1.$$

3.3. Breakdown. The MPCG algorithm could break down if $P_j^T A P_j$ is singular for any j , which happens, for example, if two or more of the preconditioners are identical to each other. As previously mentioned, $P_j^T A P_j$ is k -by- k , where k is the number of preconditioners. It is thus a very small matrix whose singularity can be easily detected. Since preconditioners are typically selected so that they produce linearly independent search directions, a breakdown should rarely occur. In any case an automatic way to avoid breakdown may be, for example, to use the singular value decomposition of $P_j^T A P_j$ to eliminate redundant search directions.

3.4. A case where PCG and MPCG are equivalent. Consider the following case of polynomial preconditioning.

PROPOSITION 3.1. *If roundoff errors are ignored, the $2j$ th iteration ($j = 1, 2, \dots$) of PCG with a preconditioner M is identical to the j th iteration of MPCG with $M_1^{-1} \equiv M^{-1}$ and $M_2^{-1} \equiv M^{-1} A M^{-1}$.*

Proof. Assuming an initial guess of $x_0 = 0$ with initial residual $r_0 = b$, the standard PCG algorithm solves for x_{2j} as the vector from the Krylov subspace

$$\mathcal{K}^{2j}(M^{-1}A; M^{-1}r_0) = \text{span}\{M^{-1}r_0, (M^{-1}A)M^{-1}r_0, \dots, (M^{-1}A)^{2j-1}M^{-1}r_0\}$$

that minimizes the energy norm of the error. Let us show that the same subspace is obtained with j iterations of MPCG with M_1 and M_2 . Since $Z_1 = P_1 = [M^{-1}r_0 | M^{-1}AM^{-1}r_0]$ and $r_1 = r_0 - AP_1\alpha_1$, we have $r_1 \in \text{span}\{r_0, AM^{-1}r_0, AM^{-1}AM^{-1}r_0\}$. The corresponding subspace from which x is chosen is spanned by $M^{-1}r_0$ and $M^{-1}AM^{-1}r_0$ in the first iteration, and is extended by $M^{-1}(AM^{-1})^2r_0$ and $M^{-1}(AM^{-1})^3r_0$ in the second iteration. The rest of the proof readily follows by induction, for any given integer j , by repeating the same argument. Since MPCG finds the vector from this subspace that minimizes the energy norm of the error, just as PCG does, it must produce the same iterates as PCG. $\square$

The result in Proposition 3.1 can be extended to more than two preconditioners: MPCG with the k preconditioning matrices $(M^{-1}A)^{j-1}M^{-1}$, $j = 1, \dots, k$, is equivalent in the same fashion as above to PCG with a preconditioner M .

3.5. The recurrence relation. For regular PCG observe that

$$\begin{aligned} p_j^T A z_{i+1} &= (A p_j)^T z_{i+1} \\ &= \left(\frac{r_{j-1} - r_j}{\alpha_j} \right)^T z_{i+1} \\ &= 0 \quad \text{for } j < i. \end{aligned}$$

Thus the A -conjugation step may ignore all but the previous search direction, giving the short recurrence. Unfortunately in MPCG, α is a vector and cannot be inverted, so the above does not easily generalize. However, note that in MPCG, $AP_j\alpha_j = r_{j-1} - r_j$. Using (3.3), this means that for $j < i$

$$(3.4) \quad \alpha_j^T P_j^T A z_{i+1} = 0,$$

and then after expanding α_j^T ,

$$\begin{aligned} ((P_j^T A P_j)^{-1} (P_j^T r_{j-1}))^T P_j^T A Z_{i+1} &= 0, \\ r_{j-1}^T (P_j (P_j^T A P_j)^{-1} P_j^T A Z_{i+1}) &= 0. \end{aligned}$$

We also know $r_s^T (P_j (P_j^T A P_j)^{-1} P_j^T A Z_{i+1}) = 0$ for $s \geq j$ since then $r_s^T P_j = 0$. Thus even if the update $P_j (P_j^T A P_j)^{-1} P_j^T A Z_{i+1}$ is not zero, it is orthogonal to all residuals from r_{j-1} up.

While this orthogonality condition is as close to a short recurrence relation as MPCG generally gets, there is an important case in which it is provably short and there is no error in the truncation: we now formulate and prove this result, beginning with a lemma.

LEMMA 3.2. *Suppose $A = B + C$, where B and C are used as the preconditioners for MPCG. Then the 2×2 matrix $Z_j^T A Z_{i+1}$ is diagonal for $j \neq i + 1$.*

Proof. Write out the columns of each Z matrix and perform the multiplication explicitly:

$$\begin{aligned} Z_j^T A Z_{i+1} &= [B^{-1} r_{j-1} | C^{-1} r_{j-1}]^T (B + C) [B^{-1} r_i | C^{-1} r_i] \\ &= \begin{bmatrix} r_{j-1}^T B^{-1} (B + C) B^{-1} r_i & r_{j-1}^T B^{-1} (B + C) C^{-1} r_i \\ r_{j-1}^T C^{-1} (B + C) B^{-1} r_i & r_{j-1}^T C^{-1} (B + C) C^{-1} r_i \end{bmatrix} \\ &= \begin{bmatrix} (\dots) & r_{j-1}^T (B^{-1} + C^{-1}) r_i \\ r_{j-1}^T (B^{-1} + C^{-1}) r_i & (\dots) \end{bmatrix} \\ &= \begin{bmatrix} (\dots) & r_{j-1}^T (Z_{i+1}^1 + Z_{i+1}^2) \\ r_{j-1}^T (Z_{i+1}^1 + Z_{i+1}^2) & (\dots) \end{bmatrix} \\ &= \begin{bmatrix} (\dots) & 0 \\ 0 & (\dots) \end{bmatrix}, \end{aligned}$$

where the last step uses (3.3). $\square$

We now prove the short recurrence for this $A = B + C$ case, as follows.

THEOREM 3.3. *Suppose $A = B + C$, where B and C are SPD and are used as the preconditioners for MPCG. Then the search directions satisfy the short recurrence relation*

$$(3.5) \quad P_{i+1} = Z_{i+1} - P_i (P_i^T A P_i)^{-1} P_i^T A Z_{i+1}.$$

Proof. To show that the sum in the general A -conjugation formula for P_{i+1} collapses to just the one term as in (3.5), we will prove that $P_j^T A Z_{i+1} = 0$ for $j < i$. We begin our induction argument with the $j = 1$ case.

For $j = 1$, $P_1 = Z_1$. By Lemma 3.2, $P_1^T A Z_{i+1}$ is diagonal. Also recall from (3.4) that $\alpha_1^T (P_1^T A Z_{i+1}) = 0$. We argue that α_1 has all nonzero entries unless $r_0 = 0$: a zero entry would indicate that no energy norm improvement in the solution is possible along the corresponding search direction; i.e., for that column, say the a th column p_1^a of P_1 , we have $r_0^T p_1^a = 0$. But $p_1^a = M_a^{-1} r_0$, so $r_0^T p_1^a$ cannot be zero, assuming that the preconditioners are positive definite. The only diagonal matrix that has a vector with all nonzero entries in its null-space is the zero matrix. Thus $P_1^T A Z_{i+1} = 0$.

Now assume that $P_s^T A Z_{i+1} = 0$ for all $s < j$, and let us work on the case for j .

Substituting the general summation formula for P_j gives

$$\begin{aligned} P_j^T A Z_{i+1} &= \left(Z_j - \sum_{s=1}^{j-1} P_s (P_s^T A P_s)^{-1} P_s^T A Z_j \right)^T A Z_{i+1} \\ &= Z_j^T A Z_{i+1} - \sum_{s=1}^{j-1} Z_j^T A P_s (P_s^T A P_s)^{-1} P_s^T A Z_{i+1}. \end{aligned}$$

Since $s \leq j-1 < i$ in the above sum, the factor $P_s^T A Z_{i+1}$ in each term is zero by induction, so the sum is zero. We are left with $P_j^T A Z_{i+1} = Z_j^T A Z_{i+1}$. Just as in the base case, we know this is a diagonal matrix by Lemma 3.2, and that $\alpha_j^T (P_j^T A Z_{i+1}) = 0$ by (3.4). Also as in the base case note that, by the energy norm minimization property, if an entry of α_j were zero, then it would have to be that the corresponding column of P_j was orthogonal to r_{j-1} . Say that such a column is p_j^a and the corresponding column of Z_j is z_j^a . Using the definition of P_j , we can expand:

$$r_{j-1}^T p_j^a = r_{j-1}^T z_j^a - \sum_{s=1}^{j-1} r_{j-1}^T P_s (P_s^T A P_s)^{-1} P_s^T A z_j^a.$$

Since $s \leq j-1$, the factors $r_{j-1}^T P_s$ are zero by (3.1). We are left with $r_{j-1}^T z_j^a = r_{j-1}^T M_a^{-1} r_{j-1}$, which must be positive if the preconditioners are positive definite. Thus every entry of α_j is nonzero, and so, just as in the base case, we must have that $P_j^T A Z_{i+1}$ is the zero matrix. $\square$

3.6. Truncated MPCG. Theorem 3.3 shows that in certain cases there is no need to use the full MPCG. Even when the recurrence relation is not short, numerical experiments indicate that it often is acceptable to truncate the A -conjugation step to the standard short recurrence (that is, only make P_{i+1} A -conjugate to P_i instead of all previous search directions). The terms we truncate are often very small, and convergence is often not significantly slowed down by the omission. Numerical results that demonstrate this are given in section 4.

While we do not have a full analytical justification for the expectation that a truncated version of MPCG will be effective, some insight may be provided by referring to the result that was obtained earlier in this section, namely that the update $P_j (P_j^T A P_j)^{-1} P_j^T A Z_{i+1}$ is orthogonal to all the residuals from r_{j-1} up.

We define MPCG(m) to be a truncated version of MPCG in which only the last m search directions are used in each iteration. Note that another parameter necessary for defining (full as well as truncated) MPCG is k , the number of preconditioners. However, to maintain simplicity of notation, we do not incorporate it into the definition.

4. Examples. We now present numerical examples and discuss different ideas for how to choose multiple preconditioners.

4.1. ADI examples. We start this section by considering two simple model problems with ADI preconditioning.

4.1.1. Two-dimensional Poisson with weak coupling in one direction. Consider

$$-u_{xx} - \varepsilon u_{yy} = f(x, y)$$

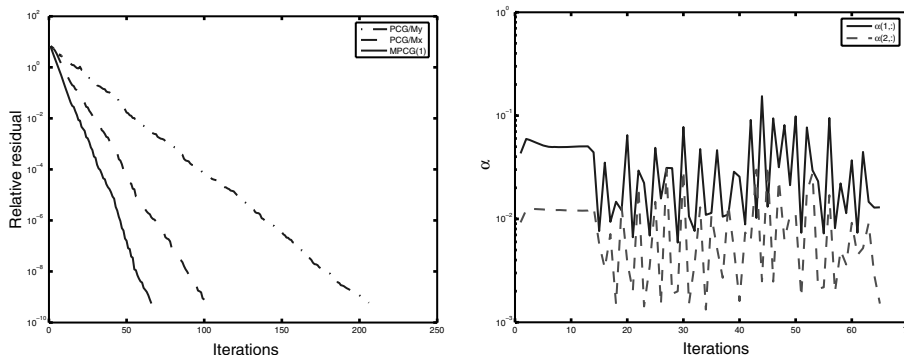


FIG. 4.1. Convergence history for the example in section 4.1.1: Relative residuals and the components of α throughout the MPCG iteration for $\varepsilon = 0.5$.

on $\Omega = (0, 1) \times (0, 1)$, with Dirichlet boundary conditions. We set the right-hand side and the boundary conditions so that $u(x, y) = \cos(\pi x) \cos(\pi y)$ is the exact solution.

Using the standard second order centered difference scheme with n grid points in each direction (that is, with mesh size $h = \frac{1}{n+1}$), the coefficient matrix is $n^2 \times n^2$, given by $A = I_n \otimes T_n + \varepsilon T_n \otimes I_n$, where $T_n = \text{tri}[-1, 2, -1]$. We select two preconditioners in an ADI fashion: $M_x = I_n \otimes T_n$ and $M_y = \varepsilon T_n \otimes I_n$. Thus, M_x is tridiagonal and corresponds to the discretized operator $-u_{xx}$, and M_y is a discrete operator corresponding to $-\varepsilon u_{yy}$. Since $A = M_x + M_y$ by construction, Theorem 3.3 holds and MPCG(1), which is based on short recurrences, produces the same iteration sequence as full MPCG (up to roundoff errors). Setting $\varepsilon = 1$ corresponds to the standard Poisson equation, for which the symmetry between x and y implies that MPCG with M_x and M_y as preconditioners is tied with the standard PCG with either M_x or M_y in terms of overall computational work. (Numerical experiments confirm this.) However, for smaller values of ε the equality of the roles played by x and y is lost, and differences between using M_x and M_y are expected. We take $\varepsilon = 0.5$ and compare the convergence of PCG and MPCG(1). Figure 4.1 illustrates the behavior of α and the convergence of MPCG(1). A 32×32 grid was used. As expected, M_x dominates the search space; the graphs for α confirm this.

Note that each iteration of MPCG(1) involves solving for two preconditioners and hence is more computationally expensive than a PCG iteration by a factor of nearly 2. The iteration counts that are presented in the graphs are 206 for PCG using M_y , 102 for PCG using M_x , and 66 for MPCG(1) using M_x and M_y . Thus, PCG with M_x outperforms MPCG(1), whereas PCG with M_y is inferior.

The point that we are making in this example is that while there might be a single preconditioner whose performance is better than a combination of preconditioners, the detection of the preconditioning for MPCG is done automatically, and does not rely on knowledge of the underlying continuous problem or properties of the matrix. MPCG could be useful in cases where one particular preconditioner clearly dominates but cannot be easily identified beforehand: a few iterations could be executed to determine what the most effective preconditioner is, and then one could switch to regular PCG with that choice.

4.1.2. ADI for three-dimensional Poisson. The three-dimensional Poisson equation, $-\nabla^2 u = f$ on the unit cube with Dirichlet boundary conditions, is discretized using standard centered finite differences and is solved using three precondi-

TABLE 4.1

The Poisson equation in three dimensions, using three preconditioners, in an ADI fashion. The right-hand-side vector in this case was random, with normal distribution.

n	n^3	MPCG (1)	Full MPCG
8	512	32	31
16	4096	61	60
24	13824	88	88

tioners in an ADI fashion: discrete operators that correspond to $-u_{xx}, -u_{yy}, -u_{zz}$. We used a random right-hand side and ran the program for several meshes. The short recurrence relation does not hold in this case; this was observed by keeping track of $P^T AZ$ throughout the iteration. However, numerical experiments indicate that MPCG(1) performs as well as full MPCG. In other words, the convergence behavior is practically unaffected by the truncation. Results are given in Table 4.1. We do not have an analytical explanation for this result.

4.2. Domain decomposition. One of the natural applications for MPCG is domain decomposition: each preconditioner corresponds to (approximately) solving a restriction of the PDE to a subdomain. MPCG will then automatically provide something akin to a coarse grid correction: the matrix equation for α is a Galerkin projection of the matrix onto a small subspace with one degree of freedom per subdomain. This allows for much greater scalability than the corresponding PCG method using just a fixed combination of subdomain solves. Note that the preconditioners for each subdomain will be singular.

For preliminary experiments we solved the standard 5-point Poisson problem on a square grid with Dirichlet boundary conditions. For preconditioners we partition the domain into disjoint rectangles, in each of which we exactly solve the restriction of the problem, i.e., inverting the submatrix of A corresponding to those unknowns. For regular PCG, we assemble these into a standard block diagonal preconditioner. For MPCG, we treat each subdomain solve as a separate preconditioner that can supply a unique search direction.

Our first observation is that if we have just two subdomains, then MPCG apparently (observed to roundoff error) preserves the short recurrence—though this situation is not covered by Theorem 3.3. In this case, MPCG is noticeably more efficient than PCG; e.g., to solve on a 100×100 grid (with 100×50 subdomains) to 10^{-10} relative residual reduction took PCG 49 iterations but MPCG just 37. The cost of these iterations is dominated by the subdomain solves, so the small amount of extra work that MPCG does per iteration is more than compensated for by the enhanced convergence rate. Figure 4.2 shows the residual norm histories.

For more subdomains, the short recurrence property is lost. However, we can see that the full (nontruncated) form of MPCG actually has significantly better scalability than standard PCG, at least in terms of iteration counts. If we keep the subdomain size constant as we increase the grid size, then the iteration count for PCG increases linearly with the side length of the grid. However, for full MPCG, the iteration count appears to increase only logarithmically; see Table 4.2 for the numbers from our numerical experiment.

Unfortunately with truncation—even keeping two or three previous iterations' search directions and not just one—the scalability is diminished and the results are noticeably slower. We conjecture that truncating to one search direction leads to a linear convergence (like PCG, but slower), but retaining more search directions

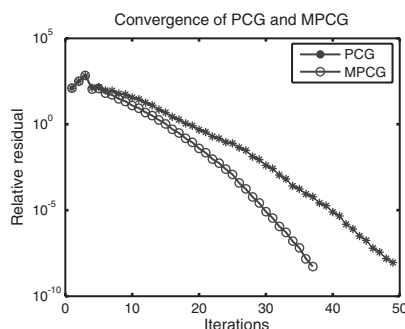


FIG. 4.2. *Residual norm history of PCG and MPCG for a two-dimensional Poisson problem on a 100×100 grid, preconditioned with two 100×50 subdomain solves.*

TABLE 4.2

Iteration counts for a two-dimensional domain decomposition scalability test, using approximately 8×8 subdomain solves for the preconditioners.

Grid size	PCG	Truncated MPCG (1)	Truncated MPCG (2)	Truncated MPCG (3)	Full MPCG
25	39	69	45	44	19
50	70	131	77	67	22
100	126	257	125	107	24

steadily improves the scalability, ultimately towards $O(\log n)$ for full MPCG. For example, MPCG(3) (keeping three previous search direction groups) appears from Table 4.2 to lead to $O(n^{2/3})$ iterations for an $n \times n$ grid.

4.3. A model bending problem. Our motivation for this example is plate and shell elasticity problems, or more generally PDEs where the matrix to solve is the sum of relatively easy to precondition parts (e.g., second order differential operators) and more challenging parts (e.g., fourth order differential operators). We used the standard centered finite difference discretization of

$$B\nabla^4 u - S\nabla^2 u + \frac{1}{\Delta t}u = f$$

with clamped boundary conditions ($u = \frac{\partial u}{\partial n} = 0$) on a unit square domain as a model for an implicit time step in a bending simulation.

The biharmonic term in this problem gives rise to a non- M -matrix and can cause standard incomplete Cholesky methods to break down, though modified incomplete Cholesky, for example, works very well for the other terms. A robust alternative that has been successfully applied to difficult shell problems is stabilized AINV [2]. We investigate using both SAINV (on the full matrix, permuted with a minimum degree ordering, with drop tolerance 0.1) and modified incomplete Cholesky (on all terms except the biharmonic operator, using the regular grid ordering, with level 0 fill) in MPCG.

Our test case uses a 100×100 grid, $B = 10^{-6}$, $\Delta t = 10^{-2}$, and various values for S . Our motivation for these specific choices is related to scaling of the operators. We present iteration counts for PCG with the two different preconditioners as well as their sum (i.e., giving them equal weight), for full MPCG and for truncated MPCG(1) and MPCG(2), in Table 4.3.

TABLE 4.3

Iteration counts (to reduce the residual by 10^{-10}) for a model bending problem. The parameter S is the coefficient that appears in the PDE.

S	SAINV PCG	MIC(0) PCG	Combined PCG	Full MPCG	Truncated MPCG(1)	Truncated MPCG(2)
10^{-3}	63	146	143	48	65	57
10^{-2}	49	126	117	40	47	47
10^{-1}	56	81	70	35	41	41
1	122	58	54	41	54	46
10	171	56	54	44	58	50
10^2	186	56	54	44	61	50

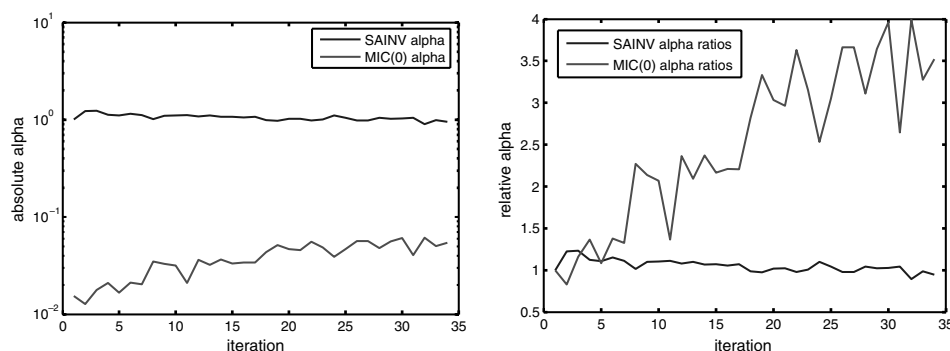


FIG. 4.3. Plots of how the two components of α (one for the SAINV preconditioner and the other for the modified incomplete Cholesky preconditioner) evolve over the iterations for full MPCG applied to the $S = 0.1$ bending problem. The right-hand plot shows the relative change from the initial value of α .

Observe that as the relative importance of the second order term changes, the effectiveness of PCG with a particular choice of preconditioners varies significantly. Meanwhile, full MPCG followed closely by truncated MPCG(2) robustly achieve the minimum iteration counts—though of course doing more work per iteration. For the more imbalanced problems (S very small or very large) it is almost certain that PCG with the appropriate preconditioner will be the clear winner in terms of actual time, but for the more interesting balanced cases—where it is unclear a priori what the appropriate preconditioner is—truncated MPCG could be a very competitive, robust choice.

To illustrate some of the dynamic behavior of MPCG, we plot the two components of α for full MPCG in the $S = 0.1$ problem in Figure 4.3. While the contribution from the SAINV preconditioner remains steady, the contribution from the modified incomplete Cholesky preconditioner steadily grows. We hypothesize this is due to SAINV being more effective overall, but MIC(0) doing a better job on low frequencies—which eventually are all that is left after SAINV deals with the rest of the spectrum. These steadily changing weights could not be duplicated by a fixed combination in regular PCG. Interestingly, we do not see the upwards trend in truncated MPCG: further investigation is required to understand this behavior.

We have observed variations on this problem where MPCG does not fare as well. From these experiments it appears that MPCG usually behaves in one of two ways (excepting the scalable domain decomposition results in the previous section, where we get the coarse grid correction effect). In some problems the multiple preconditioners

act synergistically, and the full and truncated forms of MPCG perform comparably: the additional search directions from the multiple preconditioners more than make up for the loss of global orthogonality and attendant loss of global optimality. For other problems truncated MPCG performs poorly, and while full MPCG necessarily converges in fewer iterations than simple PCG, it appears not to afford a major improvement: the extra search directions are nearly redundant, so PCG with its guarantee of global optimality is more efficient than truncated MPCG.

5. Conclusions. The MPCG method derived in this paper is an algorithm that incorporates multiple preconditioners automatically and obtains optimality in an energy minimization sense. The algorithm constructs a generalized Krylov space whose dimension is proportional to the number of preconditioners incorporated. Short recurrences cannot generally be preserved when more than one preconditioner is involved, but we were able to show analytically (Theorem 3.3) that for two preconditioners whose sum is equal to the coefficient matrix itself, a short recurrence relation holds, and the truncated algorithm MPCG(1) can be used without giving away anything. In addition, we have experimentally observed two situations which our analysis does not cover: in the three-dimensional Poisson equation with three ADI preconditioners, MPCG(1) converges as fast as full MPCG, even though the short recurrence relation does not hold. And in a nonoverlapping domain decomposition test problem the short recurrence holds for two subdomains. When more than two subdomains are applied, the short recurrence is lost, but scalability remains good, as is demonstrated in section 4.2.

In many complicated and large-scale problems, the choice of a preconditioner is not obvious, and if more than one candidate is available, a fixed combination of the preconditioners may not work well enough. This is one place where the mechanism of MPCG may come in handy, since it determines throughout the iteration how to combine the preconditioners. Even if MPCG ultimately is not faster than PCG with the right selection of preconditioner, a few iterations of MPCG may robustly identify what that selection should be.

Parallelism may be another aspect worth exploring. While we have not implemented our algorithm in a parallel environment, it is evident that the time-consuming step $Z_{i+1} = [M_1^{-1}r_i | M_2^{-1}r_i | \dots | M_k^{-1}r_i]$ can be straightforwardly parallelized. There may be useful parallel speed-up even from highly sequential preconditioners: for example, instead of running PCG with incomplete Cholesky, leaving half of a dual-processor workstation sitting idle, two variations on incomplete Cholesky could be run in parallel with MPCG.

Domain decomposition applications naturally lend themselves to an approach such as MPCG, particularly if the physics of one domain is significantly different from the physics of another domain (e.g., due to material interfaces). Singular preconditioners that practically affect only one particular subdomain could be used. Also, a flexible variant of MPCG might prove useful, allowing the preconditioners to vary throughout the iteration, for example, if they are applied to subproblems using PCG with a rough convergence tolerance.

Again recall that in the domain decomposition example our preconditioners were singular. Regular PCG of course cannot tolerate preconditioners whose null-spaces overlap the span of the right-hand side. It is tempting to ask whether we can push this further, with preconditioners that are even slightly indefinite (in different subspaces). A motivating case here is the sparse approximate inverse SPAI [6], whose definiteness is difficult to determine.

Another possible research direction is the derivation of multipreconditioned solvers for other classes of linear systems. In particular, since GMRES [13] does not possess a short recurrence relation, a multipreconditioned version of GMRES might be very competitive.

A MATLAB implementation of the MPCG method is available at [4]. The authors welcome comments and suggestions.

Acknowledgments. We thank Edmond Chow for his modified incomplete Cholesky code, which was used in the example in section 4.3, and an anonymous referee for helpful remarks.

REFERENCES

- [1] F. BANTEGNIES AND C. BREZINSKI, *The Multiparameter Conjugate Gradient Algorithm*, Technical Report 429, Laboratoire d'Analyse Numérique et d'Optimisation, Université des Sciences et Technologies de Lille, Lille, France, 2001.
- [2] M. BENZI, J. K. CULLUM, AND M. TUMA, *Robust approximate inverse preconditioning for the conjugate gradient method*, SIAM J. Sci. Comput., 22 (2000), pp. 1318–1332.
- [3] C. BREZINSKI, *Multiparameter descent methods*, Linear Algebra Appl., 296 (1999), pp. 113–141.
- [4] R. BRIDSON AND C. GREIF, *MPCG: Multi-preconditioned conjugate gradients implemented in MATLAB*, available online at <http://www.cs.ubc.ca/~rbridson/mpcg/>.
- [5] G. H. GOLUB AND Q. YE, *Inexact preconditioned conjugate gradient method with inner-outer iteration*, SIAM J. Sci. Comput., 21 (1999), pp. 1305–1320.
- [6] J. M. GROTE AND T. HUCKLE, *Parallel preconditioning with sparse approximate inverses*, SIAM J. Sci. Comput., 18 (1997), pp. 838–853.
- [7] T. GU, X. LIU, Z. MO, AND X. CHI, *Multiple search direction conjugate gradient method I: Methods and their propositions*, Int. J. Comput. Math., 81 (2004), pp. 1133–1143.
- [8] T. GU, X. LIU, Z. MO, AND X. CHI, *Multiple search direction conjugate gradient method II: Theory and numerical experiments*, Int. J. Comput. Math., 81 (2004), pp. 1289–1307.
- [9] M. R. HESTENES AND E. STIEFEL, *Methods of conjugate gradients for solving linear systems*, J. Res. Nat. Bur. Standards, 49 (1952), pp. 409–436.
- [10] Y. NOTAY, *Flexible conjugate gradients*, SIAM J. Sci. Comput., 22 (2000), pp. 1444–1460.
- [11] D. P. O'LEARY, *The block conjugate gradient algorithm and related methods*, Linear Algebra Appl., 29 (1980), pp. 293–322.
- [12] Y. SAAD, *A flexible inner-outer preconditioned GMRES algorithm*, SIAM J. Sci. Comput., 14 (1993), pp. 461–469.
- [13] Y. SAAD AND M. H. SCHULTZ, *GMRES: A generalized minimal residual algorithm for solving nonsymmetric linear systems*, SIAM J. Sci. Statist. Comput., 7 (1986), pp. 856–869.
- [14] V. SIMONCINI AND D. B. SZYLD, *Flexible inner-outer Krylov subspace methods*, SIAM J. Numer. Anal., 40 (2003), pp. 2219–2239.

EFFECTIVE PRECONDITIONING THROUGH ORDERING INTERLEAVED WITH INCOMPLETE FACTORIZATION*

INGYU LEE[†], PADMA RAGHAVAN[†], AND ESMOND G. NG[‡]

Abstract. Consider an incomplete Cholesky factorization of a sparse symmetric positive definite matrix. It is common practice to start with a symmetric permutation of the rows and columns of the matrix using an ordering to reduce fill in the complete Cholesky factor. Such an approach has two drawbacks. First, such an ordering is designed to reduce fill or operation counts in the complete Cholesky factor, and thus does not necessarily address the issue of controlling fill or operation counts during incomplete factorization. Second, because the ordering is determined in advance of the incomplete factorization, it is not easy to accommodate alternative pivoting sequences, for example, using numeric measures to promote stability, while taking into account their impact on fill and operation counts. In this paper, we consider an alternative approach that interleaves ordering with numeric computation in an incomplete Cholesky factorization. Such an interleaved scheme controls fill and operation counts. It also allows the incorporation of some limited, pivot selection strategies for stability. We demonstrate through experiments that our *interleaved minimum-degree* approach leads to fewer failures during incomplete factorization and improved convergence when the incomplete factor is used as a preconditioner for the conjugate gradient method. We conjecture that the improvements in the quality of preconditioning arise from the ability of our method to generate better approximations to the complete factor.

Key words. preconditioner, incomplete factorization, ordering

AMS subject classifications. 65F50, 65F05, 68N99

DOI. 10.1137/040618357

1. Introduction. Consider the solution of a sparse linear system $Ax = b$ when the matrix A is symmetric and positive definite. A popular iterative solver uses the method of conjugate gradients (CGs) [14] coupled with an incomplete Cholesky (IC) factor $\hat{L}$ as a preconditioner [4]. The matrix $\hat{L}$ is an approximation to the (complete) Cholesky factor L , where $A = LL^T$. Consequently, the process of computing $\hat{L}$ relies to a large extent on data structures and graph-theoretic methods used to compute L in a sparse direct solver.

During sparse Cholesky factorization, some of the elements in A that are zero will fill in and become nonzero in the factor L [9, 12]. An incomplete factor $\hat{L}$ is obtained by performing Cholesky factorization while selectively discarding some fraction of the fill-in nonzeros as soon as they are created. The two main types of IC factorization schemes involve either a symbolic *level-of-fill* approach [21] or a *drop-threshold* approach [22, 32]. For $\hat{L}$ generated from either approach, retaining more fill typically leads to a better preconditioner, i.e., one that can significantly reduce the number of CG iterations and/or prevent failure [23]. When such preconditioners

*Received by the editors November 4, 2004; accepted for publication (in revised form) by V. Simoncini November 22, 2005; published electronically March 17, 2006. This work was funded in part by the National Science Foundation through grants NSF ACI-0102537, NSF EIA-022191, and DMR-0205232 and in part by the Director, Office of Science, Division of Mathematical, Information, and Computational Sciences of the U.S. Department of Energy under contract DE-AC03-76SF00098.
<http://www.siam.org/journals/simax/27-4/61835.html>

[†]Department of Computer Science, The Pennsylvania State University, University Park, PA 16802 (inlee@cse.psu.edu, raghavan@cse.psu.edu).

[‡]Computational Research Division, Lawrence Berkeley National Laboratory, Mail Stop 50F, Berkeley, CA 94720 (EGNg@lbl.gov).

are required for the application, both level-of-fill and drop-threshold schemes benefit from a fill-reducing ordering scheme.

The fill incurred during Cholesky factorization depends on the sparsity structure of A and not on the numerical values of the elements. This fact is exploited in almost all sparse direct solvers. Numerical factorization is preceded by the ordering step, during which a symmetric permutation of A is computed using graph-theoretic methods to limit the number of nonzeros in the corresponding L . There has been some work on using Markowitz diagonal pivoting at each step of an incomplete factorization [13, 22] and on computing orderings tailored for IC factorization [5, 6]. However, typical incomplete factorization schemes tend to use orderings computed to reduce fill in the complete sparse Cholesky factor L .

Currently, level-of-fill (symbolic) and drop-threshold (numeric) incomplete factorization schemes use fill-reducing ordering schemes as follows. The ordering step is the first symbolic step (as in a direct solver) to compute a fill-reducing permutation; this permutation is directed toward reducing fill in the complete sparse Cholesky factor L and not in its incomplete form $\hat{L}$. In a subsequent step, strategies are incorporated to select elements to retain or discard in order to obtain $\hat{L}$ as an approximation to the sparse Cholesky factor L of the permuted matrix A . Thus, in broad terms, for a given permutation π of A to reduce fill in $L(\pi)$, both symbolic and numeric forms of incomplete factorization compute $\hat{L}(\pi)$ as an approximation to the complete Cholesky factor $L(\pi)$ with an error matrix $E(\pi) = L(\pi) - \hat{L}(\pi)$.

This paper focuses on the role of fill-reducing orderings in computing effective IC preconditioners. Consider computing a permutation τ using an ordering scheme that is specifically directed toward reducing fill in the incomplete factor $\hat{L}(\tau)$ corresponding to a level-of-fill or drop-threshold scheme. Now there exists a complete sparse Cholesky factor $L(\tau)$ for this permutation of A with an error matrix $E(\tau) = L(\tau) - \hat{L}(\tau)$. This approach could potentially lead to a better approximation, and thus a more effective preconditioner, than an approximation from the traditional approach; i.e., the norm of the error matrix $E(\tau)$ could be smaller than that of $E(\pi)$, where τ and π are constructed to reduce fill in $\hat{L}(\tau)$ and $L(\pi)$, respectively.

Orderings to explicitly reduce fill in the incomplete factor $\hat{L}$ could be potentially constructed using band/envelope or profile methods such as reverse Cuthill–McKee (RCM) [12], greedy minimum-degree-like heuristics [1, 11, 18, 24, 28], and certain hybrid methods [2, 30]. The focus of our paper is on a natural formulation in terms of greedy, minimum-degree-like orderings which can be easily interleaved with the incomplete factorization process.

The key idea in this paper is coupling the symbolic fill-reducing ordering and the numerical incomplete factorization steps to construct improved preconditioners. In effect, the computations from the two steps are interleaved at each stage of the elimination to construct $\hat{L}$. The ordering at each step locally reduces the number of operations in the rank-1 update, and indirectly, the fill in $\hat{L}$. Our results with a large test-suite of sparse matrices indicate that such an ordering for $\hat{L}$ improves its efficacy as a preconditioner and enables its construction with fewer breakdowns.

An outline of the paper is as follows. Section 2 contains notation and background. Section 3 contains a description of our main algorithm and its variants. Section 4 contains extensive empirical results which indicate that our new methods perform significantly better than the traditional schemes. Section 5 attempts to explain the improved performance of our methods, and section 6 contains some concluding remarks.

2. Background. In this section, we provide a brief background on IC preconditioners and their application to accelerate the convergence of the method of CGs. We start with an overview of fill-reducing ordering schemes and their role in determining the structure of the sparse Cholesky factor L and its incomplete form $\widehat{L}$. We then continue with an overview of the two main types of incomplete factorization schemes and end with a short review on re-factorizing after applying diagonal shifts if the factorization process were to breakdown.

Henceforth, we use L to denote the sparse Cholesky factor corresponding to a specific fill-reducing permutation of the coefficient matrix A . We use $\widehat{L}$ to denote its incomplete form. For any matrix B , we use $B_{*,i}$ to denote its i th column and $B_{i,*}$ to denote its i th row.

2.1. Fill-reducing orderings and the sparsity structure of L . A fill-reducing ordering of A is a symmetric permutation of its rows/columns typically computed using symbolic heuristics. Such heuristics are applied either to the graph of A or to a series of elimination graphs representing the evolving structure of A as a symbolic Cholesky factorization proceeds. These methods are thus based on a graph model of Cholesky factorization [25, 26]; more details can be found in classic books on sparse direct methods [9, 12].

As mentioned earlier, ordering schemes consist of methods from three broad classes (band/envelope, greedy, and nested-dissection) and their hybrids. Of particular relevance to this paper is the class of greedy minimum-degree-like orderings [1, 11, 18, 24, 28]. The minimum-degree algorithm was first proposed by Tinney and Walker [31] and is the symmetric analogue of the Markowitz ordering scheme [20]. The algorithm minimizes the number of operations in the rank-1 update at each step of a right-looking sparse Cholesky factorization. An efficient implementation, “multiple minimum degree” (MMD), is due to Liu [18]; Amestoy, Davis, and Duff [1] have developed “approximate minimum degree” which uses an approximation to the degree to reduce ordering time. A related ordering heuristic, also proposed by Tinney and Walker [31], is the minimum deficiency algorithm, which minimizes the number of fill entries *introduced* at each step of sparse Cholesky factorization. More recently, efficient analogues of this scheme have also been developed and they tend to reduce the fill and operation count better than minimum degree at the expense of a slight penalty in ordering time [24, 28].

Fill-reducing orderings can affect the quality of an incomplete factorization preconditioner. It has been generally observed that for very sparse $\widehat{L}$, envelope-type orderings, such as RCM, may be superior to minimum degree and its variants in accelerating the convergence of CGs [8]. However, for $\widehat{L}$ with more fill-in, minimum degree performs quite well and is generally the method of choice for a cache-efficient blocked implementations of IC factorization [23].

2.2. Orderings for $\widehat{L}$. Earlier research on developing ordering schemes to improve the quality of the preconditioner $\widehat{L}$ was based on numeric metrics. D’Azevedo, Forsyth, and Tang [5, 6] developed a greedy ordering scheme using numeric metrics related to the norm of the discarded portion of the rank-1 update matrix in a right-looking implementation of incomplete factorization. Their metric allowed the explicit selection, as the next column to be numbered, of a column with the minimum discarded fill in its rank-1 update. This metric can be viewed as a numeric analogue of the “minimum deficiency” symbolic metric of Tinney and Walker [31]; however, it is related purely to the quality of preconditioning and not to the sparsity of $\widehat{L}$. The

“minimum discarded fill” method was successful in significantly improving the quality of the preconditioner, but the associated computational overheads were observed to be significant [5, 6].

Munksgaard [22] described an incomplete factorization pivoting strategy and stated that a minimum-degree ordering during factorization improves the quality of a preconditioner when fill-in was allowed. The technique was implemented as *MA31* in the Harwell library. In our paper, we consider a similar approach. We achieve the dynamic ordering for incomplete factorization by interleaving the factorization with heuristic fill-reducing greedy ordering and explore the quality of the resulting preconditioner from various sparse matrix test-suite applications. To further improve the quality of the preconditioner, we add consistent tie-breaking strategies which enhance the robustness of our algorithm. When a matrix loses positive definiteness during an incomplete factorization, we employ a diagonal shift to compute an incomplete factorization of a nearby matrix. Both level-of-fill and drop-threshold strategies are implemented in our algorithm.

2.3. Types of IC factorization. The basic step in IC factorization is that of computing $\hat{L}$, an incomplete form of the (complete) sparse factor L for a given fill-reducing permutation π of A . There are two broad types of methods based on either symbolic or numeric measures.

The level-of-fill incomplete factorization is best described using the graph model of sparse Cholesky factorization [25, 27]. It uses a symbolic metric that characterizes the nonzero structure of L and thus of $\hat{L}$. Consider L , the Cholesky factor corresponding to a fill-reducing ordering π . If $L_{i,j} \neq 0$, then there exist one or more paths from j to i in the graph of the permuted A with intermediate vertices that are numbered less than $\min(i, j)$ [12, 25, 27]. Such a path is known as a *fill-path*. Define as $\text{length}(p)$ the number of intermediate vertices in p , a fill-path between vertices j and i . Let $\mathcal{F}_{i,j}$ be the set of fill-paths between j and i . Now the level of fill of the nonzero $L_{i,j}$ is defined as $\min_{p \in \mathcal{F}_{i,j}} \{\text{length}(p)\}$.

In a level-of-fill k IC factor $\hat{L}$, only those nonzeros in L that have levels of fill less than or equal to k are retained. Thus, $\hat{L}$ corresponding to a level-of-fill 0 has exactly the same nonzero structure as the original A permuted according to π . For larger k , more fill-in is retained in $\hat{L}$ which typically makes $\hat{L}$ a more effective preconditioner. Henceforth, we refer to this scheme as ICF (incomplete Cholesky level of fill), and, when necessary, as $\text{ICF}(k)$, where k is a small integer denoting the level of fill.

A drop-threshold IC uses a numeric approach. The columns of $\hat{L}$ are computed from left to right much as in complete sparse Cholesky factorization by accumulating updates from earlier columns, computing the square root of the diagonal element, and then scaling the subdiagonal elements by the diagonal element. However, nonzero entries with magnitude smaller than a given threshold are discarded [22, 32]. There are several ways to define and use the threshold. For example, the threshold can be a nonnegative value or a nonnegative factor relative to some measure of the size of A (such as the norm of A or $\max_{i,j} |A_{i,j}|$). Other restrictions can also be imposed, such as discarding from each column all but some m nonzero entries satisfying the threshold condition [29], where m is typically a small constant such as 5 or 10. Henceforth, we refer to such schemes as ICT (incomplete Cholesky with drop threshold), and, when necessary, as $\text{ICT}(\omega)$, where ω is the drop threshold (a small positive real number). It is customary to start ICT factorization after diagonally scaling the original matrix to obtain unit diagonal values ($D^{-1/2} \times A \times D^{1/2}$); this allows for easier application

of the threshold criterion, and a suitable rescaling can be applied when using $\hat{L}$ to accelerate CG.

Observe that for both ICF and ICT the ordering scheme dictates the filled structure of L , and thus indirectly controls the form of the incomplete factor $\hat{L}$.

2.4. Diagonal shifts on IC failures. In both ICF and ICT factorizations, a central problem is that of breakdowns when a diagonal element $\hat{L}_{i,i}$ becomes close to zero or negative. The simplest approach to this is replacing the diagonal with a small positive constant and then proceeding with the incomplete factorization. However, in practice this can lead to poor-quality preconditioners. Another approach proposed by Manteuffel for handling such failures is to consider the incomplete refactorization of the shifted matrix $A + \alpha I$ [19]. Now a key question is that of determining the proper value of the shift α ; ideally, it should be the smallest magnitude shift required to prevent breakdowns. If the shift is large, then it can affect the quality of preconditioning through $\hat{L}$. Additionally, the shift may also affect the sparsity structure (and hence memory requirements) of $\hat{L}$ in an ICT scheme. A general strategy is to start with a small α which is then iteratively increased if a breakdown occurs [16, 19].

3. An interleaved minimum-degree ordering. We now develop our scheme for interleaving ordering and incomplete factorization (ICF or ICT) based on the greedy minimum-degree metric of Tinney [31]. We henceforth refer to our ordering as “interleaved minimum degree” (IMD). The ordering τ computed by IMD is specifically designed to reduce the number of operations in computing the incomplete factor $\hat{L}$, whereas, in the traditional scheme, the ordering π is computed to reduce the number of operations to compute L .

Our method uses a purely symbolic, degree-like metric to control the number of operations in IC and the fill incurred in $\hat{L}$. In addition, we use a simple numeric measure to break ties among columns with the same degree metric. We refer to our IMD scheme in conjunction with a level-of-fill ICF(k) as IMDF; likewise, we refer to our IMD scheme in conjunction with a drop-threshold ICT as IMDT.

Consider a complete sparse Cholesky factorization of an $N \times N$ matrix when the algorithm proceeds column by column from left to right. The i th factorization step ($i = 1, \dots, N-1$) is described by the following equations starting with $H_1 = A$; I_p is the $p \times p$ identity matrix, v_i is an $N-i$ vector of the subdiagonal elements in the first column of the current H_i , and H_{i+1} is the remaining $(N-i) \times (N-i)$ submatrix:

$$H_i = \begin{bmatrix} d_{ii} & v_i^T \\ v_i & B_{i+1} \end{bmatrix} = L_i \begin{bmatrix} 1 & 0 \\ 0 & H_{i+1} \end{bmatrix} L_i^T,$$

$$L_i = \begin{bmatrix} \sqrt{d_{ii}} & 0 \\ v_i/\sqrt{d_{ii}} & I_{n-i} \end{bmatrix}, \text{ and } H_{i+1} = B_{i+1} - v_i v_i^T / d_{ii}.$$

In IC, L_i and the remaining submatrix H_{i+1} at the i th step are obtained by discarding some elements from v_i and some elements from the rank-1 update by v_i . Consequently, for any i , $1 \leq k \leq N$, IC uses an approximation of the form $\hat{H}_i = H_i - E_i$, where E_i is the error matrix corresponding to values dropped from v_i and H_{i+1} .

Our algorithm is but a natural extension of the original minimum degree to reduce the number of operations associated with the rank-1 update from the current column in a right-looking IC scheme. In IMD, ordering is interleaved with numeric computation. Consider the first step of the algorithm. The column (and, because of symmetry, the row) to be numbered i is selected as the one with minimal degree in the

current $\widehat{H}_i$. In the second phase, this column is factored and subdiagonal nonzeros in the column are dropped according to the level-of-fill or drop-threshold criterion. Additionally, the rank-1 update is performed with this column; if the current column is i and $\widehat{H}_{j,i} \neq 0$ is to be retained according to the level-of-fill or drop-threshold criterion, column i in $\widehat{H}$ is used to update column j . Finally, the degree metric is updated for all such neighboring columns j of i . Observe that j is the original column number in A , and, although this column has not yet been numbered in the IMD ordering, it will be assigned a number higher than i .

Our ordering scheme with a symbolic degree metric is fully interleaved with the incomplete factorization process. Henceforth, we use $\widehat{L}$ to denote the matrix at the i th step from an algorithmic viewpoint starting with $\widehat{L}$ initialized to A . The first i columns of this matrix are the computed columns of the final preconditioner $\widehat{L}_i$, while its rightmost $n - i$ submatrix corresponds to $\widehat{H}_{i+1}$. Figure 1 contains a schematic of our basic algorithm using this $\widehat{L}$ notation.

An additional detail involves retaining or dropping an element of the form $\widehat{L}_{p,j}$, arising from an update to column j by the current column i in which the elements $\widehat{L}_{j,i}$ and $\widehat{L}_{p,i}$ are retained. As explained above, j and p are original column numbers in A ; these columns have not yet been numbered in IMD but they will eventually be numbered higher than i . In an ICF(k) implementation, we apply the update and retain $\widehat{L}_{p,j}$ only if the level (in terms of fill-paths as discussed earlier) of this update is less than or equal to k , i.e., it arises from a fill-path of length less than or equal to k . In an ICT(ω) scheme, $\widehat{L}_{p,j}$ is retained if $|\widehat{L}_{p,j}| \geq \omega \sqrt{\widehat{L}_{j,j} \widehat{L}_{p,p}}$.

Now, we apply the update and retain $\widehat{L}_{p,j}$ only if it satisfies the threshold condition at the current point in the factorization process. We believe that our approach allows us to limit the amount of intermediate storage required in a right-looking IC implementation while allowing a consistent application of the level-of-fill or drop-threshold criterion.

In all greedy schemes, tie-breaking among columns with the same value of the metric can be of considerable importance [11]. We view tie-breaking as a potential opportunity to improve the quality of the preconditioner $\widehat{L}$ by using some suitable numeric measure that can be computed inexpensively as part of our IMD scheme. Henceforth, IMD refers to the formulation in which ties are broken arbitrarily.

Our tie-breaking schemes are based on the magnitude of the diagonal elements of columns with the same minimal value of the degree metric. The first variant we consider is motivated by the idea of stability through pivot selection. If two or more columns have the same minimal value of the metric, then the column whose diagonal value has the largest magnitude will be numbered next. We refer to this scheme as IMD(F/T)-BD (for big diagonal).

Our second tie-breaking scheme reflects a “greedy” attempt to potentially reduce breakdowns during incomplete factorization; we refer to this scheme as IMD(F/T)-SD (for small diagonal). Recall that during Cholesky factorization (whether it is incomplete or not), the diagonal values can get smaller in magnitude through subsequent rank-1 updates as $L_{i,i} = A_{i,i} - \sum_{p < i, L_{i,p} \neq 0} \{L_{i,p}^2\}$. Thus, to prevent potential breakdowns we break ties by selecting the column with the smallest diagonal among those with minimal degree. This scheme is also partly motivated by certain structural developments that occur in the course of minimum-degree orderings [11] for L . Although our scheme is for $\widehat{L}$, we still expect such behavior (albeit less pronounced). More specifically, the vertices in the filled graph, i.e., the graph of $L + L^T$, can be

```

 $\widehat{L}, flag, \alpha] = \text{IMD}[A, \alpha_0]$ 
Apply diagonal scaling, i.e.,  $A = D^{-1/2} \times A \times D^{1/2}$ 
 $N \leftarrow$  dimension of  $A$ 
 $\alpha \leftarrow \alpha_0$ 
while  $\widehat{L}$  has not been computed do
  Initialize  $\widehat{L}$  to  $A$ 
  Mark all columns “unnumbered”
  Set flag to “continue”
  while there are “unnumbered” columns and flag is “continue” do
    Select a column with minimum degree (from all unnumbered columns of  $\widehat{L}$ );
    number it  $i$ .
    Compute column  $\widehat{L}_{*,i}$ .
    if  $\widehat{L}_{i,i} \leq 0$  then
      Shift diagonal by  $\alpha$  and set  $\alpha \leftarrow 2 \times \alpha$ 
      Set flag to “breakdown”
    else
      for all remaining neighboring unnumbered columns,  $j$  do
        Retain/drop  $\widehat{L}_{j,i}$  using level-of-fill/drop-threshold criterion.
      end for
      for all remaining neighboring unnumbered columns,  $j$  do
        Update  $\widehat{L}_{*,j}$  if  $\widehat{L}_{p,j} \neq 0$ 
        Drop/retain  $\widehat{L}_{p,j} \neq 0$  using level-of-fill/drop-threshold criterion, where
         $i < p \leq n$ .
      end for
    end if
  end while
end while

```

FIG. 1. The IMD algorithm.

partitioned into groups of vertices. Two or more *indistinguishable* vertices are fully connected to each other and, in addition, they have the same set of neighbors. Thus, ordering among them does not affect sparsity. However, it does have the potential to affect the numerical process; for example, if four columns p, q, r, s are indistinguishable, then the first to be numbered will result in updates to the remaining three. Now if the column with the smallest diagonal value is numbered highest among the four, it may suffer a breakdown as its diagonal value gets reduced by updates from the three columns numbered earlier. IMD-SD attempts to prevent such breakdowns by numbering the column with the smallest diagonal value first.

As the factorization proceeds, the diagonal value of the current column may cease to be a nonzero positive value. In such a case, we apply a diagonal shift of α and reapply our algorithm to the shifted matrix $A + \alpha I$.

The extra costs of implementing the ordering interleaved with the factorization correspond to those of finding columns with minimum degree and performing the degree update. To support the latter, in our IMD with arbitrary tie-breaking we use a heap data-structure. If we assume that the maximum number of neighbors of a column is bounded above by a small constant, then this structure will at worst add an overhead of $O(N \log N)$ for a matrix with N columns. However, for IMD-SD and IMD-BD with numeric tie-breaking, the simple heap data-structure is no longer

appropriate because it does not provide an easy mechanism for identifying all columns with the same degree metric. Now, we use a combination of heaps and queues. The heap is based on the degree values; each heap-node represents a specific degree value and is in turn linked to a queue containing all columns with that value of the degree metric. Tie-breaking involves traversing the queue associated with the minimum degree. We also maintain a “reverse look-up index” per column to efficiently identify the degree queue to which it belongs; this allows associated degree updates to occur in constant time.

In summary, our method is a column-by-column implementation of minimum degree to locally reduce the number of operations in successive rank-1 updates in an IC factorization. The implementation is indeed more complex than for fixed-fill methods such as those described in [16, 15]. Furthermore, the computational costs of preconditioner construction in IC methods that allow more fill beyond the number of nonzeros in A can be substantially higher than for fixed-fill IC methods. However, the increased fill in the preconditioner often results in fewer CG iterations and enables the reliable solution of less well-conditioned problems using PCG without incurring the higher memory costs of a sparse direct solver. In the next section, we empirically demonstrate how our IMD schemes improve upon traditional ICF and ICT methods, allowing a range of fill-in to construct preconditioners that can meet the requirements of a wider range of applications.

4. Empirical results. In this section, we report on the results of our experiments for evaluating the performance of our IMD schemes. A key aspect concerns comparisons with (i) the minimum discarded fill (MDF) scheme for computing ICF preconditioners [5, 6], and (ii) the more traditional ICF and ICT preceded by the MMD and the (RCM) orderings [10] for the complete Cholesky factor L .

Henceforth, we use the following labels for the set of schemes considered in this paper. The traditional forms use an ordering method (MMD or RCM) followed by IC using either level-of-fill (ICF) or drop-threshold methods (ICT). We refer to these schemes as MMD-ICF, MMD-ICT, RCM-ICF, and RCM-ICT. As noted above we use MDF to denote the minimum discarded fill scheme, which is suitable only for ICF preconditioners, and for our interleaved minimum degree schemes, we use labels of the form IMD (for the ordering) followed by F or T (level-of-fill or drop-threshold, respectively) and by SD or BD to indicate tie-breaking using the small diagonal (SD) or big diagonal (BD) rules (as opposed to arbitrary tie-breaking). Our ICF forms are thus IMDF (arbitrary tie-breaking), IMDF-SD, and IMDF-BD. Similarly, we label our ICT counterparts IMDT, IMDT-BD, and IMDT-SD.

Quality of preconditioners: MDF and IMD schemes. Our comparisons with MDF are somewhat limited because the code is no longer available [7]. We therefore developed a simple implementation of MDF in C (as suggested by the authors of MDF) to compare the quality of the preconditioner for a set of six matrices (`bccstk14`, `bcsstk16`, `s1rmt3m1`, `s2rmq4m1`, `bodyy5`, `fdm1`) representative of our larger test-suite described in Table 1. MDF orderings are known to be significantly expensive to compute [5, 6]. Furthermore, our simple C implementation is not tuned for the best performance and is not suitable for comparing execution times.

Our experiments concern the quality of preconditioners computed by MDF and our IMDF schemes. MDF is suitable only for ICF, and we consider ICF with four levels of fill ($k = 0, 1, 2$, and 3) for each method. Observe that even if the levels of fill are the same, MDF and IMDF variants can lead to differences in the number of nonzeros in the preconditioner. We therefore report on the number of iterations

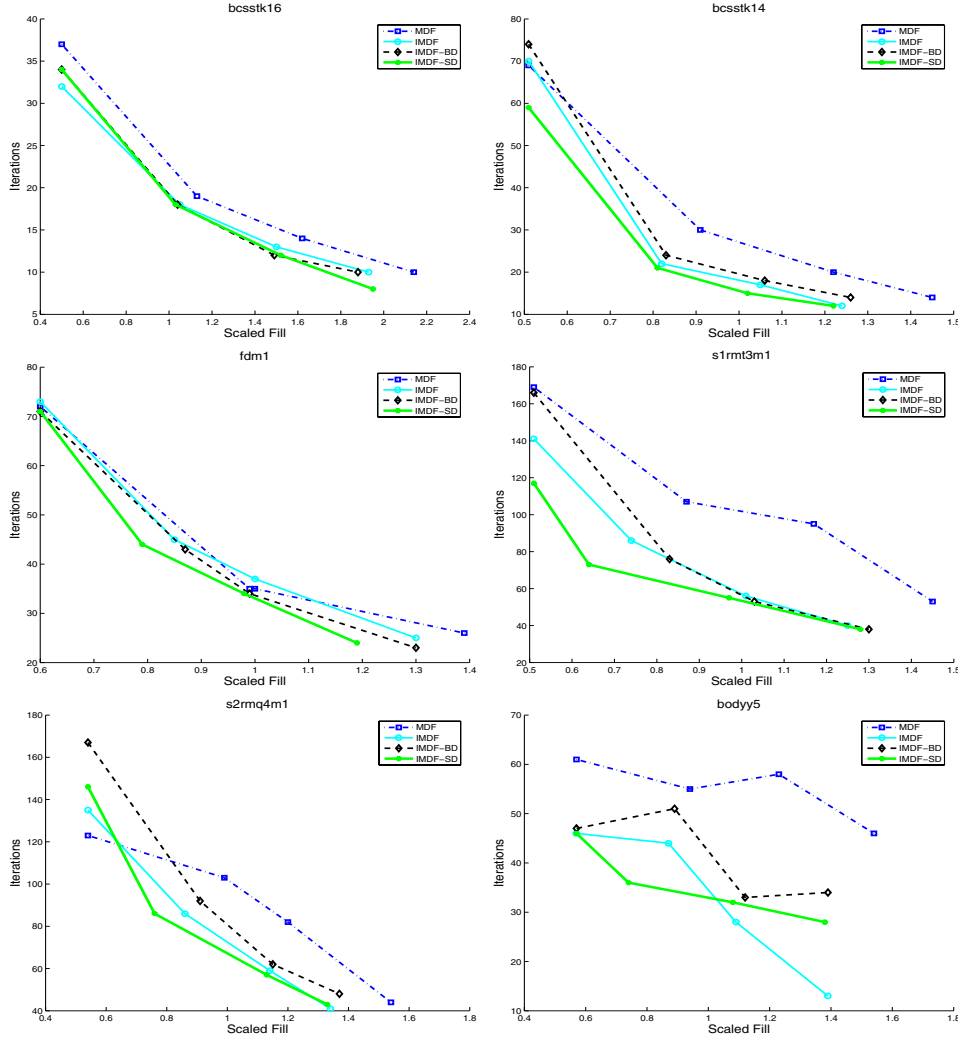


FIG. 2. Quality of preconditioners using MDF and IMDF, IMDF-BD, and IMDF-SD schemes for levels of fill 0, 1, 2, and 3. The Y-axis indicates the number of CG iterations and the X-axis indicates the number of nonzeros in the incomplete factor relative to $|A|$.

of CG (required upon preconditioning through MDF, IMDF, IMDF-BD, and IMDF-SD) as a function of the number of nonzeros in the incomplete factor (reported as the “scaled fill” $\frac{|\hat{L}|}{|A|}$). These results are shown in Figure 2; the plots clearly indicate that in most instances, our IMDF schemes produce higher quality preconditioners for approximately equal amounts of fill in the incomplete factor.

Evaluation of traditional ICF/ICT and IMD schemes. We have used a test-suite of 34 matrices shown in Table 1 for our experiments to comparatively evaluate the performance of our three IMD schemes and the traditional incomplete factorization with MMD and RCM orderings. Our experiments include ICF with four levels of fill ($k = 0, 1, 2$, and 3) and ICT with four threshold values ($\omega = 0.1, .05, .01$, and $.001$). In summary, we report on two types of preconditioning (ICF and ICT), each with

TABLE 1

Characteristics of test matrices (* indicates matrices that are used to compute overhead in section 4).

Matrix	Rank	A	Description
1138_bus	1,138	2,596	Power system networks
bcsstk19	817	3,855	Part of a suspension bridge
bcsstk08	1,074	7,107	TV studio
bcsstk09	1,083	9,760	Square plate clamped
bcsstk10	1,086	11,578	Buckling of a hot washer
bcsstk11	1,473	17,857	Ore car—lumped mass
bcsstk14*	1,806	32,630	Roof of the Omni Coliseum
bcsstk16*	4,884	147,631	U.S. Army Corps of Engineers dam
bcsstk18	11,948	80,519	R.E. Ginna Nuclear Power Station
msc00726	726	17,623	Structural engineering matrix from MSC/NASTRAN
msc01050	1,050	15,103	Structural engineering matrix from MSC/NASTRAN
msc01440	1,440	23,856	Structural engineering matrix from MSC/NASTRAN
msc04515	4,515	51,111	Structural engineering matrix from MSC/NASTRAN
msc10848	10,848	620,313	Structural engineering matrix from MSC/NASTRAN
msc23052	23,052	588,934	Structural engineering matrix from MSC/NASTRAN
s1rmt3m1	5,489	112,505	Cylindrical shell, 30×30 triangular mesh
s1rmq4m1	5,489	143,300	Cylindrical shell, 30×30 quadrilateral mesh
s2rmq4m1	5,489	143,300	Cylindrical shell, 30×30 quadrilateral mesh
s2rmt3m1	5,489	112,505	Cylindrical shell, 30×30 triangular mesh
s3rmq4m1	5,489	143,300	Cylindrical shell, 30×30 quadrilateral mesh
s3rmt3m1	5,489	112,505	Cylindrical shell, 30×30 triangular mesh
s3rmt3m3	5,357	106,526	Cylindrical shell, graded mesh with 1666 triangles
fdm1*	6,050	17,979	Five-point stencil, Poisson equations, Dirichlet boundary conditions
fdm2*	32,010	95,629	Computational chemistry problem
bodyy4*	17,546	69,548	From NASA, collected by Alex Pothen
bodyy5*	18,589	73,721	From NASA, collected by Alex Pothen
bodyy6*	19,366	76,787	From NASA, collected by Alex Pothen
5K*	5,658	217,236	Electromagnetic simulations in accelerator design
26K*	26,148	1,053,456	Electromagnetic simulations in accelerator design
56K*	56,196	11,722,070	Electromagnetic simulations in accelerator design
qa8fm	66,127	863,353	Mass matrix for 3D acoustic applications
qa8fk*	66,127	863,353	Finite element stiffness matrix for 3D acoustic applications
s3dkq4m2	90,449	2,455,670	Cylindrical shell, 150×100 quadrilateral mesh
s3dkt3m2	90,449	1,921,955	Cylindrical shell, 150×100 triangular mesh

four states of incompleteness and 34 problems for each method. Thus, there are 272 tests per method with a total of 1,360 tests over all five schemes.

In the first stage of our experiments, we compute the incomplete factor $\hat{L}$ after diagonal scaling of the original A . When breakdowns occur in this stage, we use the diagonal shift scheme of Lin and More [16]. We refactor with an initial shift $\alpha = 10^{-3}$; subsequent breakdowns result in refactoring with a doubling of the shift value α . If α becomes larger than the diagonal element in A , we conclude that the method failed to compute the incomplete Cholesky preconditioner. In the second stage of our experiments, we solve for four different right-hand-side vectors using $\hat{L}$ as a preconditioner for CG. We use the row-sum vector as one of the right-hand-side vectors (with three other random vectors) and a stopping criterion of 10^{-6} residual norm or maximum iterations of 1,000. A method fails to solve the linear system if the desired tolerance is not achieved within the maximum number of iterations.

To evaluate the quality of preconditioners in accelerating the convergence of CG,

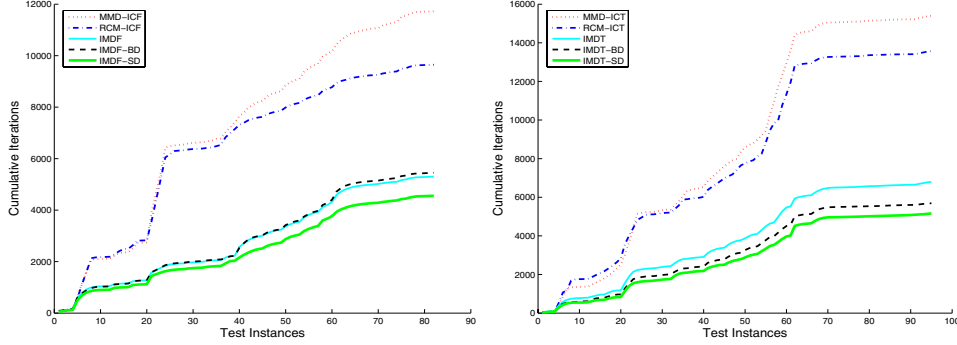


FIG. 3. Cumulative iterations for ICF schemes (left) and ICT schemes (right). In each figure, the plots from top to bottom correspond to MMD (dotted line), RCM (dashed-dotted line), IMD (thin gray line), IMD-BD (dashed line) and IMD-SD (thick gray line).

we consider the following three metrics:

- Iterations: the number of iterations for CG to converge using a row-sum vector of A for the right-hand-side vector.
- Scaled fill: defined as the number of nonzeros in the preconditioner relative to the number of nonzeros in the original matrix (the whole matrix and not just the symmetric half), i.e., $\frac{|\hat{L}|}{|A|}$.
- Operations: defined as the product “scaled fill $\times$ iterations”, a measure of the number of operations required to apply the preconditioner to solve for a single right-hand-side vector.

In the rest of this section we report on our empirical evaluation of the five methods for ICF and ICT preconditioners. We have pairs of plots, one for ICF and one for ICT with *cumulative* values of a performance metric along the Y-axis for the range of test instances. The test instances correspond to matrices in the order in which they are listed in Table 1; for each matrix, there are at most four tests for levels of fill 0, 1, 2, and 3 for ICF and for threshold values ω from 0.1, .05, .01, and .001 for ICT. If any one of the five methods fails in either the first IC stage or in the second stage during CG iterations, we exclude that instance from the figures and tables in this section. We also provide summaries of performance metrics using totals and geometric means for average values.

Figure 3 shows the performance of ICF and ICT preconditioners with test instances along the X-axis and *cumulative* “iterations” along the Y-axis. In order to take into account the sparsity of the preconditioner when evaluating its effectiveness using the “iterations” metric, we show cumulative “operations” in Figure 4. The final value of each plot is the total number of iterations or operations for a specific method over all test instances. The method with the least number of iterations and operations is the best from this perspective. We observe that our IMD schemes perform considerably better than the traditional schemes. Furthermore, IMDF-SD and IMDT-SD, the variants with tie-breaking using the small-diagonal, perform the best.

Summary results on the performance of the five methods (geometric means of all successful instances) are shown in Table 2. These summaries show that IMD variants are quite similar with respect to controlling fill and that the SD tie-breaking scheme results in fewer iterations and operations. More specific observations include the following:

- Overall, ICT preconditioners appear to outperform ICF preconditioners.

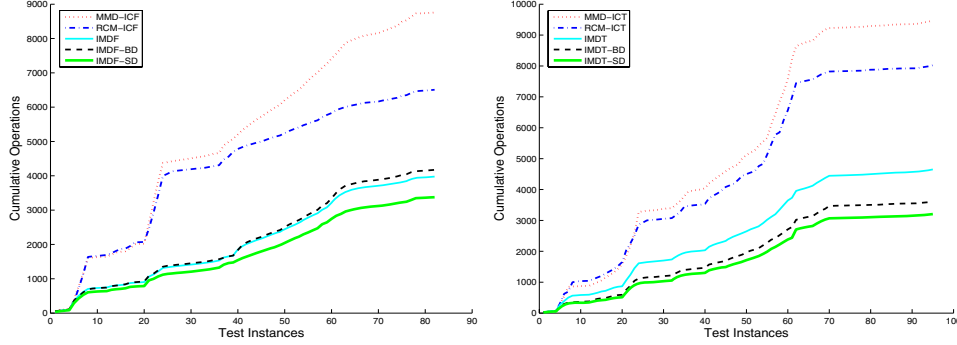


FIG. 4. Cumulative operations (scaled fill $\times$ iterations) for ICF schemes (left) and ICT schemes (right). In each figure, the plots from top to bottom correspond to MMD (dotted line), RCM (dashed-dotted line), IMD (thin gray line), IMD-BD (dashed line) and IMD-SD (thick gray line).

TABLE 2

Geometric mean values of “scaled fill,” “iterations,” and “operations.” In the lower half of the table, values are shown relative to RCM-ICF (set at 1) for ICF tests and relative to RCM-ICT (set at 1) for ICT instances; a value smaller than 1 indicates that a method performed better than RCM-ICF or RCM-ICT.

Metric	Geometric mean scaled fill, iterations, and operations									
	ICF (level-of-fill)					ICT (drop-threshold)				
	IMDF			MMD	RCM	IMDT			MMD	RCM
	SD	BD	–			SD	BD	–		
Scaled Fill	0.89	0.92	0.91	0.93	0.81	0.67	0.68	0.75	0.67	0.65
Iterations	32	37	35	60	49	25.7	28.2	34.3	53.6	48
Operations	28.8	34.8	32.5	56.4	40.3	17.4	19.4	25.9	36.2	31.2
Metric	Geomeans relative to RCM-ICF					Geomeans relative to RCM-ICT				
	IMDF	BD	–	MMD	RCM	IMDT	BD	–	MMD	RCM
	SD	BD	–			SD	BD	–		
Scaled Fill	1.09	1.13	1.12	1.14	1.0	1.04	1.05	1.16	1.04	1.0
Iterations	0.65	0.76	0.72	1.22	1.0	0.53	0.58	0.71	1.11	1.0
Operations	0.71	0.86	0.8	1.4	1.0	0.56	0.62	0.83	1.16	1.0

- On comparing ICF schemes, IMDF-SD achieves
 - 87% fewer iterations and 95% fewer operations than MMD-ICF at 14% less scaled fill.
 - 53% fewer iterations and 40% fewer operations than RCM-ICF with 9% more scaled fill.
- On comparing ICT schemes, IMDT-SD achieves
 - 108% fewer iterations and 108% fewer operations than MMD-ICT at nearly equal scaled fill.
 - 87% fewer iterations and 78% fewer operations than RCM-ICF with 3% more scaled fill.

These results are indeed very promising and demonstrate the potential of our methods for effective preconditioning to accelerate the convergence of CG.

Execution times. We now consider the time needed for computing the preconditioner and applying it to solve one and four linear systems. As described earlier, the first stage of the computation includes diagonal scaling, ordering, and incomplete numeric factorization. This stage also includes refactorizations after diagonal shifting until the preconditioner computation succeeds or fails; the latter occurs when the maximum allowable diagonal shift α is exceeded. We henceforth refer to the execution time of this stage as that for computing the preconditioner $\hat{L}$. The second stage

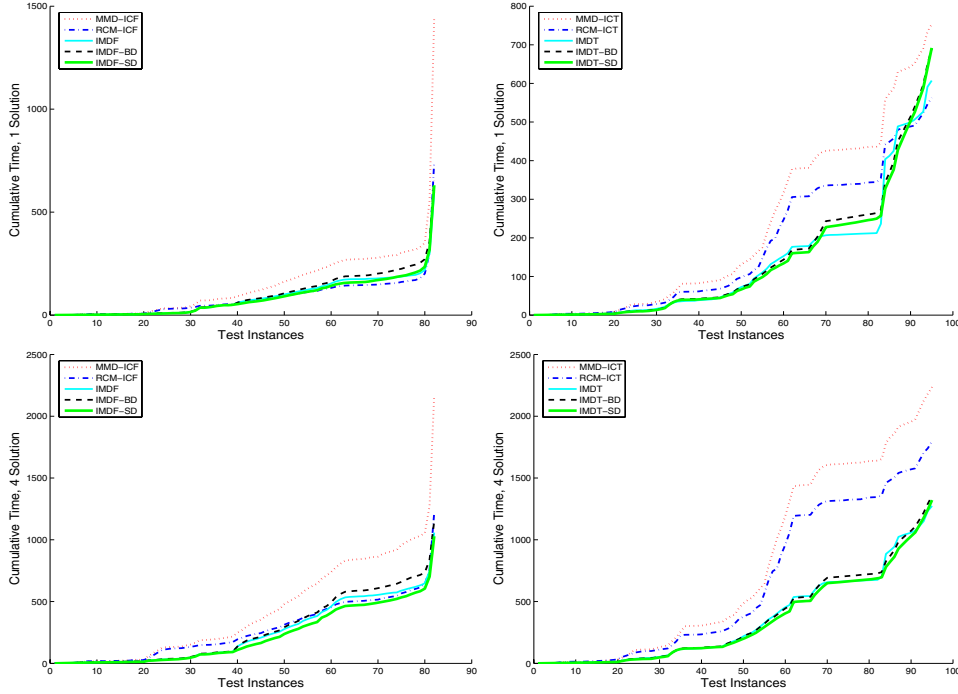


FIG. 5. Cumulative execution times for computing $\hat{L}$ and solutions. The top pair shows the results for one solution and the bottom pair for four solutions. In each pair, the figure to the left corresponds to the scheme for ICF and the right for ICT. In each figure, the plots from top to bottom correspond to MMD (dotted line), RCM (dashed-dotted line), IMD (thin gray line), IMD-BD (dashed line) and IMD-SD (thick gray line).

of computation concerns applying the preconditioner to solve for a given right-hand-side vector using CG. We consider a total of four different right-hand-side vectors, including three selected at random, and the row-sum vector.

We report cumulative times over all test instances for preconditioner construction and one solution, and preconditioner construction and four solutions, in Figure 5. Summary results for preconditioner construction, application, and times for one and four solutions are shown in Table 3. The top half of Table 3 provides geometric means of actual execution times, while the bottom half provides geometric means for ICF and ICT schemes relative to RCM-ICF and RCM-ICT, respectively. Not surprisingly, preconditioner construction time for IMDF-SD (IMDT-SD) is significantly higher than that for RCM-ICF (RCM-ICT); IMDF-SD is nearly 179% slower than RCM-ICF and IMDT-SD is approximately 411% slower than RCM-ICT. However, this higher preconditioner cost leads to an improved preconditioner with fewer CG iterations, and the total solution time is nearly equal to that of RCM-ICF and RCM-ICT. However, when preconditioner construction time is amortized over solutions for four right-hand-side vectors, IMDF-SD requires 15% less time than RCM-ICF, while IMDT-SD requires 46% less time than RCM-ICT. Our IMD variants require significantly less time than MMD-ICF/ICT to set up $\hat{L}$ and to apply the preconditioner. This improvement is a consequence of the fact that our IMD schemes require fewer refactorizations after applying diagonal shifts and result in better preconditioners.

Observe that the time for solving for four systems is typically much greater than the time for computing $\hat{L}$ for all methods. Thus, the performance benefits of our IMD

TABLE 3

Geometric mean times for computing $\hat{L}$, applying the preconditioner at CG iterations for one right-hand-side (rhs) vector, and the total times for one and four solutions. Relative values (with respect to RCM-ICF/ICT) less (greater) than 1 indicate that the method performed faster (slower) than RCM-ICF/ICT.

Geometric mean execution times										
	ICF (level-of-fill)					ICT (drop-threshold)				
	IMDF			MMD	RCM	IMDT			MMD	RCM
	SD	BD	–			SD	BD	–		
$\hat{L}$	0.36	0.4	0.12	0.34	0.13	0.49	0.49	0.21	0.26	.09
1 rhs	0.69	0.8	0.74	1.26	0.93	0.74	0.8	0.8	1.48	1.33
$\hat{L}+1$ rhs	1.47	1.66	1.31	2.25	1.43	1.65	1.73	1.32	2.11	1.64
$\hat{L}+4$ rhs	3.9	4.46	3.91	6.72	4.54	4.15	4.41	3.98	6.79	5.76
Geomeans relative to RCM-ICF						Geomeans relative to RCM-ICT				
$\hat{L}$	2.79	3.11	1.37	2.63	1.0	5.11	5.18	2.27	2.78	1.0
1 rhs	0.74	0.86	0.8	1.36	1.0	0.56	0.6	0.6	1.11	1.0
$\hat{L}+1$ rhs	1.02	1.16	0.92	1.58	1.0	1.0	1.05	0.8	1.28	1.0
$\hat{L}+4$ rhs	0.85	0.98	0.86	1.48	1.0	0.72	0.76	0.69	1.17	1.0

schemes should be more dramatic when the cost of computing the preconditioner is amortized over a longer sequence of solutions for different right-hand-side vectors.

In the context of these ICT/ICF preconditioners, a natural question concerns the relative fraction of execution time spent in nonnumeric, symbolic operations, i.e., ordering time excluding the time for setting up and maintaining the data structures used for $\hat{L}$. For the larger matrices in Table 1 (those with a * next to their names), we report the time for such nonnumeric operations as a fraction of the total time for a single solution, including preconditioner construction and its application to the CG iterations. In the traditional ICF/ICT scheme preceded by MMD, this relative fraction corresponds to the time for ordering. With respect to the total solution time using MMD-ICF/T, the fraction spent on nonnumeric computations was 0.8%, on average. Likewise, with respect to the total solution time using RCM-ICF/T, the nonnumeric fraction is 0.4%. With respect to the total solution time for IMDF/T-SD, the nonnumeric fraction is significantly larger at 12%, on average. It is indeed interesting to observe that even with this overhead, the overall solution time for IMDF/T is competitive with RCM-ICT/F and more than twice as fast as MMD-ICT/F due to improvements in the quality of preconditioners.

Failures and diagonal shifts. Recall that if failures occur during the computation of $\hat{L}$, we modify A as $A + \alpha I$ and recompute $\hat{L}$. Successive shifts double α and a method fails to compute $\hat{L}$ if α exceeds the maximum allowable shift value [16]. The actual shift value used before $\hat{L}$ is computed successfully can affect the fill incurred. The shift value also affects the quality of preconditioning; it is common for a larger shift to result in slower convergence or breakdowns during PCG.

Our interleaved IMD orderings appear to be remarkably effective in computing $\hat{L}$ with smaller shifts on average and a very small number of failures, as shown in Table 4. For very sparse $\hat{L}$, such as level-of-fill 0 and drop-threshold $\omega = 0.1$, all methods need several shifts. However, for higher fill instances, such as level-of-fill 3 and drop-threshold $\omega = .001$, the number of shifts on average for RCM and MMD methods is 4.25 to 5.6 times that of IMD-SD. The need for fewer shifts for IMD methods translates into fewer breakdowns in the process of computing $\hat{L}$. Additionally, as the final diagonal shift is smaller, the preconditioning quality is affected less adversely, translating into fewer breakdowns during PCG. The number of failures shown in Table 4 indicate that our IMD methods are remarkably robust when compared to

TABLE 4

Average number of diagonal shifts (left) and number of failures (right). Failure percentages are based on 272 test cases per method using the row-sum right-hand-side vector.

Method	Average number of diagonal shifts									Number of failures				
	ICF				ICT				ICF & ICT	IC		PCG		IC & PCG
	k				ω					k	ω	k	ω	
	0	1	2	3	0.1	.05	.01	.001						Sum (%)
MMD	3.7	3.9	2.9	2.8	5.2	4.9	5.2	4.5	4.1	11	15	15	20	61 (22)
RCM	3.5	3.9	2.4	2.5	4.7	5.0	4.5	3.4	3.7	11	17	10	14	52 (19)
IMD-SD	3.0	1.6	0.8	0.5	3.1	3.2	2.3	0.8	1.9	0	0	0	0	0 (0)
IMD-BD	3.0	1.2	0.8	0.5	3.0	3.3	2.2	0.8	1.8	0	0	0	0	0 (0)
IMD	3.0	1.5	0.7	0.6	4.2	3.7	2.7	1.1	2.2	0	0	0	0	0 (0)

the traditional RCM/MMD-based ICT and ICF. RCM-based (MMD-based) ICF and ICT methods have failure rates of approximately 19% (22%) over both stages for our test instances, while IMD and its SD and BD variants have no failures. Thus, using the degree (structural) metric for ordering $\hat{L}$ in IMD methods appears to indirectly improve the numerical properties of $\hat{L}$.

5. Toward reducing discarded fill. As shown in the preceding section, our IMD method (and its variants based on numerical metrics for tie-breaking) results in preconditioners that are significantly more effective than traditional IC using MMD and RCM orderings. We now provide some conjectures on the underlying mechanism that could potentially explain this improved performance.

Consider the error matrix E defined as the difference matrix between the sparse complete Cholesky factor L and the IC factor $\hat{L}$ using the same ordering. For the traditional ICF/ICT with the ordering π computed prior to numerical factorization, we consider the difference matrix $E(\pi) = L(\pi) - \hat{L}(\pi)$. For an ordering τ computed by our IMD schemes, there exists a complete sparse Cholesky factor $L(\tau)$ corresponding to this permutation of A with the error matrix $E(\tau) = L(\tau) - \hat{L}(\tau)$. Such an error matrix can be used as a measure of the quality of the preconditioner [3]. We conjecture that our IMD methods on average result in smaller error matrices, and thus better approximations to the corresponding true sparse Cholesky factors, i.e., $|E(\tau)| \leq |E(\pi)|$, where τ and π reduce fill in $\hat{L}(\tau)$ and $L(\pi)$, respectively. In Figure 6, we show the 2-norm of the error matrix for three matrices over all methods. This value is consistently smaller for IMD and its SD and BD variants. IMD-SD has the smallest error matrix, and it typically results in a preconditioner leading to the smallest number of PCG iterations.

Assume that the improved performance of IMD and its variants arises from the ability of such methods to compute incomplete factorizations that are close to complete factorizations; that is, the errors in the incomplete factors are small. Now this attribute must be a consequence of the minimum-degree metric applied to IC. We conjecture that in IMD, the minimum-degree metric serves as a structural (or symbolic) measure of the *discarded* matrix at each stage of the right-looking process. Note that because the degree metric is the number of subdiagonal nonzeros satisfying the ICF/ICT criterion at the current stage, IMD automatically selects a column of degree m such that m^2 , the number of operations in the corresponding rank-1 update, is minimal (and no more than m^2). This quantity m^2 also represents the number of elements in the rank-1 update matrix from which some fraction will be dropped. This fraction is a structural measure of the discarded matrix at each stage.

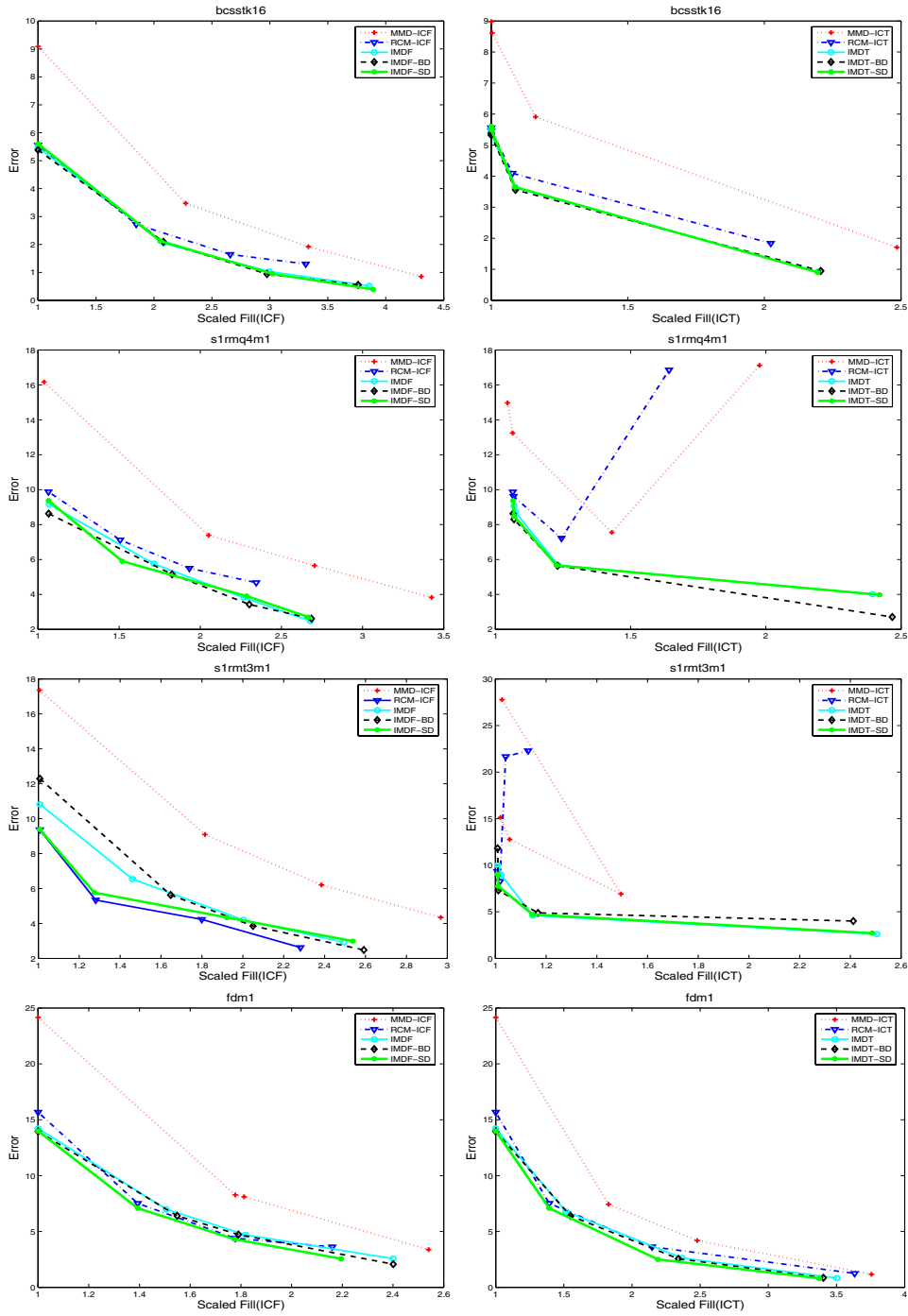


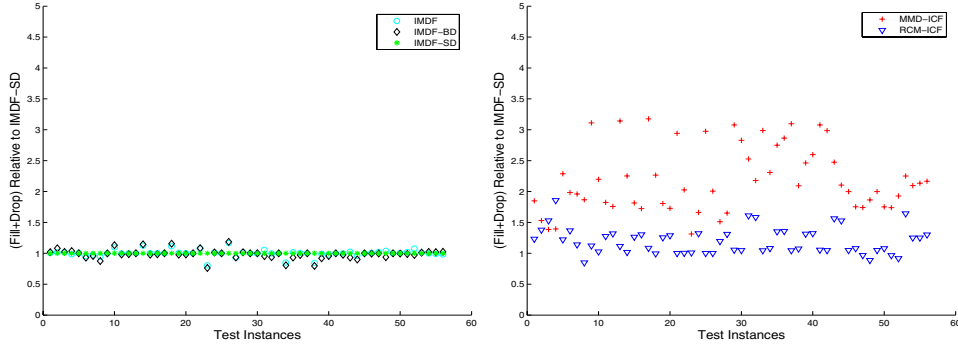
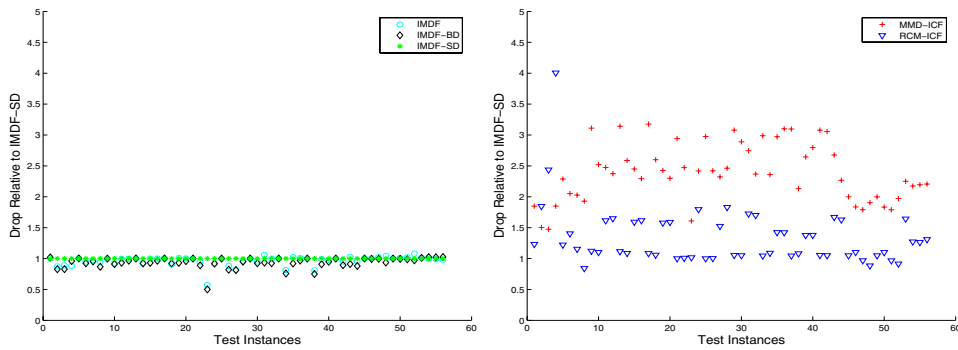
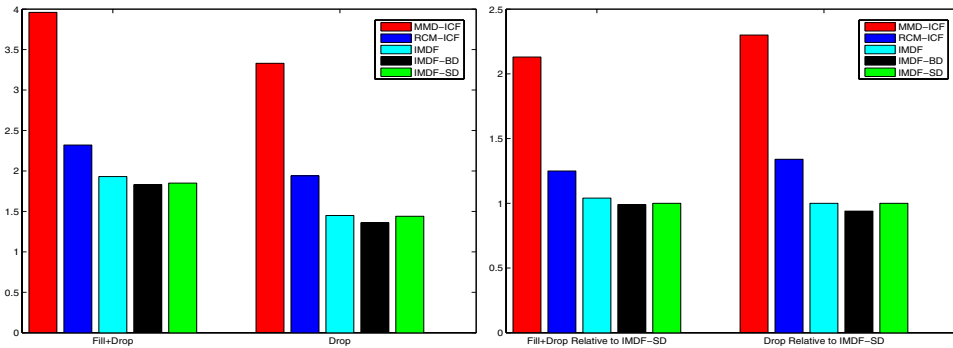
FIG. 6. Sizes of the error matrix (2-norm) for test matrices *bcsstk16*, *s1rmq4m1*, *s1rmt3m1*, and *fdm1* for ICF (left) and ICT (right) schemes.

We expect the minimal degree in IMD to be small when compared with MMD for the complete sparse Cholesky factor L . In the case of a complete sparse Cholesky factor, even for near optimal fill orderings of model sparse matrices from five-point finite-difference discretizations on a two-dimensional $\sqrt{N} \times \sqrt{N}$ grid, the minimum degree will grow to $O(\sqrt{N})$ [17]. However, for such problems, most IC schemes would limit the number of nonzeros in any column of $\hat{L}$ to within a small constant factor of those in A . This limited growth in minimum degree will be exactly computed and used in our IMD schemes. By using the actual degree in $\hat{L}$, IMD can more effectively control the sparsity of $\hat{L}$ and limit the size of the discarded submatrix at each rank-1 update. We thus conjecture that IMD results in a more *economical* incomplete factorization process that indirectly helps to reduce a structural measure of the errors incurred at each step through dropping.

To attempt to substantiate our claim, we consider ICF factorizations and compute Φ , the amount of fill-in relative to $|\hat{L}|$, and Δ , the total number of dropped elements relative to $|\hat{L}|$, for each test matrix for levels of fill 0 through 3. The number of diagonal shifts can affect both Φ and Δ values and we therefore report values only for instances that required no diagonal shifting for all methods. Figures 7 and 8 show $\Phi + \Delta$ and Δ values normalized by those for IMDF-SD (set at 1 for each instance). The average values of $\Phi + \Delta$ and Δ are shown on the left in Figure 9; averages of the values relative to IMDF-SD are shown on the right. On average MMD-ICF (RCM-ICF) requires 100% (12%) more $\Phi + \Delta$ than the IMD schemes. Considering only Δ (as a structural measure of the discarded/error matrix), MMD-ICF on average discards 84% more elements than IMDF, while RCM-ICF discards 27% more elements than IMDF.

If the degree metric in IMD does indeed serve as a structural measure to reduce the discarded fill, then the improved quality of our preconditioners would be consistent with the results reported in [5, 6]. The latter consider MDF orderings for level-of-fill IC. MDF computes an ordering by explicitly minimizing the numeric error at each step in a right-looking IC factorization. The authors of [5, 6] report that the scheme produces higher-quality preconditioners than traditional ICF using MMD or RCM at the expense of significant overheads incurred for computing the MDF ordering. As reported in section 4, in most instances, our IMDF methods produce IC preconditioners of higher-quality than MDF. The fact that MDF and our IMD methods produce higher-quality preconditioners than the traditional approaches could be a consequence of reduced discarded fill. Further improvements in the quality of preconditioners produced by our IMD methods could be a consequence of improved tie-breaking and the ability of IMD and its variants to effectively limit fill in $\hat{L}$.

6. Conclusions. In this paper, we have developed a class of new IMD schemes for effective preconditioning through interleaving of greedy minimum-degree ordering and incomplete factorization. Our methods perform significantly better than traditional MMD-IC with respect to both the time required for computing the preconditioner and its effectiveness in accelerating the convergence of CG. Empirical results indicate that on average RCM-IC (MMD-IC) requires 72–110% (124–152%) more operations than our IMD-SD (with SD tie-breaking) to solve a linear system using PCGs. This improved quality of preconditioning is achieved at the cost of increased time to compute the preconditioner; on average IMD (with no tie-breaking) requires approximately 6% (20%) more time than RCM-ICF (RCM-ICT), the fastest class of methods of computing $\hat{L}$. Tie-breaking increases the time to 44% (80%) more than RCM-ICF (RCM-ICT). However, this extra cost of computing the preconditioner can

FIG. 7. Filled and dropped elements relative to $|\hat{L}|$ ($\Phi + \Delta$) during ICF.FIG. 8. Dropped elements relative to $|\hat{L}|$ (Δ) during ICF.FIG. 9. Summary means of Φ (fill+drop) and Δ (drop) are shown on the left; means relative to IMDF-SD are shown on the right. In each group of bar plots, the methods are ordered from left to right as MMD-ICF, RCM-ICF, IMDF, IMDF-BD, and IMDF-SD.

be justified if it can be amortized over a sequence of solutions for different right-hand-side vectors. Our experiments indicate that when total time is considered, including preconditioner setup and its application to solve four systems, IMD-SD requires the least time (on average). Thus, the overall performance of our IMD schemes should be significantly better than MMD-IC or RCM-IC when the preconditioner is computed once and used to solve a larger number of linear systems. Our IMD schemes also

appear to be considerably more robust than the traditional schemes with respect to computing the preconditioner.

All the methods we have considered in this paper, including our IMD schemes and traditional ICF/ICT preceded by fill-reducing orderings, allow flexible preconditioning through variations either in the level of fill or in the threshold value. These schemes require more complicated data-structures than fixed-fill incomplete factorizations [16, 15], their computational costs can grow because of the increased fill-in, and their parallelization can be more difficult. However, these methods enable effective preconditioning on uniprocessors for systems from a wider variety of application domains by increasing the reliability of the solution process. These methods also remain memory-scalable, as shown in Table 2; on average over all levels of fill and thresholds, the scaled fill values are under 1. More important, even though preconditioner construction costs may be higher, overall solution time can be lower when such costs are amortized over a sequence of solutions. These trade-offs are revealed even more clearly when one considers the IMD schemes developed in this paper. The results in Tables 3 and 4 indicate that our IMDT-SD preconditioners are relatively expensive to compute but provide highly reliable solutions with smaller diagonal shifts to avoid breakdowns during IC, which enable, faster convergence of CG iterations.

IMD orderings reduce the number of operations in the rank-1 update of a right-looking IC factorization. We conjecture that this process results in a preconditioner $\hat{L}$, which may be a better approximation to the complete Cholesky factor L , thus providing more effective preconditioning. We further conjecture that such better approximations are achieved by using the degree metric. A smaller degree leads to a smaller upper bound in the number of elements that could be discarded at each rank-1 update, thus serving as a structural measure for limiting the error incurred at each step of the factorization. This conjecture is somewhat substantiated by our experiments, which indicate that the norm of the error matrix is smaller and that (on average) the number of drops incurred using IMDF is substantially smaller than for MMD-ICF and RCM-ICF.

REFERENCES

- [1] P. R. AMESTOY, T. A. DAVIS, AND I. S. DUFF, *An approximate minimum degree ordering algorithm*, SIAM J. Matrix Anal. Appl., 17 (1996), pp. 886–905.
- [2] C. ASHCRAFT AND J. W. H. LIU, *Robust ordering of sparse matrices using multisection*, SIAM J. Matrix Anal. Appl., 19 (1998), pp. 816–832.
- [3] O. AXELSSON AND N. MUNKSGAARD, *Analysis of incomplete factorizations with fixed storage allocation*, in Preconditioning Methods: Analysis and Applications, D. Evans, ed., Gordon and Breach, New York, 1983, pp. 219–241.
- [4] P. CONCUS, G. GOLUB, AND D. O’LEARY, *A generalized conjugate gradient method for the numerical solution of elliptic partial differential equations*, in Sparse Matrix Computations, J. R. Bunch and D. J. Rose, eds., Academic Press, New York, 1976, pp. 309–332.
- [5] E. F. D’AZEVEDO, P. A. FORSYTH, AND W.-P. TANG, *Ordering methods for preconditioned conjugate gradient methods applied to unstructured grid problems*, SIAM J. Matrix Anal. Appl., 13 (1992), pp. 944–961.
- [6] E. D’AZEVEDO, P. FORSYTH, AND W. TANG, *Towards a cost-effective ILU preconditioner with high level fill*, BIT, 32 (1992), pp. 442–463.
- [7] E. D’AZEVEDO, P. FORSYTH, AND W. TANG, *On the Availability of Minimum Discarded Fill Ordering Codes*, Personal communication, 2004.
- [8] I. DUFF AND G. MEURANT, *The effect of ordering on preconditioned conjugate gradients*, BIT, 29 (1989), pp. 635–637.
- [9] I. S. DUFF, A. M. ERISMAN, AND J. K. REID, *Direct Methods for Sparse Matrices*, Oxford University Press, Oxford, UK, 1986.

- [10] J. LIU, E. CHU, A. GEORGE, AND E. NG, *SPARSEPAK: Waterloo Sparse Matrix Package User's Guide for SPARSEPAK*, Tech. Report CS-84-36, Department of Computer Science, University of Waterloo, Waterloo, Ontario, Canada, 1984.
- [11] A. GEORGE AND J. W. H. LIU, *The evolution of the minimum degree ordering algorithm*, SIAM Rev., 31 (1989), pp. 1–19.
- [12] A. GEORGE AND J. W. H. LIU, *Computer Solution of Large Sparse Positive Definite Systems*, Prentice-Hall, Englewood Cliffs, NJ, 1981.
- [13] NUMERICAL ALGORITHMS GROUP, *The NAG C Library Manual, Mark 7*, Oxford, UK, 2002.
- [14] M. R. HESTENES AND E. STIEFEL, *Methods of conjugate gradients for solving linear systems*, J. Research Nat. Bur. Standards, 49 (1952), pp. 409–436.
- [15] M. T. JONES AND P. E. PLASSMANN, *Blocksolve95 Users Manual: Scalable Library Software for the Parallel Solution of Sparse Linear Systems*, Tech. Report ANL-95/48, Argonne National Laboratory, Argonne, IL, 1995.
- [16] C.-J. LIN AND J. J. MORÉ, *Incomplete Cholesky factorizations with limited memory*, SIAM J. Sci. Comput., 21 (1999), pp. 24–45.
- [17] R. J. LIPTON, D. J. ROSE, AND R. E. TARJAN, *Generalized nested dissection*, SIAM J. Numer. Anal., 16 (1979), pp. 346–358.
- [18] J. W. H. LIU, *Modification of the minimum degree algorithm by multiple elimination*, ACM Trans. Math. Software, 11 (1985), pp. 141–153.
- [19] T. A. MANTEUFFEL, *An incomplete factorization technique for positive definite linear systems*, Math. Comput., 34 (1980), pp. 473–497.
- [20] H. M. MARKOWITZ, *The elimination form of the inverse and its application to linear programming*, Management Sci., 3 (1957), pp. 255–269.
- [21] J. A. MEIJERINK AND H. A. VAN DER VORST, *An iterative solution method for linear systems of which the coefficient matrix is a symmetric M-matrix*, Math. Comp., 31 (1977), pp. 148–162.
- [22] N. MUNKSGAARD, *Solving sparse symmetric sets of linear equations by preconditioned conjugate gradients*, ACM Trans. Math. Software, 6 (1980), pp. 206–219.
- [23] E. NG, B. PEYTON, AND P. RAGHAVAN, *A blocked incomplete cholesky preconditioner for hierarchical-memory computers*, in Iterative Methods in Scientific Computation IV, D. R. Kincaid and A. C. Elster, eds., IMACS Ser. Comput. Appl. Math. 5, International Association for Mathematics and Computers in Simulation, Rutgers University, New Brunswick, NJ, 1999, pp. 211–221.
- [24] E. G. NG AND P. RAGHAVAN, *Performance of greedy ordering heuristics for sparse Cholesky factorization*, SIAM J. Matrix Anal. Appl., 20 (1999), pp. 902–914.
- [25] S. PARTER, *The use of linear graphs in Gauss elimination*, SIAM Rev., 3 (1961), pp. 119–130.
- [26] D. J. ROSE, *A graph-theoretic study of the numerical solution of sparse positive definite systems of linear equations*, in Graph Theory and Computing, R. C. Read, ed., Academic Press, New York, 1972, pp. 183–217.
- [27] D. J. ROSE, R. E. TARJAN, AND G. S. LUEKER, *Algorithmic aspects of vertex elimination on graphs*, SIAM J. Comput., 5 (1976), pp. 266–283.
- [28] E. ROTHBERG AND S. C. EISENSTAT, *Node selection strategies for bottom-up sparse matrix ordering*, SIAM J. Matrix Anal. Appl., 19 (1998), pp. 682–695.
- [29] Y. SAAD, *ILUT: A dual threshold incomplete factorization*, Numer. Linear Algebra Appl., 1 (1994), pp. 387–402.
- [30] J. SCHULZE, *Towards a tighter coupling of bottom-up and top-down sparse matrix ordering methods*, BIT, 41 (2001), pp. 800–841.
- [31] W. F. TINNEY AND J. W. WALKER, *Direct solutions of sparse network equations by optimally ordered triangular factorization*, in Proc. IEEE, 55 (1967), pp. 1801–1809.
- [32] Z. ZLATEV, *Use of iterative refinement in the solution of sparse linear systems*, SIAM J. Numer. Anal., 19 (1982), pp. 381–399.

ON THE SENSITIVITY OF SOME SPECTRAL PRECONDITIONERS*

L. GIRAUD^{†‡} AND S. GRATTON[†]

Abstract. It is well known that the convergence of the conjugate gradient method for solving symmetric positive definite linear systems depends to a large extent on the eigenvalue distribution. In many cases, it is observed that “removing” the extreme eigenvalues can greatly improve the convergence. Several preconditioning techniques based on approximate eigenlements have been proposed in the past few years that attempt to tackle this problem. The proposed approaches can be split into two main families, depending on whether the extreme eigenvalues are moved exactly to one or are shifted to close to one. The first technique is often referred to as the deflation approach, while the latter is referred to as a coarse grid preconditioner by analogy to techniques first used in domain decomposition methods. Many variants exist in the two families that reduce to the same preconditioners if the exact eigenlements are used. In this paper we investigate the behavior of some of these techniques when the eigenlements are known only approximately. We use the first-order perturbation theory for eigenvalues and eigenvectors to investigate the behavior of the spectrum of the preconditioned systems using first-order approximation. We illustrate the sharpness of the first-order approximation and show the effect of the inexactness of the eigenlements on the behavior of the resulting preconditioner when applied to accelerating the conjugate gradient method.

Key words. spectral preconditioners, coarse grid preconditioner, deflating preconditioner, sensitivity analysis

AMS subject classifications. 15A06, 65F10, 65F15

DOI. 10.1137/040617546

1. Introduction. In many problems the convergence of Krylov solvers can be significantly slowed by the presence of small eigenvalues in the spectrum of the matrices involved in the solution of the linear systems. This occurs, for instance, when the conjugate gradient (CG) method is implemented to solve linear systems arising from the discretization of second-order elliptic problems. For symmetric positive definite (SPD) linear systems it is well known that the convergence of the CG method to solve $Ax = b$ depends to a large extent on the eigenvalue distribution of the coefficient matrix A . This can be illustrated by the bound on the rate of convergence of the CG method given by [10], viz.

$$(1.1) \quad \|x_k - x^*\|_A \leq 2\|x_0 - x^*\|_A \left(\frac{\sqrt{\kappa} - 1}{\sqrt{\kappa} + 1} \right)^k,$$

where $\kappa = \lambda_{\max}(A)/\lambda_{\min}(A)$ is the spectral condition number of A , the A -norm of x is $\|x\|_A = \sqrt{x^T A x}$, and the exact solution is $x^* = A^{-1}b$. This analysis leads to the idea of improving the convergence of the CG method by using a preconditioner M such that the ratio $\lambda_{\max}(MA)/\lambda_{\min}(MA)$ is less than κ . In this paper, we are interested in spectral preconditioners (and some of their variants) that exploit some information on the eigenpairs of A . The underlying idea driving these approaches is to capture in a low-dimensional space the modes that do not quickly converge

*Received by the editors October 24, 2004; accepted for publication (in revised form) by E. Ng December 3, 2005; published electronically March 17, 2006.

<http://www.siam.org/journals/simax/27-4/61754.html>

[†]CERFACS, 42 av. Gaspard Coriolis, 31057 Toulouse cedex 1, France (giraud@cerfacs.fr, gratton@cerfacs.fr).

[‡]Current address: ENSEEIHT-IRIT, 2 rue C. Camichel, 31 071 Toulouse, France.

with a first-level preconditioner. In order to be efficient and keep the dimension of the low-dimensional space reasonably small, these techniques are generally used in combination with a first-level preconditioner that does a good job of clustering most eigenvalues near to one with relatively few outliers near the origin [3, 4, 6, 13, 18]. These spectral preconditioners can be split into two main families, depending on their effect on the spectrum. They are referred to as deflation preconditioners [4, 8, 9, 14] if they attempt to move a subset of eigenvalues to a positive quantity σ ; they are referred to as coarse grid preconditioners [3, 9] if they attempt only to shift the subset close to σ . The latter name comes from domain decomposition and was first introduced in [2]. For this reason $\sigma = 1$ is often considered in practice. An impressive example of the efficiency of a spectral preconditioner is provided by the atmosphere data assimilation area [8]. In this application, nonlinear least-squares problems with more than 10^7 unknowns are solved daily using a Gauss–Newton approach. The sequence of linear least-squares involved in the nonlinear solution scheme is solved by the CG method. The correspondence between the CG and Lanczos methods is exploited on each linear problem to extract approximate spectral information. This spectral information is used to design a deflation spectral preconditioner for the subsequent linear least-squares problem.

When the exact eigenvectors are used, many of these spectral preconditioners reduce to the same expressions. The aim of this paper is to use the first-order perturbation theory for eigenvalues and eigenvectors to investigate the behavior of some of these preconditioners when they are constructed using approximate eigenelements. The paper is organized as follows. After the introduction of some notation, section 3 is devoted to the sensitivity analysis. We first establish in section 3.2 the theoretical results and illustrate in section 3.4 the sharpness of the expressions on a set of test matrices. Section 3.3 is devoted to the derivation of the condition number associated with the eigenvalues of the preconditioned matrix. We also indicate how this information can help screen the eigenelement information to obtain a nice clustering around σ for the spectrum of the preconditioned matrix. In section 4, we illustrate the behavior of the preconditioned CG (PCG) for the solution of linear systems associated with matrices from the Harwell–Boeing collection [7] when the spectral information is approximated. Finally, we conclude with some remarks in section 5.

2. Spectral preconditioner variants. We first consider one representative of the deflation preconditioners and one of the coarse grid preconditioners. Let $V = [v_1, \dots, v_n] \in \mathbb{R}^{n \times n}$ be an eigenbasis of A and $\{\lambda_i\}_{i=1, \dots, n}$ be the set of corresponding eigenvalues sorted by increasing magnitude. In order to move $\{\lambda_i\}_{i=1, \dots, k}$ to σ , we define the following deflation preconditioner:

$$(2.1) \quad M^{def} = I + \sum_{i=1}^k \left(\frac{\sigma}{\lambda_i} - 1 \right) \frac{v_i v_i^T}{\|v_i\|^2}.$$

The columns of V also form an eigenbasis of $M^{def}A$. For $V_k = [v_1, \dots, v_k]$, this preconditioner is such that $M^{def}AV_k = \sigma V_k$ and $M^{def}Aw = Aw$ if $V_k^T w = 0$, which shows that M^{def} moves the eigenvalues $\{\lambda_i\}_{i=1, \dots, k}$ to σ and leaves the rest of the spectrum unchanged. This technique is expected to be especially efficient in the case where $\{\lambda_i\}_{i=k+1, \dots, n}$ are already in the neighborhood of σ , in which case $\lambda_{\max}(MA)/\lambda_{\min}(MA)$ becomes close to one. Because (1.1) provides only an upper-bound for the convergence rate, it might be argued that, if $\{\lambda_i\}_{i=k+1, \dots, n}$ are already close to σ , shifting $\{\lambda_i\}_{i=1, \dots, k}$ to any quantity close to σ (and not necessarily to

σ exactly) does still improve the convergence of CG. To this end, if $\{\lambda_i\}_{i=1,\dots,k}$ are small, we can use the coarse grid preconditioner

$$(2.2) \quad M^{coarse} = I + \sigma V_k (\text{diag}(\lambda_i))^{-1} V_k^T,$$

where $\text{diag}(\lambda_i)$ denotes the diagonal matrix with entries λ_i . The columns of V also form an eigenbasis of $M^{coarse}A$. This preconditioner is such that $M^{coarse}Av_i = (\sigma + \lambda_i)v_i$ and $M^{coarse}Aw = Aw$ if $V_k^T w = 0$. That is, the eigenvalues $\{\lambda_i\}_{i=1,\dots,k}$ are shifted to $\sigma + \lambda_i$, while the rest of the spectrum is unchanged. This latter technique is particularly suited when $\{\lambda_i\}_{i=1,\dots,k}$ are small.

In the previous paragraph the spectral transformations associated with M^{def} and M^{coarse} rely on the facts that (λ_i, v_i) are exact eigenpairs of an SPD matrix. This implies two intensively used properties, $Av_i = \lambda_i v_i$ and $v_i^T v_j = 0$ if $i \neq j$. We assume now that we have access only to approximate spectral information, and denote by $(\tilde{\lambda}_i, \tilde{v}_i)$ the inexact eigenpairs such that the two latter properties do not necessarily hold. We suppose only that $\tilde{V}_k = [\tilde{v}_1, \dots, \tilde{v}_k]$ is such that $\tilde{V}_k^T \tilde{V}_k$ and $\tilde{V}_k^T A \tilde{V}_k$ are nonsingular. We can write the two above preconditioners in a form that does a weaker usage of the properties of the exact eigenpairs. The inexact “deflation” preconditioner then reads

$$(2.3) \quad M_1 = I + \tilde{V}_k \left(\sigma \left(\tilde{V}_k^T A \tilde{V}_k \right)^{-1} - \left(\tilde{V}_k^T \tilde{V}_k \right)^{-1} \right) \tilde{V}_k^T.$$

Noticing that $V_k^T AV_k$ is diagonal, we can be tempted to approximate it by the diagonal of the Rayleigh quotients. This gives

$$(2.4) \quad M_1^{rayl} = I + \tilde{V}_k \left(\sigma \left(\text{diag}(\tilde{v}_i^T A \tilde{v}_i) \right)^{-1} - \left(\tilde{V}_k^T \tilde{V}_k \right)^{-1} \right) \tilde{V}_k^T.$$

Furthermore, because the diagonal elements of M_1 are simply the eigenvalues, we can also use the following preconditioner formulation:

$$(2.5) \quad M_1^{eig} = I + \tilde{V}_k \left(\sigma \left(\text{diag}(\tilde{\lambda}_i) \tilde{v}_i^T \tilde{v}_i \right)^{-1} - \left(\tilde{V}_k^T \tilde{V}_k \right)^{-1} \right) \tilde{V}_k^T.$$

Using the orthogonality property of the eigenvectors, M_1 reduces to

$$(2.6) \quad M_{1,\perp}^{rayl} = I + \sum_{i=1}^k \left(\sigma \left(\frac{\tilde{v}_i^T A \tilde{v}_i}{\tilde{v}_i^T \tilde{v}_i} \right)^{-1} - 1 \right) \frac{\tilde{v}_i \tilde{v}_i^T}{\tilde{v}_i^T \tilde{v}_i},$$

which could also lead us to consider

$$(2.7) \quad M_{1,\perp}^{eig} = I + \sum_{i=1}^k \left(\frac{\sigma}{\tilde{\lambda}_i} - 1 \right) \frac{\tilde{v}_i \tilde{v}_i^T}{\tilde{v}_i^T \tilde{v}_i}.$$

Similarly, for the coarse grid preconditioners we consider the following variants:

$$(2.8) \quad M_{+1} = I + \sigma \tilde{V}_k \left(\tilde{V}_k^T A \tilde{V}_k \right)^{-1} \tilde{V}_k^T,$$

$$(2.9) \quad M_{+1}^{rayl} = I + \sigma \tilde{V}_k \left(\text{diag}(\tilde{v}_i^T A \tilde{v}_i) \right)^{-1} \tilde{V}_k^T,$$

$$(2.10) \quad M_{+1}^{eig} = I + \sigma \tilde{V}_k \left(\text{diag}(\tilde{\lambda}_i \tilde{v}_i^T \tilde{v}_i) \right)^{-1} \tilde{V}_k^T.$$

To study the performance of the above preconditioners in the presence of inexact spectral information, we assume that the spectral information is not related to A but to a nearby matrix $A + tE$, where t is a real parameter and $\|E\| = 1$. No assumption is made on the matrix E , and our analysis applies in particular to the cases where E is a symmetric and/or a low-rank matrix. Let $\lambda_i(t)$ and $v_i(t)$ denote the eigenvalues and eigenvectors, respectively, of $A + tE$. If A has only simple eigenvalues, it can be shown [16] that the eigenvalues of $A + tE$ are differentiable functions of t in a neighborhood $\mathcal{V}$ of $t = 0$. If the eigenvectors are normalized using $v_i(t)^T v_i = 1$, the eigenvectors are also differentiable functions of t in a neighborhood of $t = 0$. Note that none of the preconditioners assume that the eigenvectors have unit length. Indeed, the preconditioners are invariant by any nonzero scaling of the eigenvectors. Therefore the normalization $v_i(t)^T v_i = 1$ can be assumed for the analysis without loss of generality.

DEFINITION 1. We denote by $M_1(t)$, $M_1^{rayl}(t)$, $M_1^{eig}(t)$, $M_{1,\perp}^{rayl}(t)$, $M_{1,\perp}^{eig}(t)$, $M_{+1}(t)$, $M_{+1}^{rayl}(t)$, and $M_{+1}^{eig}(t)$ the preconditioners obtained by setting $\tilde{v}_i = v_i(t)$ and $\tilde{\lambda}_i = \lambda_i(t)$ in the definitions (2.3)–(2.10).

In this paper, we carry out a first-order analysis which shows the asymptotic sensitivity of the eigenvalues of the preconditioned matrix for small enough values of the parameter t . The approach is related to the backward error framework since the inaccuracies in v_i and λ_i are modeled by a perturbation of the coefficient matrix A .

3. Sensitivity analysis.

3.1. Notation. We introduce some notation and basic results used throughout this paper. For any square matrix $X \in \mathbb{R}^{n \times n}$, let $\underline{X}_i$ denote the $n \times (n - 1)$ matrix whose columns are those of X except for the i th, that is, $\underline{X}_i = [X(:, j)]_{j=1, \dots, n; j \neq i}$. For an $n \times n$ matrix X , $\{\lambda_1(X), \dots, \lambda_n(X)\}$ are the eigenvalues of X , where multiple eigenvalues are repeated. We also assume that $|\lambda_1(X)| \leq \dots \leq |\lambda_n(X)|$. The i th eigenvalue of A is denoted by λ_i when there is no possible confusion. Let $A \in \mathbb{R}^{n \times n}$ be an SPD matrix, where

$$AV = VD \quad \text{with} \quad V^T V = I \quad \text{and} \quad D = \text{diag}(\lambda_i)_{i=1, \dots, n}$$

denotes its spectral decomposition. We assume that all the eigenvalues of A are simple.

For a vector x , $\|x\| = (\sum_{i=1}^n x_i^2)^{1/2}$ is the Euclidean vector norm, and $\|A\| = \max_{\|x\|=1} \|Ax\|$ is the spectral norm of the matrix A . The operator $\circ$ denotes the Hadamard product: $A \circ B = [a_{ij}b_{ij}] \in \mathbb{C}^{m \times n}$, for A and $B \in \mathbb{C}^{m \times n}$. The spectral norm is submultiplicative with respect to the Hadamard product (see [1, p. 332]): $\|A \circ B\| \leq \|A\| \|B\|$.

Let $\tilde{V}$ and $\tilde{D}$ be defined by $\tilde{V} = [v_1(t), \dots, v_n(t)]$ and $\tilde{D} = \text{diag}(\tilde{\lambda}_i) = \text{diag}(\lambda_i(t))$. For sufficiently small $t \in \mathcal{V}$, we have

$$(A + tE)\tilde{V} = \tilde{V}\tilde{D}.$$

Note that $\tilde{V}(0) = V$. Using the Landau little “o” notation, the first order expansion of the eigenvalues and eigenvectors in the direction E (see [16]) is $\tilde{V} = V + \delta V(t) + o(t)$, where the i th column of $\delta V(t)$ is defined by

$$\delta v_i(t) = t \underline{V}_i (\lambda_i I - B_i)^{-1} \underline{V}_i^T E v_i,$$

and $\lambda_i(t) = \lambda_i + \delta\lambda_i(t) + o(t)$, where $\delta\lambda_i(t) = tv_i^T E v_i$ and the $(n-1) \times (n-1)$ diagonal matrix $B_i = \text{diag}(\lambda_j)_{j=1, \dots, n; j \neq i}$. The first-order expansion of the eigenvalues of the preconditioned matrices will be expressed in terms of the following $k \times k$ matrices W and Y defined in function of the eigenvalues of A by their (ℓ, s) -entries:

$$(3.1) \quad W_{\ell, \ell} = 0; \quad W_{\ell, s} = \frac{\sigma}{\lambda_s - \lambda_\ell} \sqrt{\frac{\lambda_\ell}{\lambda_s}} \quad \text{for } \ell \neq s,$$

$$(3.2) \quad Y_{\ell, \ell} = 0; \quad Y_{\ell, s} = \frac{\sigma - \lambda_s}{\lambda_s - \lambda_\ell} \sqrt{\frac{\lambda_\ell}{\lambda_s}} \quad \text{for } \ell \neq s.$$

Similarly we also introduce the $k \times k$ matrix $\Delta = V_k^T E V_k$ and the diagonal matrix $J = \text{diag}(-\frac{\sigma}{\lambda_\ell})_{\ell=1, \dots, k}$.

3.2. First-order expansion of the preconditioned matrix eigenvalues.

THEOREM 1. *The deflation preconditioner $M_1(t)$ is such that the eigenvalues of the preconditioned matrix $M_1(t)A$ are*

$$\begin{cases} \mu_i(t) = \sigma + o(t) & \text{if } i \leq k, \\ \mu_i(t) = \lambda_i(A) + o(t) & \text{if } i > k. \end{cases}$$

Proof of Theorem 1. Let us first mention that the eigenvalues of $M_1(t)A$ are also those of $AM_1(t)$ and then write

$$AM_1(t) = A + \sigma A \tilde{V}_k (\tilde{V}_k^T A \tilde{V}_k)^{-1} \tilde{V}_k - A \tilde{V}_k (\tilde{V}_k^T \tilde{V}_k)^{-1} \tilde{V}_k^T.$$

The entry (ℓ, s) of $(\tilde{V}^T \tilde{V})$ is defined by

$$\begin{aligned} (\tilde{V}_k^T \tilde{V}_k)_{\ell, s} &= (v_\ell^T + \delta v_\ell^T)(v_s + \delta v_s) \\ &= v_\ell^T v_s + v_\ell^T \delta v_s + \delta v_\ell^T v_s + o(t). \end{aligned}$$

Using (A.1), (A.2) and the facts that $v_\ell^T v_\ell = 1$, $\underline{V}_\ell^T v_\ell = 0$, and $v_\ell^T \underline{V}_\ell = 0$, we obtain

$$(3.3) \quad (\tilde{V}_k^T \tilde{V}_k) = I + C + o(t)$$

with $c_{\ell, \ell} = 0$ and

$$(3.4) \quad c_{\ell, s} = (\lambda_\ell - \lambda_s)^{-1} (v_\ell^T E^T v_s - v_\ell^T E v_s) t \quad \text{for } \ell \neq s.$$

Consequently

$$\begin{aligned} (\tilde{V}_k^T \tilde{V}_k)^{-1} &= I - (V_k^T V_k)^{-1} C (V_k^T V_k)^{-1} + o(t) \\ &= I - C + o(t) \end{aligned}$$

and

$$\begin{aligned} A \tilde{V}_k (\tilde{V}_k^T \tilde{V}_k)^{-1} \tilde{V}_k^T &= A V_k V_k^T + A \delta V_k V_k^T - A V_k C V_k^T \\ &\quad + A V_k \delta V_k^T + o(t) \\ (3.5) \quad &= A V_k V_k^T + A (\delta V_k V_k^T \\ &\quad + (\delta V_k V_k^T)^T - V_k C V_k^T) + o(t). \end{aligned}$$

Furthermore,

$$\begin{aligned}
 \tilde{V}_k^T A \tilde{V}_k &= V_k^T A V_k + (\delta V_k^T A V_k) + (\delta V_k^T A V_k)^T + o(t) \\
 (3.6) \quad &= V_k^T A V_k + (\delta V_k^T V_k D_k) + (\delta V_k^T V_k D_k)^T + o(t) \\
 &= D_k + G + o(t),
 \end{aligned}$$

where

$$g_{\ell,s} = \delta v_\ell^T v_s \lambda_s + \lambda_\ell v_\ell^T \delta v_s$$

and $D_k = \text{diag}(\lambda_i)_{i=1,\dots,k}$. Using (A.1), (A.2), and (A.3), we obtain $g_{\ell,\ell} = 0$ and

$$(3.7) \quad g_{\ell,s} = (\lambda_\ell - \lambda_s)^{-1} (\lambda_s v_\ell^T E^T v_s - \lambda_\ell v_\ell^T E v_s) t \quad \text{for } \ell \neq s.$$

Consequently we have

$$\begin{aligned}
 (3.8) \quad (\tilde{V}_k^T A \tilde{V}_k)^{-1} &= (V_k^T A V_k)^{-1} - (V_k^T A V_k)^{-1} G (V_k^T A V_k)^{-1} + o(t) \\
 &= D_k^{-1} - D_k^{-1} G D_k^{-1} + o(t)
 \end{aligned}$$

and

$$\begin{aligned}
 (3.9) \quad \tilde{V} \left(\tilde{V}_k^T A \tilde{V}_k \right)^{-1} \tilde{V}^T &= V_k D_k^{-1} V_k^T + (\delta V_k D_k^{-1} V_k^T) + (\delta V_k D_k^{-1} V_k^T)^T \\
 &\quad - V_k D_k^{-1} G D_k^{-1} V_k^T + o(t).
 \end{aligned}$$

Finally, combining (3.5) and (3.9), we obtain

$$\begin{aligned}
 A M_1(t) &= A + A \tilde{V}_k ((\tilde{V}_k A \tilde{V})^{-1} - (\tilde{V}^T \tilde{V}_k)^{-1}) \tilde{V}_k \\
 &= A M_1(0) + \delta M_1 + o(t)
 \end{aligned}$$

with

$$\begin{aligned}
 \delta M_1 &= A (\delta V_k (I + \sigma D_k^{-1}) V_k^T + (\delta V_k (I + \sigma D_k^{-1}) V_k^T)^T \\
 &\quad - V_k (C + \sigma D_k^{-1} G D_k^{-1}) V_k^T).
 \end{aligned}$$

We recall that

$$A M_1(0) = V \underbrace{\text{diag}(\sigma, \dots, \sigma)}_k, \lambda_{k+1}, \dots, \lambda_n V^T.$$

- For $j > k$, the first-order approximation of the simple eigenvalues reads

$$\mu_j = \lambda_j + v_j^T \delta M_1 v_j + o(t) = \lambda_j + o(t),$$

since $\delta M_1 v_j = 0$ because of (A.3) and the orthogonality of V .

- For $j \leq k$, the first-order approximation of the semisimple multiple eigenvalue σ reads (see [12, p. 402] or [17])

$$\mu_j = \sigma + \lambda_j (V_k^T \delta M_1 V_k) + o(t),$$

where $(V_k^T \delta M_1 V_k)$ is a $k \times k$ matrix with diagonal entries equal to zero and off-diagonal entries

$$\begin{aligned}
 (V_k^T \delta M_1 V_k)_{\ell,s} &= v_\ell^T A (\delta V_k (I + \sigma D_k^{-1}) V_k^T + (\delta V_k (I + \sigma D_k^{-1}) V_k^T)^T \\
 &\quad - V_k (C + \sigma D_k^{-1} G D_k^{-1}) V_k^T) v_s \\
 &= \lambda_\ell (v_\ell^T \delta V_k (I + \sigma D_k^{-1}) V_k^T v_s + v_\ell^T V_k (I + \sigma D_k^{-1}) \delta V_k^T v_s \\
 &\quad - v_\ell^T V_k (C + \sigma D_k^{-1} G D_k^{-1}) V_k^T v_s) \\
 &= \lambda_\ell (v_\ell^T \delta v_s (\sigma \lambda_s^{-1} + 1) + (\sigma \lambda_\ell^{-1} + 1) \delta v_\ell^T v_s \\
 &\quad - (c_{\ell,s} + \sigma (\lambda_s \lambda_\ell)^{-1} g_{\ell,s})) \\
 &= 0. \quad \square
 \end{aligned}$$

THEOREM 2. *The deflation preconditioner $M_1^{rayl}(t)$ is such that the eigenvalues of the preconditioned matrix $M_1^{rayl}(t)A$ are*

$$\begin{cases} \mu_i(t) = \sigma + \lambda_i(W \circ \Delta + W^T \circ \Delta^T)t + o(t) & \text{if } i \leq k, \\ \mu_i(t) = \lambda_i(A) + o(t) & \text{if } i > k. \end{cases}$$

Proof of Theorem 2. We have

$$AM_1^{rayl}(t) = A + \sigma A \tilde{V}_k (\text{diag}(\tilde{v}_i^T A \tilde{v}_i))^{-1} \tilde{V}_k - A \tilde{V}_k (\tilde{V}_k^T \tilde{V}_k)^{-1} \tilde{V}_k^T.$$

From (3.6) it follows that $\text{diag}(\tilde{v}_i^T A \tilde{v}_i) = D_k + o(t)$, which yields

$$(3.10) \quad \tilde{V}_k (\text{diag}(\tilde{v}_i^T A \tilde{v}_i))^{-1} \tilde{V}_k = V_k D_k^{-1} V_k^T + \delta V_k D_k^{-1} V_k + V_k D_k^{-1} \delta V_k + o(t).$$

Using (3.5), we find that $AM_1^{rayl}(t) = AM_1^{rayl}(0) + \delta M_2 + o(t)$, where

$$\begin{aligned}
 \delta M_2 &= A (\sigma \delta V_k D_k^{-1} V_k^T + \sigma V_k D_k^{-1} \delta V_k^T - \delta V_k V_k^T - (\delta V_k V_k^T)^T + V_k C V_k^T) \\
 &= A (\delta V_k (\sigma D_k^{-1} - I) V_k^T + V_k (\sigma D_k^{-1} - I) \delta V_k^T + V_k C V_k^T).
 \end{aligned}$$

- For $j > k$, the first-order approximation of the simple eigenvalues reads

$$\mu_j = \lambda_j + v_j^T \delta M_2 v_j + o(t) = \lambda_j + o(t),$$

since $v_j \delta M_2 v_j = 0$ because of (A.3) and the orthogonality of V .

- For $j \leq k$, the first-order approximation of the semisimple multiple eigenvalue σ reads as in the proof of Theorem 1,

$$\mu_j = \sigma + \lambda_j (V_k^T \delta M_2 V_k) + o(t),$$

where $(V_k^T \delta M_2 V_k)$ is a $k \times k$ matrix. The diagonal element of $(V_k^T \delta M_2 V_k)$ is

$$\begin{aligned}
 (V_k^T \delta M_2 V_k)_{\ell,\ell} &= v_\ell^T A (\delta V_k (\sigma D_k^{-1} - I) V_k^T + V_k (\sigma D_k^{-1} - I) \delta V_k^T + V_k C V_k^T) v_\ell \\
 &= \lambda_\ell ((\sigma \lambda_\ell^{-1} - 1) v_\ell^T \delta v_\ell + (\sigma \lambda_\ell^{-1} - 1) \delta v_\ell^T v_\ell + c_{\ell\ell}) \\
 (3.11) \quad &= 0,
 \end{aligned}$$

where the final equality uses (A.3) and $c_{\ell\ell} = 0$. For the (ℓ, s) off-diagonal element,

$$\begin{aligned}
 (V_k^T \delta M_2 V_k)_{\ell,s} &= v_\ell^T A (\delta V_k (\sigma D_k^{-1} - I) V_k^T + V_k (\sigma D_k^{-1} - I) \delta V_k^T + V_k C V_k^T) v_s \\
 &= \lambda_\ell ((\sigma \lambda_s^{-1} - 1) v_\ell^T \delta v_s + (\sigma \lambda_\ell^{-1} - 1) \delta v_\ell^T v_s + c_{\ell,s}) \\
 &= t v_\ell^T E v_s \frac{\lambda_\ell}{\lambda_s} \frac{\sigma}{\lambda_s - \lambda_\ell} - t v_\ell^T E^T v_s \frac{\sigma}{\lambda_s - \lambda_\ell}.
 \end{aligned}$$

We consider the diagonal scaling

$$F = D_k^{-1/2} V_k^T \delta M_2 V_k D_k^{1/2}.$$

The matrix F has the same eigenvalues as $V_k^T \delta M_2 V_k$. The end of the proof follows by noticing that the diagonal entries of $F_{\ell,\ell}$ are zeros. The (ℓ, s) off-diagonal entry verifies

$$\begin{aligned} F_{\ell,s} &= \frac{\sigma}{\lambda_s - \lambda_\ell} \left(\sqrt{\frac{\lambda_\ell}{\lambda_s}} v_\ell^T E v_s - \sqrt{\frac{\lambda_s}{\lambda_\ell}} v_\ell^T E^T v_s \right) \\ (3.12) \quad &= \frac{\sigma}{\lambda_s - \lambda_\ell} \sqrt{\frac{\lambda_\ell}{\lambda_s}} v_\ell^T E v_s + \frac{\sigma}{\lambda_\ell - \lambda_s} \sqrt{\frac{\lambda_s}{\lambda_\ell}} v_s^T E v_\ell \\ &= W_{\ell,s} \Delta_{\ell,s} + W_{s,\ell} \Delta_{s,\ell} = (W \circ \Delta + W^T \circ \Delta^T)_{\ell,s}. \quad \square \end{aligned}$$

THEOREM 3. *The deflation preconditioner $M_1^{eig}(t)$ is such that the eigenvalues of the preconditioned matrix $M_1^{eig}(t)A$ are*

$$\begin{cases} \mu_i(t) = \sigma + \lambda_i((W + J) \circ \Delta + W^T \circ \Delta^T)t + o(t) & \text{if } i \leq k, \\ \mu_i(t) = \lambda_i(A) + o(t) & \text{if } i > k. \end{cases}$$

Proof of Theorem 3. We have

$$AM_1^{eig}(t) = A + \sigma A \tilde{V}_k (\text{diag}(\tilde{\lambda}_i))^{-1} \tilde{V}_k^T - A \tilde{V}_k (\tilde{V}_k^T \tilde{V}_k)^{-1} \tilde{V}_k^T.$$

Since $\text{diag}(\tilde{\lambda}_i \tilde{v}_i^T \tilde{v}_i) = D_k + t \text{diag}(v_i^T E v_i) + o(t)$, we obtain

$$\begin{aligned} \tilde{V}_k (\text{diag}(\tilde{\lambda}_i \tilde{v}_i^T \tilde{v}_i))^{-1} \tilde{V}_k^T &= -t V_k^T D_k^{-2} \text{diag}(v_i^T E v_i) V_k^T \\ (3.13) \quad &+ \delta V_k D_k^{-1} V_k^T + V_k D_k^{-1} \delta V_k^T + o(t). \end{aligned}$$

Therefore, $AM_1^{eig}(t) = AM_1^{eig}(0) + \delta M_3 + o(t)$, where

$$\begin{aligned} \delta M_3 &= -t \sigma A V_k D_k^{-2} \text{diag}(v_i^T E v_i) V_k^T + A (\sigma \delta V_k D_k^{-1} V_k^T + \sigma V_k D_k^{-1} \delta V_k^T \\ &\quad - \delta V_k V_k^T - (\delta V_k V_k^T)^T + V_k C V_k^T) \\ &= -t \sigma V_k D_k^{-1} \text{diag}(v_i^T E v_i) V_k^T + \delta M_2. \end{aligned}$$

- For $j > k$, the first-order approximation of the simple eigenvalues reads

$$\mu_j = \lambda_j + v_j^T \delta M_3 v_j + o(t) = \lambda_j + o(t),$$

since $v_j \delta M_3 v_j = 0$ because of (A.3) and the orthogonality of V .

- For $j \leq k$, the first-order approximation of the semisimple multiple eigenvalue σ reads as in the proof of Theorem 1,

$$\mu_j = \sigma + \lambda_j (V_k^T \delta M_3 V_k) + o(t),$$

where $(V_k^T \delta M_3 V_k)$ is a $k \times k$ matrix. The diagonal element of $(V_k^T \delta M_3 V_k)$ is

$$(V_k^T \delta M_3 V_k)_{\ell,\ell} = (V_k^T \delta M_2 V_k)_{\ell,\ell} - \sigma \frac{v_\ell^T E v_\ell}{\lambda_\ell} t = 0 - \sigma \frac{v_\ell^T E v_\ell}{\lambda_\ell} t,$$

where the final equality uses (3.11). For the (ℓ, s) off-diagonal element,

$$\begin{aligned} (V_k^T \delta M_3 V_k)_{\ell, s} &= (V_k^T \delta M_2 V_k)_{\ell, s} + (t \sigma D_k^{-1} \text{diag}(v_i^T E v_i))_{\ell, s} \\ &= (V_k^T \delta M_2 V_k)_{\ell, s}, \end{aligned}$$

and the result follows from a diagonal scaling by considering the matrix

$$D_k^{-1/2} V_k^T \delta M_3 V_k D_k^{1/2},$$

which has the same eigenvalues as $V_k^T \delta M_3 V_k$.

Arguments similar to those used to derive (3.12) conclude the proof. $\square$

THEOREM 4. *The deflation preconditioner $M_{1,\perp}^{rayl}(t)$ is such that the eigenvalues of the preconditioned matrix $M_{1,\perp}^{rayl}(t)A$ are*

$$\begin{cases} \mu_i(t) = \sigma + \lambda_i(Y \circ \Delta + Y^T \circ \Delta^T)t + o(t) & \text{if } i \leq k, \\ \mu_i(t) = \lambda_i(A) + o(t) & \text{if } i > k. \end{cases}$$

Proof of Theorem 4. A first-order expansion shows that $AM_{1,\perp}^{rayl}(t) = AM_{1,\perp}^{rayl}(0) + \delta M_4 + o(t)$, where

$$\delta M_4 = \sum_{i=1}^k \left(\left(\frac{\sigma}{\lambda_i} - 1 \right) (A \delta v_i v_i^T + A v_i \delta v_i^T) - \frac{\sigma}{\lambda_i^2} (\delta v_i^T A v_i + v_i^T A \delta v_i) A v_i v_i^T \right).$$

- For $j > k$, the first-order approximation of the simple eigenvalues reads

$$\mu_j = \lambda_j + v_j^T \delta M_4 v_j + o(t) = \lambda_j + o(t),$$

since $v_j^T \delta M_4 v_j = 0$ because of (A.3) and the orthogonality of V .

- For $j \leq k$, the first-order approximation of the semisimple multiple eigenvalue σ reads as in the proof of Theorem 1,

$$\mu_j = \sigma + \lambda_j (V_k^T \delta M_4 V_k) + o(t),$$

where $(V_k^T \delta M_4 V_k)$ is a $k \times k$ matrix. The diagonal element of $(V_k^T \delta M_4 V_k)$ is

$$(V_k^T \delta M_4 V_k)_{\ell, \ell} = 0,$$

using the orthogonality of V . For the (ℓ, s) off-diagonal element, using (A.2) gives

$$\begin{aligned} (V_k^T \delta M_4 V_k)_{\ell, s} &= \left(\frac{\sigma}{\lambda_s} - 1 \right) v_\ell^T A \delta v_s + \left(\frac{\sigma}{\lambda_\ell} - 1 \right) v_\ell^T A v_\ell \delta v_\ell^T v_s \\ &= (\lambda_s - \lambda_\ell)^{-1} \left(\left(\frac{\sigma \lambda_\ell}{\lambda_s} - \lambda_\ell \right) v_\ell^T E v_s - (\sigma - \lambda_\ell) v_\ell^T E^T v_s \right) t \\ &= (\lambda_s - \lambda_\ell)^{-1} \left(\frac{\lambda_\ell}{\lambda_s} (\sigma - \lambda_s) v_\ell^T E v_s - (\sigma - \lambda_\ell) v_\ell^T E^T v_s \right) t, \end{aligned}$$

and the result follows from a diagonal scaling by considering the matrix

$$D_k^{-1/2} V_k^T \delta M_4 V_k D_k^{1/2},$$

which has the same eigenvalues as $V_k^T \delta M_4 V_k$.

Arguments similar to those used to derive (3.12) conclude the proof. $\square$

THEOREM 5. *The deflation preconditioner $M_{1,\perp}^{eig}(t)$ is such that the eigenvalues of the preconditioned matrix $M_{1,\perp}^{eig}(t)A$ are*

$$\begin{cases} \mu_i(t) = \sigma + \lambda_i((Y + J) \circ \Delta + Y^T \circ \Delta^T)t + o(t) & \text{if } i \leq k, \\ \mu_i(t) = \lambda_i(A) + o(t) & \text{if } i > k. \end{cases}$$

Proof of Theorem 5. A first-order expansion shows that $AM_{1,\perp}^{eig}(t) = AM_{1,\perp}^{eig}(0) + \delta M_5 + o(t)$, where

$$\delta M_5 = \sum_{i=1}^k \left(\left(\frac{\sigma}{\lambda_i} - 1 \right) (A \delta v_i v_i^T + A v_i \delta v_i^T) - \sigma \frac{v_i^T E v_i}{\lambda_i^2} A v_i v_i^T \right).$$

- For $j > k$, the first-order approximation of the simple eigenvalues reads

$$\mu_j = \lambda_j + v_j^T \delta M_5 v_j + o(t) = \lambda_j + o(t),$$

since $v_j^T \delta M_5 v_j = 0$ because of (A.3) and the orthogonality of V .

- For $j \leq k$, the first-order approximation of the semisimple multiple eigenvalue σ reads as in the proof of Theorem 1,

$$\mu_j = \sigma + \lambda_j(V_k^T \delta M_5 V_k) + o(t),$$

where $(V_k^T \delta M_5 V_k)$ is a $k \times k$ matrix. The diagonal element of $(V_k^T \delta M_5 V_k)$ is

$$(V_k^T \delta M_5 V_k)_{\ell,\ell} = -\sigma \frac{v_\ell^T E v_\ell}{\lambda_\ell} t,$$

using the orthogonality of V . For the (ℓ, s) off-diagonal element, using (A.2) gives

$$(V_k^T \delta M_5 V_k)_{\ell,s} = \left(\frac{\sigma}{\lambda_s} - 1 \right) v_\ell^T A \delta v_s + \left(\frac{\sigma}{\lambda_\ell} - 1 \right) v_\ell^T A v_\ell \delta v_\ell^T v_s = (V_k^T \delta M_4 V_k)_{\ell,s},$$

and the result follows from a diagonal scaling by considering the matrix

$$D_k^{-1/2} V_k^T \delta M_5 V_k D_k^{1/2},$$

which has the same eigenvalues as $V_k^T \delta M_5 V_k$.

Arguments similar to those used to derive (3.12) conclude the proof. $\square$

THEOREM 6. *The coarse-grid preconditioner $M_{+1}(t)$ (resp., $M_{+1}^{rayl}(t)$) is such that the eigenvalues of the preconditioned matrix $M_{+1}(t)A$ (resp., $M_{+1}^{rayl}(t)A$) are*

$$\begin{cases} \mu_i(t) = \sigma + \lambda_i(A) + o(t) & \text{if } i \leq k, \\ \mu_i(t) = \lambda_i(A) + o(t) & \text{if } i > k. \end{cases}$$

Proof of Theorem 6. Let

$$AM_{+1}(t) = A + \sigma A \tilde{V}_k (\tilde{V}_k^T A \tilde{V}_k)^{-1} \tilde{V}_k;$$

from (3.9) we have

$$\begin{aligned} AM_{+1}(t) &= A + \sigma AV_k (V_k^T AV_k)^{-1} V_k^T \\ &\quad + \sigma A \left(\delta V_k D_k^{-1} V_k^T + (\delta V_k D_k^{-1} V_k^T)^T - V_k D_k^{-1} G D_k^{-1} V_k^T \right) + o(t) \\ &= AM_{+1}(0) + \delta M_6 + o(t) \end{aligned}$$

with

$$(3.14) \quad \delta M_6 = \sigma A \left(\delta V_k D_k^{-1} V_k^T + (\delta V_k D_k^{-1} V_k^T)^T - V_k D_k^{-1} G D_k^{-1} V_k^T \right).$$

For any j the first-order approximation of the simple eigenvalues reads

$$\mu_j = \sigma + \lambda_j + v_j^T \delta M_6 v_j + o(t),$$

with $\delta M_6 v_j = 0$ because of (A.3) and the orthogonality of V . A comparison of (3.10) and (3.9) shows that the proof for $M_{+1}^{rayl}(t)$ is obtained by setting $G = 0$ in the above proof. $\square$

THEOREM 7. *The coarse grid preconditioner $M_{+1}^{eig}(t)$ is such that the eigenvalues of the preconditioned matrix $M_{+1}^{eig}(t)A$ are*

$$\begin{cases} \mu_i(t) = \sigma + \lambda_i(A) - \sigma \frac{v_i^T E v_i}{\lambda_i(A)} t + o(t) & \text{if } i \leq k, \\ \mu_i(t) = \lambda_i(A) + o(t) & \text{if } i > k. \end{cases}$$

Proof of Theorem 7. Let

$$AM_{+1}^{eig}(t) = A + \sigma A \tilde{V} \text{diag}(\tilde{\lambda}_i \tilde{v}_i^T \tilde{v}_i)^{-1} \tilde{V};$$

from (3.13) we have

$$\begin{aligned} AM_{+1}^{eig}(t) &= AM_{+1}^{eig}(0) - \sigma V_k^T D_k^{-1} \text{diag}(v_i^T \delta A v_i) V_k^T + \sigma A \delta V_k D_k^{-1} V_k^T \\ &\quad + \sigma A V_k D_k^{-1} \delta V_k^T + o(t) \\ &= AM_{+1}^{eig}(0) + \delta M_7 + o(t), \end{aligned}$$

with

$$\delta M_7 = \sigma V_k^T D_k^{-1} \text{diag}(v_i^T \delta A v_i) V_k^T + \sigma A \delta V_k D_k^{-1} V_k^T + \sigma A V_k D_k^{-1} \delta V_k^T.$$

For any j the first-order approximation of the simple eigenvalues reads

$$\mu_j = \sigma + \lambda_j + v_j^T \delta M_7 v_j + o(t),$$

with $v_j^T \delta M_7 v_j = -\sigma \frac{v_j^T E v_j}{\lambda_j} t$ because of (A.3) and the orthogonality of V . $\square$

Remark 1. All the theoretical study has been made assuming that all the eigenvalues of A are simple. Actually, the results are still true if some of the λ_i for $i > k$ are multiple (i.e., the ones that are not targeted by the preconditioners).

TABLE 3.1

Matrices X_1 and X_2 for the spectral preconditioners and some cases of ill-conditioning. The terms “cluster” and “small” refer respectively to the presence of cluster or of small isolated eigenvalues.

Prec.	X_1	X_2	Some cases of ill-conditioning
M_1	0	0	None
M_1^{rayl}	W	0	Cluster, small
M_1^{eig}	W	J	Cluster, small
$M_{1,\perp}^{rayl}$	Y	0	Cluster, small
$M_{1,\perp}^{eig}$	Y	J	Cluster, small
M_{+1}	0	0	None
M_{+1}^{rayl}	0	0	None
M_{+1}^{eig}	0	J	Small

3.3. Sensitivity and backward errors. For all the preconditioners considered in this paper, the eigenvalues of the preconditioned matrices write $\mu_i(t) = \mu_i(0) + \lambda_i((X_1 + X_2) \circ \Delta + X_1^T \circ \Delta^T)t + o(t)$, where the X_i are matrices depending on selected preconditioner and the targeted eigenvalues. We summarize the various values of the matrices X_1 and X_2 for the different preconditioners in Table 3.1. We can therefore define a condition number κ_i for the eigenvalue μ_i in the direction of E (see [15]) by

$$(3.15) \quad \kappa_i = \lim_{u \rightarrow 0} \sup_{0 < |t| < u} \frac{|\mu_i(t) - \mu_i(0)|}{|t|} = |\lambda_i((X_1 + X_2) \circ \Delta + X_1^T \circ \Delta^T)|.$$

Notice that κ_i is not the usual condition number of the eigenvalue of a matrix [10, p. 323], but the condition number of the map $\mu_i(t)$. Taking norms and using the submultiplicativity of the spectral norm with respect to the Hadamard product yields

$$(3.16) \quad \begin{aligned} \kappa_i &\leq \|(X_1 + X_2) \circ \Delta + X_1^T \circ \Delta^T\| \\ &\leq (2\|X_1\| + \|X_2\|)\|\Delta\| = (2\|X_1\| + \|X_2\|)\|V_k^T E V_k\| \\ &\leq (2\|X_1\| + \|X_2\|), \end{aligned}$$

where we have used that $\|E\| = 1$. That majorization $\|(X_1 + X_2) \circ \Delta + X_1^T \circ \Delta^T\| \leq (2\|X_1\| + \|X_2\|)\|\Delta\|$ is a worst case bound may be pessimistic in the case where Δ has zero entries corresponding to large entries in X_1 and X_2 . Equations (3.15) and (3.16) show that if the entries of X_1 and X_2 are small, the condition number of the eigenvalues μ_i is small. By inspecting the equalities (3.1) and (3.2) it follows that asymptotically for $t \rightarrow 0$,

- the preconditioners M_1 , M_{+1} , and M_{+1}^{rayl} are stable (i.e., X_1, X_2 are the zero matrix),
- the preconditioners M_1^{rayl} , $M_{1,\perp}^{rayl}$, M_1^{eig} , and $M_{1,\perp}^{eig}$ may present an instability if for some (s, ℓ) the ratio $\frac{\sigma}{\lambda_s - \lambda_\ell} \sqrt{\lambda_\ell / \lambda_s}$ is large. In the above statement we have assumed that λ_s is far from σ , which seems to be a reasonable assumption, as otherwise we would not have targeted this eigenvalue. This instability happens, for instance, if some eigenvalues are clustered or small and isolated.

In Table 3.1 we summarize the situation where a high sensitivity of the eigenvalues is expected. In the case where E is a real symmetric matrix, M_1^{eig} and $M_{1,\perp}^{eig}$ are the same preconditioners, and elementary calculations show that $X_1 \circ \Delta + X_1^T \circ \Delta^T$ are the same in both cases.

For the solution of linear systems, it is possible to combine backward error with sensitivity analysis to obtain an estimate of the forward error on the solution [11], under the assumption that the dependency of the solution on the matrix is not too nonlinear. Similar assumptions yield, in our framework, that if the eigenpairs used in the preconditioners are exact eigenpairs of $A + tE$, we can estimate the perturbation on the eigenvalues of the preconditioners induced by the use of inexact spectral information by the quantity $(2\|X_1\| + \|X_2\|)$. If the eigenelements $(\tilde{\lambda}_i, \tilde{v}_i)$ are given, they can readily be considered as exact eigenpairs of the matrix

$$(3.17) \quad A - (A\tilde{V}_k - \tilde{D}_k\tilde{V}_k)\tilde{V}_k^\dagger,$$

where $\dagger$ denotes the Moore–Penrose inverse, i.e., $tE = -(A\tilde{V}_k - \tilde{D}_k\tilde{V}_k)\tilde{V}_k^\dagger$. Therefore an estimate for the gap $|\mu_i - \tilde{\mu}_i|$ is $(2\|X_1\| + \|X_2\|)\|(A\tilde{V}_k - \tilde{D}_k\tilde{V}_k)\tilde{V}_k^\dagger\|$. We note that if $\tilde{V}_k$ has orthogonal columns, then $\tilde{V}_k^\dagger$ reduces to $\tilde{V}_k^T$, and an upper bound for the gap is $(2\|X_1\| + \|X_2\|)\|A\tilde{V}_k - \tilde{D}_k\tilde{V}_k\|$.

This analysis shows that the gap $|\mu_i - \tilde{\mu}_i|$ depends on two quantities:

- the quantities $(2\|X_1\| + \|X_2\|)$, which characterize the mathematical sensitivity of the eigenvalues of the preconditioned matrix,
- the residual error $\|(A\tilde{V}_k - \tilde{D}_k\tilde{V}_k)\tilde{V}_k^\dagger\|$, which from (3.17) is an upper bound for the backward error associated with the approximate eigenpairs.

For a targeted gap ρ_{targ} , the first-order analysis shows that the residual error has to be small enough so that its product with the sensitivity term does not exceed ρ_{targ} . A screening of spectral information could be based on this idea.

3.4. Numerical illustrations. To illustrate and assess the correctness of the first-order expansions given in the previous section we consider 10×10 matrices with prescribed eigenvalues. We take $\sigma = 1$, $A = QDQ^T$, where Q is the orthogonal matrix whose columns are the eigenvectors of the two-dimensional Laplacian. The matrix D is chosen to illustrate the high sensitivity of the preconditioners when the eigenvalues of A are small or clustered. We use the preconditioners to shift the three smallest eigenvalues of A . We consider the perturbed matrix $A + tE$, where E is a random matrix with unit spectral norm and t is the perturbation parameter. Using the routine `eig` of MATLAB, we compute the eigenpairs $(\lambda_i(t), v_i(t))$ of $A + tE$. The three eigenpairs corresponding to those selected on the original matrix are used to construct the approximate “deflation” and “coarse grid” preconditioners. An additional eigenvalue computation, still with `eig`, is performed on the preconditioned matrix to obtain the eigenvalues $\tilde{\mu}_i$. The estimates using the first-order expansions of Theorems 1–7 are denoted by $\bar{\mu}_i$.

In a first experiment we choose $D = \text{diag}(1, 2, \dots, 10)$ and decide to move the three smallest eigenvalues. In Table 3.2, we report on the distance $\tilde{\mu}_i - \bar{\mu}_i$, for $i = 1, \dots, 4$, the four smallest eigenvalues of the preconditioned system. We see that for a perturbation size of $t = 10^{-2}$ the first-order estimation accurately approximates the computed eigenvalues, both the multiple ones (first three) and the simple one (fourth): $|\tilde{\mu}_i - \bar{\mu}_i|$ has several order of magnitude less than t .

As mentioned in section 3.3, the first-order expansions reveal the sensitivity of the spectrum of the preconditioned matrices to the magnitude and clustering of the targeted eigenvalues.

We first investigate the case of a small isolated eigenvalue. For the matrix D , we consider $\lambda_1 = 10^{-4}$, $\lambda_2 = 10^{-2}$, and $\lambda_3 = 10^{-1}$; the seven other eigenvalues are in the neighborhood of 2. In Table 3.3 we display, for the perturbation size $t = 10^{-5}$, the

TABLE 3.2

Difference between the computed and first-order estimates of the eigenvalues for $t = 10^{-2}$.

Precond.	$ \tilde{\mu}_1 - \bar{\mu}_1 $	$ \tilde{\mu}_2 - \bar{\mu}_2 $	$ \tilde{\mu}_3 - \bar{\mu}_3 $	$ \tilde{\mu}_4 - \bar{\mu}_4 $
M_1	3.8e-07	7.1e-08	2.8e-08	3.8e-08
M_1^{rayl}	1.5e-06	1.5e-07	1.2e-06	3.8e-08
M_1^{eig}	2.3e-05	3.4e-07	3.5e-07	3.9e-08
$M_{1,\perp}^{rayl}$	1.4e-06	8.1e-08	9.6e-07	3.8e-08
$M_{1,\perp}^{eig}$	2.3e-05	6.7e-07	5.1e-07	3.9e-08
M_{+1}	5.7e-07	2.5e-07	3.9e-05	3.9e-05
M_{+1}^{rayl}	2.0e-06	3.0e-07	3.9e-05	4.0e-05
M_{+1}^{eig}	2.2e-05	2.2e-07	1.2e-05	1.3e-05

TABLE 3.3

Small isolated eigenvalue: computed and estimated cluster radius for $t = 10^{-5}$.

Precond.	$\tilde{\rho}$	$\bar{\rho}$	ρ
M_1	3.7e-07	0.0e+00	0.0e+00
M_1^{rayl}	2.0e-03	2.0e-03	0.0e+00
M_1^{eig}	1.8e-01	2.1e-01	0.0e+00
$M_{1,\perp}^{rayl}$	2.0e-03	2.0e-03	0.0e+00
$M_{1,\perp}^{eig}$	1.8e-01	2.1e-01	0.0e+00
M_{+1}	1.0e-01	1.0e-01	1.0e-01
M_{+1}^{rayl}	1.0e-01	1.0e-01	1.0e-01
M_{+1}^{eig}	1.7e-01	2.1e-01	1.0e-01

exact radius $\rho = \max_{i=1,2,3} |1 - \mu_i|$, the computed radius $\tilde{\rho} = \max_{i=1,2,3} |1 - \tilde{\mu}_i|$, and the estimated radius $\bar{\rho} = \max_{i=1,2,3} |1 - \bar{\mu}_i|$. These quantities measure the radius of the cluster obtained around 1. If the exact eigenpairs were available, we would have ρ equal to zero for the deflation preconditioners and equal to $10^{-1} = \max\{\lambda_1, \lambda_2, \lambda_3\}$ for the coarse grid preconditioners. The comparison of the first two results shows again the sharpness of the estimations given by the first-order approximations, which are fairly accurate for all the preconditioners. Inspecting the first and third column, we see that M_1 does a fairly good job in moving the eigenvalues close to 1. This nice clustering deteriorates when one uses the Rayleigh quotients for approximating the eigenvalues (i.e., M_1^{rayl} and $M_{1,\perp}^{rayl}$) and is even worse when we use the perturbed eigenvalues computed by **eig** (i.e., M_1^{eig} and $M_{1,\perp}^{eig}$). Regarding the coarse grid preconditioners, we see that both M_{+1} and M_{+1}^{rayl} behave similarly as if they were constructed with exact eigenpairs: they shift λ_i to $1 + \lambda_i$. The preconditioner M_{+1}^{eig} is slightly more sensitive to perturbations. In addition, we see that the coarse grid preconditioners give rise to a cluster radius similar to that from M_1^{eig} and $M_{1,\perp}^{eig}$. We recall that M_1^{eig} and $M_{1,\perp}^{eig}$ are supposed to better cluster around 1, because they are designed to translate to 1 exactly and not to $1 + \lambda_i$.

In order to study the sensitivity of the spectrum of the preconditioners when the targeted eigenvalues are clustered we define the matrix D by taking its three smallest diagonal entries in a cluster of radius 10^{-4} around 10^{-1} ; the other seven eigenvalues are in the neighborhood of 2. We consider a perturbation of size $t = 10^{-5}$ for the numerical experiments reported in Table 3.4. The first and third columns show that M_1 is efficient in clustering the targeted eigenvalues close to 1. The other variants

TABLE 3.4

Clustered eigenvalues: computed and estimated cluster radius for $t = 10^{-5}$.

Precond.	$\tilde{\rho}$	$\bar{\rho}$	ρ
M_1	4.2e-10	0.0e+00	0.0e+00
M_1^{rayl}	2.5e-02	2.7e-02	0.0e+00
M_1^{eig}	2.5e-02	2.7e-02	0.0e+00
$M_{1,\perp}^{rayl}$	2.3e-02	2.4e-02	0.0e+00
$M_{1,\perp}^{eig}$	2.3e-02	2.4e-02	0.0e+00
M_{+1}	1.0e-01	1.0e-01	1.0e-01
M_{+1}^{rayl}	1.2e-01	1.0e-01	1.0e-01
M_{+1}^{eig}	1.2e-01	1.0e-01	1.0e-01

of the deflation approaches behave rather poorly as $\tilde{\rho}$ is about 10^{-2} . Regarding the coarse grid preconditioners, M_{+1} behaves the best. The two others are still performing well, as $\tilde{\rho}$ does not grow much beyond its exact value ρ . Tables 3.2–3.4 correspond to three different matrices and illustrate the sharpness of the first-order expansions. Table 3.2 shows that the first-order approximation is valid for some matrices, even for large perturbations ($t = 10^{-2}$). These experimental results are in agreement with what could be expected from Table 3.1.

4. Use of spectral preconditioners in CG. In this section we illustrate the effect of using the approximate eigenpairs on the convergence behavior of PCG. In that respect we consider the 685×685 BUS685 matrix, denoted B_{685} , from the Harwell–Boeing collection. We compute an incomplete Cholesky (IC) factorization CC^T with threshold $4 \cdot 10^{-1}$, which is our first-level preconditioner, and consequently $\sigma = 1$. We apply the various spectral preconditioners to the matrix $A = C^{-1}B_{685}C^{-T}$. As in the previous series of experiments, we use the eigenvectors of the perturbed matrix $A + tE$ to build the preconditioners. Consequently we use eigenelements that have a backward error on the order of t .

The right-hand side is chosen so that the solution of $Ax = b$ is the vector of all ones: $x = (1, \dots, 1)^T$, $b = Ax$. For the numerical experiments the initial guess is the zero vector, and we decide to stop the PCG iterations when the normalized unpreconditioned residual is reduced by 10^{-9} , so that the stopping criterion is independent of the preconditioner. Even though this quantity might be a byproduct of the PCG solver, we explicitly compute the unpreconditioned residual to decide when to stop the iterations.

In Table 4.1 we report on the number of PCG iterations for the various deflation and coarse grid preconditioners for $t = 10^{-12}$ and $t = 10^{-3}$. We vary from 1 to 10 the number ne of eigenelements used to build the preconditioners. The eigenvalues of A obtained with `eig` are reported in Table 4.2. For the smallest perturbation, that is, $t = 10^{-12}$, all the preconditioners behave exactly the same. Because the IC preconditioner has already clustered many of the eigenvalues close to 1, moving the smallest eigenvalues (that vary from $5.67 \cdot 10^{-4}$ to $3.05 \cdot 10^{-2}$; see Table 4.2) exactly to 1 or shifting them by 1 leads to the same behavior of PCG. However, when a perturbation is applied, that is, when the eigenelements are less accurately computed, some differences appear. Both M_1 and M_{+1} perform similarly and outperform the others. Then the various variants that approximate the eigenvalues using Rayleigh quotients perform similarly. The worse behavior is observed for the variants that make use of the approximate eigenvalues.

TABLE 4.1
Number of iterations for PCG for perturbed eigenpairs on the BUS 685 matrix.

$t = 10^{-12}$										
ne	1	2	3	4	5	6	7	8	9	10
All prec.	161	147	132	118	106	95	88	81	81	79
$t = 10^{-3}$										
ne	1	2	3	4	5	6	7	8	9	10
M_1	168	154	142	129	122	114	107	104	101	98
M_1^{rayl}	168	155	152	142	144	139	133	131	126	121
M_1^{eig}	168	156	154	141	143	140	134	194	183	183
$M_{1,\perp}^{rayl}$	168	155	151	139	138	133	125	124	121	117
$M_{1,\perp}^{eig}$	168	156	152	140	139	133	125	184	173	171
M_{+1}	168	154	142	129	121	114	107	104	101	98
M_{+1}^{rayl}	168	155	152	139	138	133	125	124	121	117
M_{+1}^{eig}	168	156	152	140	139	133	125	184	174	171

TABLE 4.2
The ten smallest eigenvalues targeted by the spectral preconditioners on the BUS 685 matrix.

λ_1 5.67e-04	λ_2 2.91e-03	λ_3 3.58e-03	λ_4 4.71e-03	λ_5 6.19e-03
λ_6 8.35e-03	λ_7 1.85e-02	λ_8 2.05e-02	λ_9 3.04e-02	λ_{10} 3.05e-02

5. Concluding remarks. We use the first-order perturbation theory for eigenvalues and eigenvectors to investigate the behavior of deflation and coarse preconditioners for SPD linear systems. Our analysis shows a better stability of the preconditioners M_1 and M_{+1} compared to the other preconditioning variants that exploit some additional properties that are true only for exact eigenpairs. We have established theoretical results that provide us with first-order approximations of the eigenvalues of the preconditioned matrix. These estimates give sharp approximations. They reveal the possible instabilities that might lead to poor preconditioners if inexact eigeninformation is used. They show that targeting small eigenvalues or small clusters may require a backward stable calculation of the eigenelements. An important result of this work is that the efficiency of a spectral preconditioner should not be assessed using only exact eigenpairs. In practice these preconditioners may be built using approximate information, that is, either computed with an eigensolver or obtained by some approximations, as in domain decomposition [5]. Unsymmetric spectral preconditioners exist, and similar results can be developed; this topic will be the basis for some future work.

Appendix A. Some useful equalities.

$$(A.1) \quad v_\ell^T \delta v_s = t(\lambda_s - \lambda_\ell)^{-1} v_\ell^T E v_s \quad \text{if } \ell \neq s,$$

$$(A.2) \quad \delta v_\ell^T v_s = t(\lambda_\ell - \lambda_s)^{-1} v_\ell^T E^T v_s \quad \text{if } \ell \neq s,$$

$$(A.3) \quad v_\ell^T \delta v_\ell = \delta v_\ell^T v_\ell = 0.$$

Proof. For $\ell \neq s$

$$\begin{aligned} v_\ell^T \delta v_s &= v_\ell^T \underline{V}_s (\lambda_s I - B_s)^{-1} \underline{V}_\ell^T E v_s \\ &= (0, \dots, \delta_{j,t}, \dots, 0) (\lambda_s I - B_s)^{-1} \underline{V}_\ell^T E v_s \\ &= (\lambda_s - \lambda_\ell)^{-1} (0, \dots, \delta_{j,t}, \dots, 0) \underline{V}_\ell^T E v_s \\ &= t (\lambda_s - \lambda_\ell)^{-1} v_\ell^T E v_s. \end{aligned}$$

Similarly

$$\begin{aligned} \delta v_\ell^T v_s &= v_\ell^T E^T \underline{V}_\ell (\lambda_\ell I - B_\ell)^{-T} \underline{V}_\ell v_s \\ &= v_\ell^T E^T \underline{V}_\ell (\lambda_\ell I - B_\ell)^{-T} (0, \dots, \delta_j, s, \dots, 0)^T \\ &= v_\ell^T E^T \underline{V}_\ell (0, \dots, \delta_j, s, \dots, 0)^T (\lambda_\ell - \lambda_s)^{-1} \\ &= t v_\ell^T E^T v_s (\lambda_\ell - \lambda_s)^{-1}. \end{aligned}$$

For $\ell = s$,

$$\begin{aligned} \delta v_\ell^T v_\ell &= v_\ell^T \underline{V}_\ell (\lambda_\ell I - B_\ell)^{-1} \underline{V}_\ell^T E v_\ell \\ &= 0 \quad \text{because } v_\ell^T \underline{V}_\ell = (0, \dots, 0) \\ &= v_\ell^T \delta v_\ell. \quad \square \end{aligned}$$

REFERENCES

- [1] R. A. HORN AND C. R. JOHNSON, *Topics in Matrix Analysis*, Cambridge University Press, Cambridge, UK, 1995.
- [2] J. H. BRAMBLE, J. E. PASCIAK, AND A. H. SCHATZ, *The construction of preconditioners for elliptic problems by substructuring I*, Math. Comp., 47 (1986), pp. 103–134.
- [3] B. CARPENTIERI, I. S. DUFF, AND L. GIRAUD, *A class of spectral two-level preconditioners*, SIAM J. Sci. Comput., 25 (2003), pp. 749–765.
- [4] B. CARPENTIERI, L. GIRAUD, AND S. GRATTON, *Additive and Multiplicative Spectral Two-level Preconditioners for General Linear Systems*, Technical Report TR/PA/04/38, CERFACS, Toulouse, France, 2004.
- [5] T. F. CHAN AND T. P. MATHEW, *Domain Decomposition Algorithms*, Acta Numer. 3, Cambridge University Press, Cambridge, UK, 1994, pp. 61–143.
- [6] I. S. DUFF, L. GIRAUD, J. LANGOU, AND E. MARTIN, *Using spectral low rank preconditioners for large electromagnetic calculations*, Int. J. Numer. Methods Engrg., 62 (2005), pp. 416–434.
- [7] I. S. DUFF, R. G. GRIMES, AND J. G. LEWIS, *Sparse matrix test problems*, ACM Trans. Math. Softw., 15 (1989), pp. 1–14.
- [8] M. FISHER, *Minimization algorithms for variational data assimilation*, in Proceedings of the ECMWF seminar “Recent developments in numerical methods for atmospheric modelling,” Reading, UK, 1998, pp. 364–385.
- [9] J. FRANK AND C. VUIK, *On the construction of deflation-based preconditioners*, SIAM J. Sci. Comput., 23 (2001), pp. 442–462.
- [10] G. H. GOLUB AND C. F. VAN LOAN, *Matrix Computations*, 3rd ed., Johns Hopkins University Press, Baltimore, MD, 1996.
- [11] N. J. HIGHAM, *Accuracy and Stability of Numerical Algorithms*, 2nd ed., SIAM, Philadelphia, 2002.
- [12] P. LANCASTER AND M. TISMENETSKY, *The Theory of Matrices*, 2nd ed., Academic Press, New York, 1985.
- [13] R. NABBEN AND C. VUIK, *A comparison of deflation and coarse grid correction applied to porous media flow*, SIAM J. Numer. Anal., 42 (2004), pp. 1631–1647.
- [14] A. PADIY, O. AXELSSON, AND B. POLMAN, *Generalized augmented matrix preconditioning approach and its application to iterative solution of ill-conditioned algebraic systems*, SIAM J. Matrix Anal. Appl., 22 (2000), pp. 793–818.
- [15] J. R. RICE, *A theory of condition*, SIAM J. Numer. Anal., 3 (1966), pp. 287–310.
- [16] G. W. STEWART, *Matrix Algorithms. Volume II: Eigensystems*, SIAM, Philadelphia, 2001.
- [17] J. SUN, *Multiple eigenvalue sensitivity*, Linear Algebra Appl., 138 (1990), pp. 183–211.
- [18] H. WAISMAN, J. FISH, R. S. TUMINARO, AND J. SHADID, *The generalized global basis method*, Int. J. Numer. Methods in Engrg., 61 (2004), pp. 1243–1269.

PRECONDITIONED ITERATIVE METHODS FOR WEIGHTED TOEPLITZ LEAST SQUARES PROBLEMS*

MICHELE BENZI[†] AND MICHAEL K. NG[‡]

Abstract. We consider the iterative solution of weighted Toeplitz least squares problems. Our approach is based on an augmented system formulation. We focus our attention on two types of preconditioners: a variant of constraint preconditioning, and the Hermitian/skew-Hermitian splitting (HSS) preconditioner. Bounds on the eigenvalues of the preconditioned matrices are given in terms of problem and algorithmic parameters, and numerical experiments are used to illustrate the performance of the preconditioners.

Key words. augmented matrix, iterative methods, preconditioning, splittings, constraint preconditioning, eigenvalue bounds, image restoration

AMS subject classifications. Primary, 65F10, 65N22, 65F50; Secondary, 15A06

DOI. 10.1137/040616048

1. Introduction. Linear systems with Toeplitz and Toeplitz-related coefficient matrices arise in many different applications; see [28]. While many efficient algorithms have been developed for solving problems with Toeplitz structure, a few emerging applications lead to Toeplitz-related problems for which the available algorithms are not directly applicable.

In this paper, we consider preconditioners for weighted Toeplitz least squares problems

$$(1.1) \quad \min_x \|Ax - b\|_2^2,$$

where the rectangular coefficient matrix A and the right-hand side b are of the form

$$(1.2) \quad A = \begin{bmatrix} DK \\ \mu I \end{bmatrix} \quad \text{and} \quad b = \begin{bmatrix} Df \\ 0 \end{bmatrix}.$$

Here K is a Toeplitz matrix, D a nonconstant diagonal matrix with real positive entries, f a given right-hand side, $\mu > 0$ a regularization parameter [38], and I the identity matrix. Applications leading to such least squares problems include image reconstruction [17] and nonlinear image restoration [2]. For these applications the size of the problems can be very large, easily over a million unknowns. Moreover, in some applications, the problems have to be solved in real time. Because of the local nature and spatially variant property of weighted Toeplitz matrices, the displacement rank [22] of A can be very large. Efficient and effective preconditioners need to be

*Received by the editors September 30, 2004; accepted for publication (in revised form) by E. Chow May 6, 2005; published electronically March 17, 2006.

<http://www.siam.org/journals/simax/27-4/61604.html>

[†]Department of Mathematics and Computer Science, Emory University, Atlanta, GA 30322 (benzi@mathcs.emory.edu). The work of this author was supported in part by National Science Foundation grant DMS-0207599.

[‡]Department of Mathematics, The University of Hong Kong, Pokfulam Road, Hong Kong, and Department of Mathematics, Hong Kong Baptist University, Kowloon Tong, Hong Kong (mng@maths.hku.hk). The research of this author was supported in part by Hong Kong Research Grants Council grants HKU 7130/02P, HKU 7046/03P, and HKU 7035/04P, and HKU CRCG grants 10205964, 10205775, and 10204437.

investigated to develop fast iterative methods for solving such weighted Toeplitz least squares problems.

Around 1947, Levinson [25], in studying Wiener filters, developed the first fast direct method for solving symmetric positive definite Toeplitz systems $Kx = b$. The method begins with the solution of a 1-by-1 system and then increases the order, using the lower-order solutions to obtain higher-order solutions recursively. In effect, it obtains the inverse Cholesky factorization of K . The fast Cholesky factorization, or Schur's method, was invented by Kailath, Kung, and Morf [21] in 1979 by exploiting the displacement structure of K and its Schur complements. Both methods have a complexity of $O(n^2)$. Superfast direct methods of complexity $O(n \log^2 n)$ were developed in the 1980s; see, for instance, Ammar and Gragg [1]. All these methods make use of the Toeplitz structure (small displacement rank) to derive efficient solvers; see [24, 31]. They are not effective or efficient for Toeplitz-related systems with large displacement rank.¹

An alternative is to make use of iterative methods instead of direct ones. One can use the conjugate gradient method to find the solution to the least squares problem without explicitly forming the normal equations; see [9]. The convergence of the conjugate gradient algorithm and its variants depends on the singular values of the matrix A . If the singular values cluster around a point (away from zero), convergence will be rapid. Thus, to make the algorithm a useful iterative method, one usually preconditions the system. More precisely, one can use the conjugate gradient method to solve

$$\min_y \|b - AP^{-1}y\|_2^2,$$

and then set $x = P^{-1}y$.

Circulant matrices C are often employed to precondition Toeplitz matrices. Since circulant matrices can always be diagonalized by the discrete Fourier matrix [14], the product of C with any n -vector v can be computed easily by fast Fourier transforms (FFTs) in $O(n \log n)$ operations. The matrix-vector multiplication Kv can also be computed by using FFTs in $O(n \log n)$ operations by first embedding K into a $2n$ -by- $2n$ circulant matrix. It follows that the number of operations per iteration is $O(n \log n)$. The convergence rates of the method for Toeplitz systems have been analyzed by several authors; see [28]. The main result is that if the diagonals of the Toeplitz matrix K are the Fourier coefficients of a positive function, then the spectrum of the preconditioned system $C^{-1}K$ will be clustered around 1 and the method will converge superlinearly. Hence the complexity of solving a large class of Toeplitz systems is $O(n \log n)$ operations. Similar convergence results for Toeplitz least squares problems with small displacement rank can be found in [10, 11, 12, 13]. For example, when D is the identity matrix, $(I + C^T C)^{1/2}$ will be a good preconditioner.

However, such a preconditioner is not well suited for least squares problems with large displacement rank in (1.1). Since C is a good preconditioner for K , one might expect the matrix $(I + C^T D \cdot DC)^{1/2}$ to be a good preconditioner for (1.1). However, such a preconditioner is not easy to obtain, and it is too expensive. One possible choice is to approximate D by a circulant matrix C_D ; see [20]. It is clear that the

¹For any square matrix A , consider the matrix operator

$$\Delta(A) = AZ - ZA, \quad Z = (z_{ij}), \quad z_{i+1,i} = 1, \quad z_{i,j} = 0, \text{ otherwise.}$$

The displacement rank of A associated with Δ is defined as the rank of $\Delta(A)$.

eigendecomposition of $(I + C^T C_D^T C_D C)^{1/2}$ can be obtained easily by using FFTs. Unfortunately, if the entries in D exhibit large variations, the performance of this kind of preconditioner is disappointing; i.e., the number of iterations can be very large; see the numerical results in [26].

In this paper, we propose a different approach to solving (1.1), based on an augmented system formulation. If K is m -by- n (with $m \geq n$), it is straightforward to see that problem (1.1) with K given by (1.2) is equivalent to the $(m+n)$ -by- $(m+n)$ augmented system

$$(1.3) \quad \begin{bmatrix} D^{-2} & K \\ K^T & -\mu I \end{bmatrix} \begin{bmatrix} y \\ x \end{bmatrix} = \begin{bmatrix} f \\ 0 \end{bmatrix},$$

where the auxiliary variable $y = D(f - Kx)$ represents a weighted residual. Furthermore, both problem (1.1)–(1.2) and problem (1.3) are equivalent to the normal equations

$$(1.4) \quad (K^T D^2 K + \mu I) x = K^T D^2 f,$$

an n -by- n linear system. As is well known, problem (1.4) is a symmetric positive definite system which may be rather ill-conditioned, even in the presence of regularization. On the other hand, system (1.3) is symmetric indefinite. In recent years, much effort has been put into developing effective preconditioners for linear systems of this type; see [8] for a survey. We identify two preconditioning techniques that appear to be potentially well suited for augmented systems corresponding to weighted Toeplitz least-squares problems: a variant of constraint preconditioning, and the Hermitian/skew-Hermitian splitting (HSS) preconditioner.

The remainder of the paper is organized as follows. In section 2 we recall the idea of constraint preconditioning and discuss how it applies to the regularized weighted least squares problems. In section 3 we recall the Hermitian and skew-Hermitian preconditioner and its application to saddle point (augmented) systems. In section 4 we discuss some theoretical properties of the HSS preconditioner, including eigenvalue bounds for the preconditioned matrices. In section 5 we discuss nonlinear image restoration problems, an important class of applications leading to weighted Toeplitz least squares problems. Numerical experiments are presented in section 6, and in section 7 we briefly sum up with conclusions.

2. Constraint preconditioning. Constraint preconditioning has been extensively used in the solution of saddle point systems arising from mixed finite element formulations of elliptic partial differential equations [3, 16, 32, 33]. The method has also been used for the solution of saddle point (KKT) systems in optimization [15, 18, 23, 27].

We begin by describing constraint preconditioning in the case where K has full column rank ($= n$) and $\mu = 0$ (no regularization). Letting $W = D^{-2} = \text{diag}(w_1, w_2, \dots, w_m)$, the augmented system is

$$\begin{bmatrix} W & K \\ K^T & O \end{bmatrix} \begin{bmatrix} y \\ x \end{bmatrix} = \begin{bmatrix} f \\ 0 \end{bmatrix}.$$

In this paper the constraint preconditioner is the matrix

$$(2.1) \quad P_c = \begin{bmatrix} \gamma I & K \\ K^T & O \end{bmatrix},$$

where γ is the arithmetic average of the diagonal entries of W :

$$\gamma = \frac{w_1 + w_2 + \cdots + w_m}{m}.$$

The application of the preconditioner requires solving a linear system with coefficient matrix P_C in each iteration. Since P_C has a block Toeplitz structure, it is clear that fast solvers can be used to efficiently apply the preconditioner. Note that γI is the nearest Toeplitz (indeed, circulant) approximation to W . Some authors reserve the name “constraint preconditioning” to a preconditioner of the form (2.1) with $\gamma = 1$. We found that such a preconditioner is inferior to the one where γ is given by the arithmetic average of the diagonal entries of W .

Letting $\widehat{W} = \gamma^{-1}W$, the preconditioned matrix is

$$\mathcal{M} = \begin{bmatrix} \gamma I & K \\ K^T & O \end{bmatrix}^{-1} \begin{bmatrix} W & K \\ K^T & O \end{bmatrix} = \begin{bmatrix} (I - \Pi)\widehat{W} + \Pi & O \\ X & I \end{bmatrix},$$

where Π is the orthogonal projector onto the range of K and $X = (K^T K)^{-1} K^T (W - \gamma I)$. Hence, $\lambda = 1$ is an eigenvalue of $\mathcal{M}$ of multiplicity at least $2n$. It can be shown that the remaining eigenvalues are eigenvalues of the symmetric matrix $(I - \Pi)\widehat{W}(I - \Pi)$. Note that in the special case $m = n$, we have $\Pi = I$, and all the eigenvalues of the preconditioned matrix are equal to 1; furthermore,

$$\mathcal{M} - I = \begin{bmatrix} O & O \\ X & O \end{bmatrix} \Rightarrow (\mathcal{M} - I)^2 = O.$$

Therefore the minimum polynomial of the preconditioned matrix has degree 2, and GMRES [34] applied to the preconditioned system delivers the solution in at most two steps, independently of W . Of course, if $m = n$ and K is nonsingular, then the solution is simply given by $x = K^{-1}f$ and $y = 0$, and the augmented system formulation is neither necessary nor recommended.

In the more general case where $m \neq n$, it can be shown that GMRES with constraint preconditioning terminates after at most $m - n + 2$ steps if K is m -by- n ; see [23]. Therefore, constraint preconditioning is an excellent strategy when K has full rank, $\mu = 0$, and $m - n$ is small, no matter what W is.

Things, however, can be quite different when K is rank deficient or highly ill-conditioned and regularization must be included ($\mu \neq 0$). In this case the constraint preconditioner needs to be regularized as well, and the preconditioned matrix becomes

$$\mathcal{M}_\mu = \begin{bmatrix} \gamma I & K \\ K^T & -\mu I \end{bmatrix}^{-1} \begin{bmatrix} W & K \\ K^T & -\mu I \end{bmatrix}.$$

This modified constraint preconditioner has been studied by various authors, most recently in [3]. It can be shown that when $\mu > 0$, the preconditioned matrix $\mathcal{M}_\mu$ has the eigenvalue 1 with multiplicity n , and that all the remaining eigenvalues are real.

If K and W are well conditioned or moderately ill conditioned, the spectrum is clustered around 1 and convergence is fast. If, however, K is very ill conditioned (as it will be if regularization is needed), many of the eigenvalues of $\mathcal{M}_\mu$ will be close to zero and the preconditioner quality will deteriorate; see the numerical experiments in section 6.

3. Hermitian and skew-Hermitian preconditioning. The augmented system (1.3) can be rewritten in the equivalent nonsymmetric form

$$(3.1) \quad \begin{bmatrix} W & K \\ -K^T & \mu I \end{bmatrix} \begin{bmatrix} y \\ x \end{bmatrix} = \begin{bmatrix} f \\ 0 \end{bmatrix} \quad \text{or} \quad Mu = c.$$

Let now

$$(3.2) \quad M = \begin{bmatrix} W & K \\ -K^T & \mu I \end{bmatrix} = \begin{bmatrix} W & O \\ O & \mu I \end{bmatrix} + \begin{bmatrix} O & K \\ -K^T & O \end{bmatrix} = H + S$$

be the splitting of M into its symmetric and skew-symmetric parts. Note that H , the symmetric part of M , is a positive diagonal matrix and hence positive definite. This means that M itself is positive definite, in the sense that

$$z^T M z = z^T H z > 0 \quad \text{for all } z \in \mathbb{R}^{m+n}, \quad z \neq 0,$$

and therefore all the eigenvalues of M have positive real part.

Consider the preconditioning matrix

$$(3.3) \quad P := \frac{1}{2\alpha}(H + \alpha I)(S + \alpha I),$$

where I denotes the $(m+n)$ -by- $(m+n)$ identity and $\alpha > 0$. Note that $H + \alpha I$ is diagonal and positive definite, and $S + \alpha I$ is nonsingular. Thus, this preconditioner is well defined even when some of the entries on the diagonal of W are zero. We note that this case does indeed arise in many situations of practical interest [9, 20]. The scalar $1/(2\alpha)$ in (3.3) has no impact on the preconditioned system $P^{-1}Mu = P^{-1}c$ (or $MP^{-1}w = c$, $u = P^{-1}w$ for right preconditioning), but we retain it as a normalization factor (see below). The stationary iterative process

$$(3.4) \quad u^{(k+1)} = u^{(k)} + P^{-1}(c - Mu^{(k)}), \quad k = 0, 1, \dots,$$

where $u^{(0)}$ is an arbitrary initial guess, has been introduced in [4]. It is known as the HSS iteration.² It was shown in [4] that if H is positive definite, the iteration (3.4) converges to the unique solution of $Mu = c$ for any initial guess and any $\alpha > 0$. This result was extended in [7] to generalized saddle point problems, in which the symmetric part of M may be singular. When H is positive definite, it is possible to give an explicit formula for the value of α that minimizes an upper bound for the spectral radius of the iteration matrix $I - P^{-1}M$ in terms of the extreme eigenvalues of H ; see [4].

Note that each iteration (3.4) requires solving two linear systems:

$$(3.5) \quad (H + \alpha I)u^{(k+\frac{1}{2})} = r^{(k)}, \quad (S + \alpha I)u^{(k+1)} = 2\alpha u^{(k+\frac{1}{2})},$$

where $r^{(k)} = c - Mu^{(k)}$ is the residual at the k th step. In [4, 7] one can find a discussion of the effect of inexact solves in (3.5). Inexact solves may result in considerable savings in terms of overall solution time. In this paper, however, we do not consider the effect of inexact solves.

²In [4] the coefficient matrix M was allowed to have complex entries, and the corresponding splitting was $M = H + S$, where H was the Hermitian part of M and S the skew-Hermitian part. In this paper we consider only matrices with real entries, but we retain the name HSS.

The convergence of (3.4), even with the “optimal” value of α (which may be difficult to estimate in practice) can be quite slow. In [7], it was proposed to use HSS as a preconditioner for Krylov subspace methods (such as GMRES or Bi-CGSTAB [40]); see also [6]. The preconditioner was applied to a variety of saddle point systems arising from the discretization of PDEs, with good results. In this paper, we are interested in using HSS exclusively as a preconditioner. It was observed in [7] that while it is in many cases possible to find efficient solvers for the first of the systems (3.5), solving the second system is usually less straightforward. For the problems studied in this paper, the matrix $H + \alpha I$ is diagonal; therefore the first system in (3.5) can be solved at a cost of just $O(n)$ operations. The second system is of the form

$$(3.6) \quad \begin{bmatrix} \alpha I & K \\ -K^T & \alpha I \end{bmatrix} \begin{bmatrix} v \\ w \end{bmatrix} = \begin{bmatrix} g \\ h \end{bmatrix}.$$

System (3.6) is equivalent to the reduced system

$$(3.7) \quad (K^T K + \alpha^2 I) w = \alpha h + K^T g.$$

Once (3.7) has been solved for w , v can be obtained from $\alpha v = g - Kw$. If K is a Toeplitz matrix, computing v from w can be done in $O(n \log n)$ operations using FFTs. Hence, the problem is reduced to solving linear systems of the type (3.7). But this is precisely the normal equation formulation of a standard regularized least squares problem for a Toeplitz (or Toeplitz-related) matrix K , and as we mentioned in the Introduction, there are efficient methods for solving (3.7). In particular, for certain types of problems that are important in image processing applications there are $O(n \log n)$ methods for solving (3.7). In this case, the cost of applying the HSS preconditioner is $O(n \log n)$ per iteration. See also the discussion at the end of section 6.

We further note that the scalar shift α^2 plays the role of a regularization parameter in the reduced system (3.7). This suggests that perhaps a sensible choice for α could be $\alpha \approx \sqrt{\mu}$, where μ is the regularization parameter for the weighted least squares problem. We will see in the section on numerical experiments that this is a good choice in some cases, but not always.

4. Spectral properties of the HSS preconditioner. The purpose of this section is to derive some bounds on the eigenvalues of the preconditioned matrix $P^{-1}M$ (or MP^{-1} , which has the same spectrum), where M is the nonsymmetric augmented matrix (3.1) and P is the HSS preconditioner (3.3). Although one should be careful not to infer too much from the eigenvalue distribution alone when the preconditioned matrix is nonnormal (as is the case here), our experience is that for the problems considered in this paper the distribution of the eigenvalues is strongly correlated with the rate of convergence of GMRES. In particular, convergence will be fast if most of the eigenvalues are enclosed in a small rectangular region of the complex plane well separated from the origin.

Let $T = I - P^{-1}M$ be the iteration matrix in the stationary iteration (3.4) associated with the HSS splitting. Then it can be shown [4, 7] that $\rho(T) < 1$ (where $\rho(T)$ denotes the spectral radius of T) for all $\alpha > 0$, and therefore the iteration (3.4) is unconditionally convergent. This implies that the eigenvalues of $P^{-1}M$ are contained in the disk of radius 1 centered at $(1, 0)$. In particular, the preconditioned matrix $P^{-1}M$ is *positive stable*: all its eigenvalues have positive real part, a desirable

property for Krylov subspace methods. In addition, if H is positive definite with extreme eigenvalues $\lambda_1 = \lambda_{\max}(H)$, $\lambda_{m+n} = \lambda_{\min}(H)$, then using $\alpha = \sqrt{\lambda_1 \lambda_{m+n}}$ minimizes an upper bound on $\rho(T)$; see [4]. When H is a diagonal matrix, as in (3.1), the “optimal” α is of course explicitly available. However, this is usually not the best choice when P is used as a preconditioner for a Krylov subspace method [6].

More refined bounds on the eigenvalues of $P^{-1}M$ can be found in [36] under the assumption that W is a symmetric positive definite matrix (not necessarily diagonal) and $\mu = 0$. In our setting, this would correspond to a weighted least squares problem without Tikhonov regularization. The results in [36] show that for small α the spectrum of $P^{-1}M$ is real and is entirely contained in two small intervals, one to the right of 0 and the other to the left of 2. The first interval contains n eigenvalues, which tend to 0 as $\alpha \rightarrow 0$, and the second interval contains the remaining m eigenvalues, which tend to 2 as $\alpha \rightarrow 0$. Here $m \geq n$. In practice this means that α should be taken small but not too small. In the case $\mu \neq 0$, however, things appear to be more complicated, and a thorough analysis has so far eluded our efforts.

It is straightforward to see that the iteration matrix $T = I - P^{-1}M$ is similar to

$$\hat{T} = (\alpha I - H)(\alpha I + H)^{-1}(\alpha I - S)(\alpha I + S)^{-1},$$

and therefore T and $\hat{T}$ have the same eigenvalues. Note that $(\alpha I - H)(\alpha I + H)^{-1}$ is symmetric and $(\alpha I - S)(\alpha I + S)^{-1}$, being the Cayley transform of a skew-symmetric matrix, is orthogonal [19, p. 440]. When $W = \text{diag}(w_1, w_2, \dots, w_m)$ we find

$$(\alpha I - H)(\alpha I + H)^{-1} = \begin{bmatrix} E & O \\ O & \omega I \end{bmatrix}, \quad E = (\alpha I - W)(\alpha I + W)^{-1}, \quad \omega = \frac{\alpha - \mu}{\alpha + \mu},$$

and therefore, letting $\Sigma = \alpha I + \frac{1}{\alpha}K^TK$,

$$\hat{T} = \begin{bmatrix} E & O \\ O & \omega I \end{bmatrix} \begin{bmatrix} \alpha I & -K \\ K^T & \alpha I \end{bmatrix} \begin{bmatrix} \frac{1}{\alpha}I - \frac{1}{\alpha^2}K\Sigma^{-1}K^T & -\frac{1}{\alpha}K\Sigma^{-1} \\ \frac{1}{\alpha}\Sigma^{-1}K^T & \Sigma^{-1} \end{bmatrix}.$$

We already know that the eigenvalues of $\hat{T}$ are bounded above by 2. Lower bounds, on the other hand, are more difficult to come by. However, the particular choice $\alpha = \mu$ (which implies $\omega = 0$) leads to a significant simplification:

$$\hat{T} = \begin{bmatrix} E - \frac{2}{\mu}EK\Sigma^{-1}K^T & -2EK\Sigma^{-1} \\ O & O \end{bmatrix}.$$

In this case, the spectrum of $\hat{T}$ consists of the spectrum of the m -by- m matrix $E - \frac{2}{\mu}EK\Sigma^{-1}K^T$, plus the eigenvalue 0 with multiplicity n . It follows that the preconditioned matrix $P^{-1}M$, which is similar to $I - \hat{T}$, has m eigenvalues equal to 1, while the remaining n eigenvalues are of the form $1 - \nu_i$ where $\nu_1, \dots, \nu_m$ are the eigenvalues of the matrix

$$E - \frac{2}{\mu}EK\Sigma^{-1}K^T = E \left(I - \frac{2}{\mu}K\Sigma^{-1}K^T \right) = EF,$$

where we have let $F = I - \frac{2}{\mu}K\Sigma^{-1}K^T$. Note that both E and F are symmetric. The diagonal matrix E is negative definite, provided that $0 < \mu < w_{\min} \equiv \min_i \{w_i\}$, and it has spectral norm $\|E\|_2 < 1$. The eigenvalues of F can be easily expressed in terms

of the singular values σ_i of K as follows:

$$\lambda_i(F) = 1 - 2 \left(\frac{\sigma_i^2}{\mu^2 + \sigma_i^2} \right) = \frac{\mu^2 - \sigma_i^2}{\mu^2 + \sigma_i^2},$$

and therefore

$$\|F\|_2 = \max_i \frac{|\mu^2 - \sigma_i^2|}{\mu^2 + \sigma_i^2} \leq 1.$$

If K has full column rank and $0 < \mu \leq \sigma_{\min}(K)$, then F is negative semidefinite. Therefore if $\mu < \min\{\min_i\{w_i\}, \sigma_{\min}(K)\}$, then the eigenvalues of EF are all real and between 0 and 1, and so are those of $P^{-1}M$. Note, however, that the condition $0 < \mu \leq \sigma_{\min}(K)$ is unrealistic in the case of discrete ill-posed problems, where typically $\sigma_{\max}(K) \approx 1$ and $\sigma_{\min}(K) \approx 0$. Therefore, F is typically indefinite. In this case, provided that E is negative definite, the spectrum of the matrix EF is real and contained in the interval $(-1, 1)$. Clearly, the eigenvalues of $P^{-1}M$ are bounded from below by $1 - \|E\|_2$. To get a lower bound for this expression, recall that

$$\|E\|_2 = \max_{1 \leq i \leq m} \frac{|\mu - w_i|}{|\mu + w_i|} = \max_{1 \leq i \leq m} \frac{1 - \mu x_i}{1 + \mu x_i},$$

where we have set $x_i = 1/w_i$. The function

$$\phi(x) = \frac{1 - \mu x}{1 + \mu x}$$

is monotonically decreasing; hence it is maximized when $x = x_* = 1/w_{\max}$, where $w_{\max}$ denotes the largest entry in W . Moreover, $\phi(x_*) = \frac{2\mu}{\mu + w_{\max}}$. Hence, we have established the following lower bound:

$$\lambda(P^{-1}M) \geq \frac{2\mu}{\mu + w_{\max}}.$$

On the other hand, if $w_{\min} < \mu \leq w_{\max}$, then both E and F are indefinite and $P^{-1}M$ will have eigenvalues with nonzero imaginary part, in general. In this case, the imaginary part of the eigenvalues is between -1 and 1 , and the real part is between $\frac{2\mu}{\mu + w_{\max}}$ and 2 . Hence, we have established the following result.

THEOREM 4.1. *Let $w_{\min}$, $w_{\max}$ denote the smallest and largest entries, respectively, of the diagonal matrix W , with $w_{\min} > 0$. Let M be as in (3.2), and let P denote the corresponding HSS preconditioner with $\alpha = \mu$. Also, let $a := 2\mu/(\mu + w_{\max})$. Then $P^{-1}M$ has m eigenvalues equal to 1, and the remaining $n - m$ eigenvalues are contained in the region*

$$\mathcal{R} = \{x + iy \in \mathbb{C}; a \leq x < 2, -1 < y < 1\} \cap D(1, 1),$$

where $D(1, 1) = \{z \in \mathbb{C}; |z - 1| < 1\}$. If, moreover, the regularization parameter μ satisfies $\mu < w_{\min}$, then the eigenvalues of $P^{-1}M$ are all real and contained in $\{1\} \cup [a, 2)$.

Note that the inclusion region, $\mathcal{R}$, is independent of K . The theorem implies that no matter how ill-conditioned K is, the eigenvalues of $P^{-1}M$ remain bounded away from zero, and they will all be real as long as μ is sufficiently small. Furthermore, for a fixed value of $w_{\max}$ the real part of the eigenvalues of $P^{-1}M$ not equal to 1

increases as μ grows, and all the eigenvalues not equal to 1 tend to the real value $\lambda = 2$ as $\mu \rightarrow \infty$. Note, however, that the preconditioned matrix $P^{-1}M$ is generally nonnormal. Hence, the condition number of the eigenvector matrix may also play a role in the convergence of a Krylov subspace solver like GMRES; see [35, pp. 206–207]. Since the eigenvectors generally depend on K , so does the rate of convergence. Bounds on the location of the eigenvalues like the one given in the foregoing theorem may be useless if the eigenvector matrix of $P^{-1}M$ is highly ill-conditioned. Unfortunately, bounds on the condition number of the eigenvector matrix are even harder to come by than eigenvalue bounds. We have numerically estimated the condition number of the eigenvector matrix in a few cases of interest; see the results in section 6.

In practice, μ is typically small and the largest entry in W can be rather large; hence the lower bound $a := 2\mu/(\mu + w_{\max})$ can be tiny. Numerical experiments indicate that this lower bound can be orders of magnitude smaller than the actual minimum of the real part of the eigenvalues of $P^{-1}M$. Furthermore, as we will see, taking $\alpha = \mu$ does not lead to very good performance: better results are obtained with a different value of α . Unfortunately, it is difficult to give meaningful bounds on the eigenvalues of $P^{-1}M$ when $\alpha \neq \mu$.

5. Nonlinear image restoration. In this section we discuss nonlinear image restoration problems, an important class of applications leading to weighted Toeplitz least squares problems.

In the literature on image restoration, a blurred image is often modeled as the linear convolution of an original image with the *point spread function* of the blur. However, in practice, image formation systems or image sensors usually incorporate a built-in nonlinearity. For instance, the nonlinearity is introduced in the transformation of light intensity to the output units of the imaging system such as current intensity in photo-electric systems and photographic films. The modeling of sensor nonlinearities was first studied by Andrews and Hunt [2]. In matrix-vector notation, the general space-invariant imaging system with additive noise can be represented by the following nonlinear equation:

$$(5.1) \quad f = s(Kx) + \eta,$$

where f , x , and η represent the observed, the original image, and the noise vectors, respectively. Here $s(\cdot)$ denotes a point nonlinearity, and the matrix K is a blurring matrix. Because of the blurring process, the boundary values of f are not completely determined by the original image x inside the scene. They are also affected by the values of x outside the scene. We remark that K is block Toeplitz with Toeplitz blocks (BTTB) when zero boundary conditions are applied, and block Toeplitz-plus-Hankel with Toeplitz-plus-Hankel blocks (BTHTHB) when reflective boundary conditions are used [29]. Both theoretical and experimental results in [29, 30] show that the restoration results using reflective boundary conditions are better than those using zero boundary conditions.

Different nonlinear image restoration algorithms have been proposed and analyzed. For instance, Andrews and Hunt [2] proposed using Taylor series expansion about the mean value of the observed image to approximate (5.1) by a linear equation. An approximate filter for linear image restoration can then be derived. Trussell and Hunt [39] applied the maximum a posteriori probability (MAP) estimation scheme in nonlinear image restoration algorithms. This approach results in an iterative solution algorithm the computational complexity of which is very large. Tekalp and

Pavlović [37] also proposed to transform the noisy and blurred image into “the exposure domain” using the inverse of the nonlinear sensor characteristics. A linear minimum mean square error deconvolution filter was derived by using the linear convolutional model in the presence of multiplicative noise in the exposure domain. In this paper, we consider solving nonlinear least squares problems with regularization,

$$(5.2) \quad \min_x \|f - s(Kx)\|_2^2 + \mu \|x\|_2^2,$$

to restore the original image. In [41], Zervakis and Venetsanopoulos have considered using the Gauss–Newton method to solve the nonlinear least squares problem (5.2). Given an initial guess $x^{(0)}$, for $j = 0, 1, \dots$, we solve the linear least squares problem

$$(5.3) \quad x^{(j+1)} = \arg \min_x \left\{ \|f - s(Kx^{(j)}) - D_s^{(j)} K(x - x^{(j)})\|_2^2 + \mu \|x\|_2^2 \right\}$$

until $\|x^{(j+1)} - x(\mu)\|_2$ is small enough, where $x(\mu)$ is the solution of (5.2) with regularization parameter μ . Here $D_s^{(j)}$ is a diagonal matrix with diagonal entries

$$[D_s^{(j)}]_{ii} = \frac{\partial s}{\partial \xi} \Big|_{\xi = \sum_l K_{il} x_l^{(j)}}.$$

We remark that under the practical assumption on the nonlinear function $s(\cdot)$, the diagonal entries of $D_s^{(j)}$ are always positive values; see Andrew and Hunt [2]. The least squares problem (5.3) is equivalent to

$$[\mu I + K^T (D_s^{(j)})^2 K] (x^{(j+1)} - x^{(j)}) = K^T D_s^{(j)} [f - s(Kx^{(j)})] - \mu x,$$

i.e.,

$$(5.4) \quad [\mu I + K^T (D_s^{(j)})^2 K] x^{(j+1)} = K^T D_s^{(j)} [f - s(Kx^{(j)}) + D_s^{(j)} Kx^{(j)}].$$

In the Gauss–Newton method, it is important to choose a good initial guess. We propose the following algorithm to compute $x^{(0)}$:

(1) Solve $f = s(\hat{f})$ for $\hat{f}$ (in many practical applications, s^{-1} can be easily obtained).

(2) Choose a suitable parameter μ_0 and solve

$$(5.5) \quad (\mu_0 I + K^T K) x^{(0)} = K^T \hat{f}.$$

Numerical results in the next section show that the use of image $x^{(0)}$ as initial guess is quite effective for the nonlinear least squares problem in (5.2). We also see that in computing the initial guess and in each Gauss–Newton iteration, we are required to solve BTTB-related systems with diagonal matrix $D = I$ (cf. (5.5)) and $D = (D_s^{(j)})^2$ (cf. (5.4)), respectively.

6. Numerical examples. We first test the two preconditioners on a sequence of weighted Toeplitz least squares problems of the form (3.1), where $K = (k_{ij})$ is an n -by- n full Toeplitz matrix defined by $k_{ij} = 1/(\sqrt{|i-j|} + 1)$, D is an n -by- n positive diagonal random matrix with $\kappa_2(D) \approx 10^3$ (so that $\kappa_2(W) \approx 10^6$), and $\mu = 10^{-3}$. Note that because D is randomly generated, this set of test problems should not

TABLE 6.1
Comparison of preconditioners, 1D problem with well-conditioned K , $\mu = 0.001$.

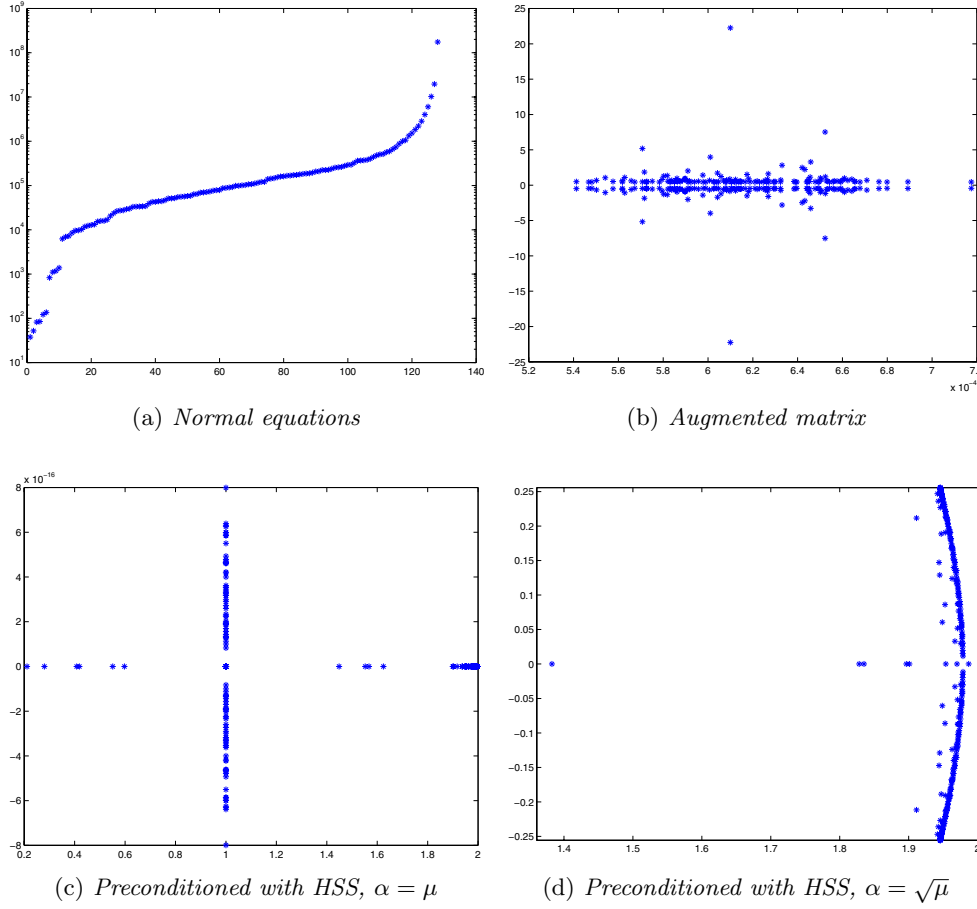
n	CG	GMRES	HSS($\alpha = \mu$)	HSS($\alpha = 0.05$)	HSS($\alpha = \sqrt{\mu}$)	CP
64	159	48	13	7	6	3
128	424	66	13	7	7	3
256	> 1000	90	18	7	7	3
512	> 1000	132	57	16	17	3
1024	> 1000	168	72	14	16	3

be thought of as a sequence of increasingly finer discretizations of one and the same underlying continuous problem. Nevertheless, all the generated problems share similar characteristics. Because of the element of randomness in D , we repeated this set of test runs several times. The iteration counts were found to be fairly stable. Here we report average iteration numbers over five test runs. The initial guess was always the zero vector, and the stopping criterion was a reduction of the initial residual norm by seven orders of magnitude.

Iteration counts for various methods are shown in Table 6.1. Here we have denoted by “CG” the conjugate gradient method on the normal equations (1.4) without preconditioning, by “GMRES” the GMRES method applied to the augmented system (3.1) without preconditioning, and by “HSS(α)” the GMRES method with HSS preconditioning with parameter α applied to the augmented system. Finally, under “CP” we report results for the (modified) constraint preconditioner applied to the augmented system (1.3). As the results show, the number of CG iterations (without preconditioning) grows very fast as the size of the least squares problem goes from $n = 64$ to $n = 1024$. The number of GMRES iterations grows much more slowly, but it is still prohibitive (note that this is full GMRES; no restart was used). When HSS preconditioning is used, the convergence is generally faster. Using $\alpha = \mu$, however, is not satisfactory, since the number of iterations still grows fairly quickly with n , and of course each iteration is now more expensive due to the additional $O(n \log n)$ operations per iteration required by the preconditioner solve. Much better results are obtained with $\alpha = 0.05$. In the last column we show the iteration counts for $\alpha = \sqrt{\mu} \approx 0.0316$, a value not too different from 0.05. The results are very similar to the case with $\alpha = 0.05$, showing that the performance of the preconditioner is not overly sensitive to the value of α . A nice property of the HSS preconditioner is that for the problems considered in this paper, the same value of α works well for all values of n . Note also that the number of iterations appears to be leveling off as n grows. Nevertheless, the results in the last column show that the modified constraint preconditioner works best, with convergence taking place in three iterations regardless of problem size.

In Figure 6.1 we show plots of the eigenvalues of the following matrices: the normal equation matrix $K^T D^2 K + \mu I$, the augmented matrix M in (3.1), the matrix M preconditioned by the HSS preconditioner with $\alpha = \mu$, and M preconditioned by the HSS preconditioner with $\alpha = \sqrt{\mu}$. The eigenvalue plot for the case $\alpha = 0.05$ is qualitatively similar to that for $\alpha = \sqrt{\mu}$. Here $n = m = 128$.

Note the logarithmic scale on the y -axis in Figure 6.1(a), and the 10^{-4} factor multiplying the x -axis in Figure 6.1(b). Also note that the tiny nonzero imaginary part of the eigenvalues in Figure 6.1(c) is due to round-off: the eigenvalues are actually all real, and in fact in this example the conditions for a real spectrum given in Theorem 4.1 are satisfied. In this problem the matrix K is fairly well conditioned, and most of the ill-conditioning is due to the weighting matrix D . We have also observed

FIG. 6.1. *Eigenvalue plots, 1D problem with well-conditioned K .*

that the preconditioned matrix appears to be diagonalizable with a well-conditioned matrix of eigenvectors, a typical value of the condition number being around 60.

In Figure 6.2(a) we show the spectrum of the augmented matrix preconditioned with the modified constraint preconditioner: note the tight clustering of the eigenvalues near 1.

The second set of experiments is similar to the first one, but now K has entries

$$k_{ij} = \frac{1}{\sqrt{2\pi}\sigma} \exp \left[\frac{-|i-j|^2}{(2\sigma^2)} \right].$$

Choosing $\sigma = 2$ results in a highly ill-conditioned Toeplitz matrix with rapidly decaying singular values. The results are reported in Table 6.2. The value $\alpha = 6 \cdot 10^{-5}$ was found to be the optimal value of α for $n = 64$, and we used the same value of α for all values of n . Under “HSS($\alpha = \alpha_{best}$)” we report the best result we could obtain by fine-tuning α . In all cases the value of α_{best} was quite close to $6 \cdot 10^{-5}$, and the corresponding iteration counts not very far apart. This shows, again, that a good value of α can be found on a small problem and used for larger problems of the same type.

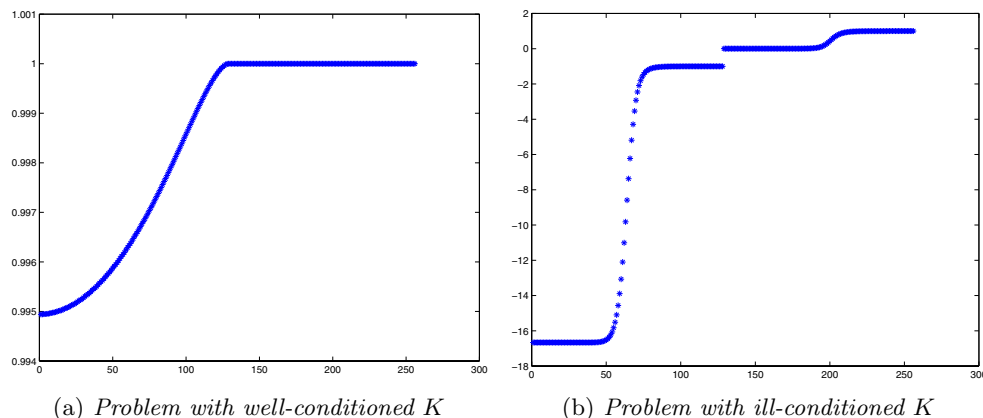


FIG. 6.2. Eigenvalue plots, modified constraint preconditioner.

TABLE 6.2
Comparison of preconditioners, 1D problem with ill-conditioned K , $\mu = 0.001$.

n	CG	GMRES	HSS($\alpha = \mu$)	HSS($\alpha = 6 \cdot 10^{-5}$)	HSS($\alpha = \alpha_{best}$)	CP
64	761	117	55	43	43	37
128	> 1000	224	106	74	74	67
256	> 1000	410	159	95	84	125
512	> 1000	770	236	127	117	271
1024	> 1000	> 1000	250	129	117	553

We further observe that this problem is much harder than the previous one, and that using HSS with a good value of α results in much faster convergence than using the modified constraint preconditioner. Also note that, again, the number of iterations with the HSS preconditioners appears to level off as n increases. Eigenvalue plots for this problem are given in Figure 6.2(b) for the modified constraint preconditioner and in Figure 6.3 for the remaining cases. Note the presence of tight clusters of tiny eigenvalues in all cases, except for HSS preconditioning with the optimal α (Figure 6.3(d)), for which there is a visible (albeit small) gap separating the smallest eigenvalues from zero. The condition number of the eigenvectors of the preconditioned matrix is now $O(10^4)$ or less. While larger than in the previous example, this value can still be considered moderate.

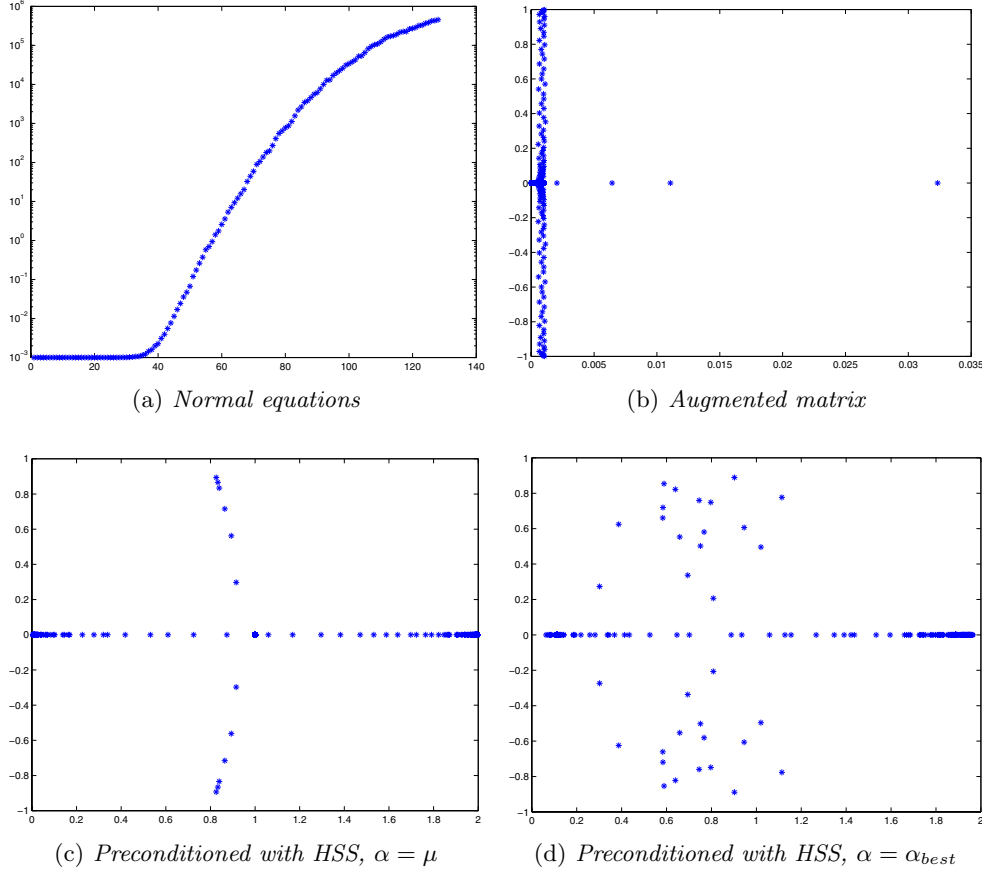
Next, we consider the solution of nonlinear image restoration problems. We test two 128×128 images: Bridge (Figure 6.4(a)) and Cameraman (Figure 6.4(b)). The pointwise nonlinearity employed is of the logarithmic form,

$$s(x) = 30 \log(x)$$

(tested in [41]), and the discrete point spread function is given by

$$k(x, y) = \exp \left[\frac{-(x^2 + y^2)}{2} \right].$$

We construct the observed image by forming the vector $f = s(Kx) + \eta$, where x is a vector formed by row ordering the original image. Here the reflective boundary condition is employed, and therefore the matrix K is BTHTHB. In our tests, the

FIG. 6.3. Eigenvalue plots, 1D problem with ill-conditioned K .

noise η is set to Gaussian white noise with noise-to-signal ratios of 40dB and 30dB. Observed images for noise-to-signal ratio of 40dB are shown in Figures 6.4(c) and (d).

In the preconditioned method for solving the system (5.5), we use the zero vector as the initial guess, and the stopping criterion is $\|r^{(i)}\|_2 / \|K^T \hat{g}\|_2 < 10^{-7}$, where $r^{(i)}$ is the residual after i iterations.

In the preconditioned GMRES(30) methods for solving the augmented system corresponding to (5.4), we use the solution of (5.5) as the initial guess, and the stopping criterion is

$$\frac{\|r^{(i)}\|_2}{\|K^T D_s^{(j)}[f - s(Kx^{(j)}) + D_s^{(j)}Kx^{(j)}]\|_2} < 10^{-7},$$

where $r^{(i)}$ is the residual after the i th iteration.

In Figures 6.5(a) and (b), we present our initial guesses for the restored images, i.e., the solutions of (5.5). The optimal regularization parameter μ_0 is chosen such that it minimizes the relative error of $x^{(0)}(\mu_0)$; i.e., it minimizes

$$R_0 = \frac{\|x - x^{(0)}(\mu_0)\|_2}{\|x\|_2}.$$

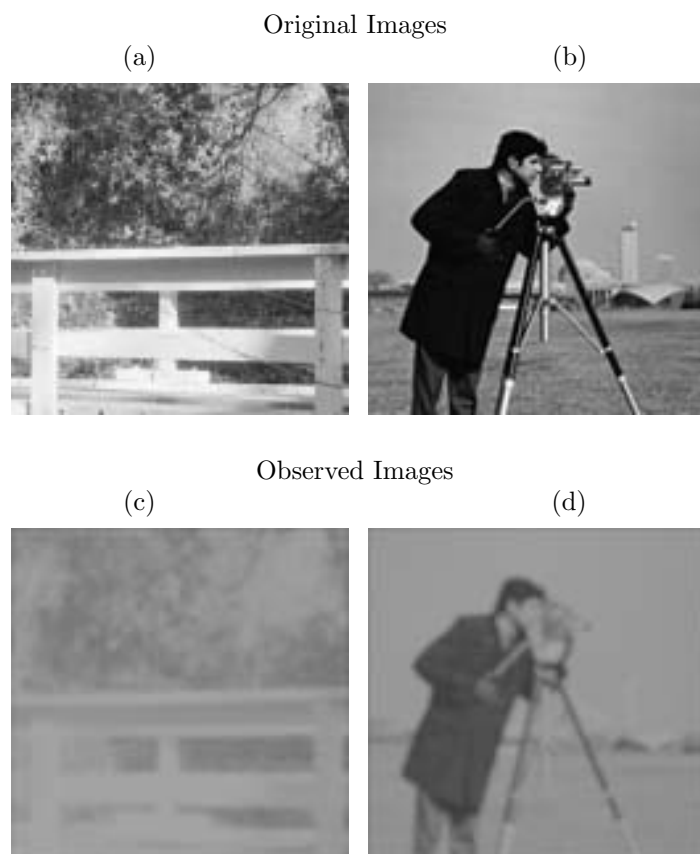


FIG. 6.4. Original and observed images of Bridge (left) and Cameraman (right).

TABLE 6.3

Relative errors in the restored images (the number in the bracket refers to the corresponding optimal regularization parameters).

SNR	Bridge		Cameraman	
	R_0	R_1	R_0	R_1
40dB	0.0532 (2.0×10^{-3})	0.0512 (7.5×10^{-3})	0.0548 (5.0×10^{-3})	0.0504 (3.0×10^{-4})
30dB	0.0622 (1.0×10^{-2})	0.0612 (4.0×10^{-3})	0.0824 (2.0×10^{-2})	0.0801 (1.2×10^{-3})

The restored images are shown in Figures 6.5(c) and (d). Again, the optimal regularization parameter μ is chosen such that

$$R_1 = \frac{\|x - x(\mu)\|_2}{\|x\|_2}$$

is minimized, where $x(\mu)$ is the solution of (5.4). It can be seen from the figures that the quality of restored images is visually better than that of initial guess images. The errors R_0 and R_1 are reported in Table 6.3, and their corresponding optimal regularization parameters are also given. We also found that the relative errors of the restored images computed by several iterations of the Gauss–Newton method do

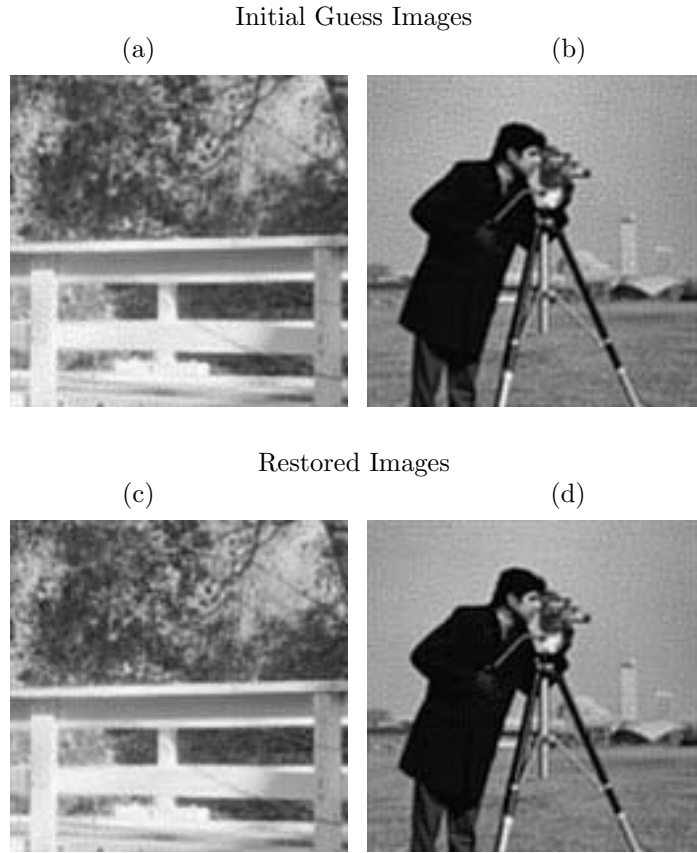


FIG. 6.5. Initial guess and restored images of Bridge (left) and Cameraman (right).

TABLE 6.4
Comparison of preconditioners for the nonlinear image restoration problem.

Method	Bridge		Cameraman	
	30dB	40dB	30dB	40dB
CG	289	273	334	768
Circulant-PCG	183	167	189	195
GMRES	> 1000	> 1000	> 1000	> 1000
HSS ($\alpha = 0.01$)	82	76	98	69
HSS ($\alpha = 0.05$)	48	46	51	45
HSS ($\alpha = 0.10$)	39	37	38	49
HSS ($\alpha = 0.50$)	95	87	105	270
CP	312	295	378	> 1000

not improve, and the visual quality of these restored images are about the same as in Figures 6.5(c) and (d).

The number of iterations for CG and GMRES(30) with various preconditioners for solving the augmented system corresponding to (5.4) is listed in Table 6.4. We see that HSS-preconditioned GMRES(30) converges much faster than the other methods tested. These include the unpreconditioned conjugate gradient method applied to (5.4) (under “CG”), unpreconditioned GMRES(30) applied to the corresponding

augmented system (under “GMRES”), the preconditioned conjugate gradient method with the circulant preconditioner (under “Circulant-PCG”), and GMRES(30) with the modified constraint preconditioner (under “CP”). We note that the cost per iteration of the conjugate gradient and the GMRES methods depends mainly on the Toeplitz matrix-vector multiplications. This product can be formed with two two-dimensional FFTs of size $2n$ -by- $2n$, i.e., at about the same cost as four two-dimensional FFTs of size n -by- n . With circulant, HSS, and CP preconditioners, we have to solve their corresponding systems in each preconditioned CG/GMRES iteration, and thanks to their special structure, they can be solved by using two two-dimensional FFTs of size n -by- n . Thus the overhead of using different preconditioners is about 50%. Therefore the costs per iteration using different preconditioners are comparable.

7. Conclusions and future work. We have investigated the application of two types of preconditioners to the solution of weighted Toeplitz least squares problems written in augmented system form. Theoretical considerations and numerical experiments indicate that the modified constraint preconditioner is extremely effective when the problem is well conditioned or moderately ill conditioned and $m = n$. On the other hand, this preconditioner is not effective for the highly ill-conditioned cases that arise from the solution of ill-posed problems.

The HSS preconditioner was found to perform quite well (though not as well as constraint preconditioning) for moderately ill-conditioned problems. For harder problems the convergence rate, while not spectacular, was much better than with constraint preconditioning; moreover, for large enough n the rate of convergence appears to be constant. A positive observation is that the iteration parameter α in the HSS preconditioner can be fine-tuned on a small problem instance and then used, with good results, on larger problems.

Although we limited ourselves to the simplest possible form of the regularization term, the techniques used here can easily be applied to the case where the regularization term is of the form $\mu\|Lx\|_2^2$, where L is a smoothing operator. In particular, HSS preconditioning can be efficiently applied as long as L has a nice structure, such as Toeplitz or Toeplitz-related.

We conclude by mentioning another possible use of HSS preconditioning. In some applications (see, e.g., [5]) it is necessary to solve problems of the form

$$(7.1) \quad (K^T K + \beta D)x = K^T f,$$

where K is a Toeplitz-related matrix, $\beta > 0$, and D is a diagonal matrix with positive entries on the main diagonal. Clearly, problem (7.1) is equivalent to the augmented system

$$(7.2) \quad \begin{bmatrix} I & K \\ -K^T & \beta D \end{bmatrix} \begin{bmatrix} y \\ x \end{bmatrix} = \begin{bmatrix} f \\ 0 \end{bmatrix},$$

for which the HSS preconditioner is well suited.

Acknowledgment. The first author would like to thank the University of Hong Kong for its kind support and warm hospitality during his stay there in January, 2004.

REFERENCES

- [1] G. AMMAR AND W. B. GRAGG, *Superfast solution of real positive definite Toeplitz systems*, SIAM J. Matrix Anal. Appl., 9 (1988), pp. 61–76.

- [2] H. ANDREWS AND B. HUNT, *Digital Image Restoration*, Prentice-Hall, Englewood Cliffs, NJ, 1977.
- [3] O. AXELSSON AND M. NEYTCHIEVA, *Preconditioning methods for linear systems arising in constrained optimization problems*, Numer. Linear Algebra Appl., 10 (2003), pp. 3–31.
- [4] Z.-Z. BAI, G. H. GOLUB, AND M. K. NG, *Hermitian and skew-Hermitian splitting methods for non-Hermitian positive definite linear systems*, SIAM J. Matrix Anal. Appl., 24 (2003), pp. 603–626.
- [5] M. BELGE, M. E. KILMER, AND E. L. MILLER, *Wavelet domain image restoration with adaptive edge-preserving regularization*, IEEE Trans. Image Process., 9 (2000), pp. 597–608.
- [6] M. BENZI, M. J. GANDER, AND G. H. GOLUB, *Optimization of the Hermitian and skew-Hermitian splitting iteration for saddle-point problems*, BIT, 43 (2003), pp. 881–900.
- [7] M. BENZI AND G. H. GOLUB, *A preconditioner for generalized saddle point problems*, SIAM J. Matrix Anal. Appl., 26 (2004), pp. 20–41.
- [8] M. BENZI, G. H. GOLUB, AND J. LIESEN, *Numerical solution of saddle point problems*, Acta Numer., 14 (2005), pp. 1–137.
- [9] A. BJÖRCK, *Numerical Methods for Least Squares Problems*, SIAM, Philadelphia, 1996.
- [10] R. H. CHAN, J. G. NAGY, AND R. J. PLEMMONS, *FFT-based preconditioners for Toeplitz-block least squares problems*, SIAM J. Numer. Anal., 30 (1993), pp. 1740–1768.
- [11] R. H. CHAN, J. G. NAGY, AND R. J. PLEMMONS, *Circulant preconditioned Toeplitz least squares iterations*, SIAM J. Matrix Anal. Appl., 15 (1994), pp. 80–97.
- [12] R. H. CHAN, J. G. NAGY, AND R. J. PLEMMONS, *Displacement preconditioner for Toeplitz least squares iterations*, Electron. Trans. Numer. Anal., 2 (1994), pp. 44–56.
- [13] R. H. CHAN, J. G. NAGY, AND R. J. PLEMMONS, *Generalization of Strang’s preconditioner with applications to Toeplitz least squares problems*, Numer. Linear Algebra Appl., 3 (1996), pp. 45–64.
- [14] P. J. DAVIS, *Circulant Matrices*, John Wiley and Sons, New York, 1979.
- [15] N. DYN AND W. E. FERGUSON, JR., *The numerical solution of equality-constrained quadratic programming problems*, Math. Comp., 41 (1983), pp. 165–170.
- [16] R. E. EWING, R. LAZAROV, P. LU, AND P. VASSILEVSKI, *Preconditioning indefinite systems arising from mixed finite element discretization of second order elliptic problems*, in Lectures Notes in Math. 1457, O. Axelsson and L. Yu. Kolotilina, eds., Springer-Verlag, Berlin, 1990, pp. 28–43.
- [17] J. A. FESSLER AND S. D. BOOTH, *Conjugate-gradient preconditioning methods for shift-variant PET image reconstruction*, IEEE Trans. Image Process., 8 (1999), pp. 688–699.
- [18] N. I. M. GOULD, M. E. HRIBAR, AND J. NOCEDAL, *On the solution of equality constrained quadratic programming problems arising in optimization*, SIAM J. Sci. Comput., 23 (2001), pp. 1376–1395.
- [19] R. A. HORN AND C. R. JOHNSON, *Topics in Matrix Analysis*, Cambridge University Press, Cambridge, UK, 1994.
- [20] X.-Q. JIN, *A preconditioner for constrained and weighted least squares problems with Toeplitz structure*, BIT, 36 (1996), pp. 101–109.
- [21] T. KAILATH, S. Y. KUNG, AND M. MORF, *Displacement ranks of a matrix*, Bull. Amer. Math. Soc., 1 (1979), pp. 769–773.
- [22] T. KAILATH AND A. H. SAYED, *Displacement structure: Theory and applications*, SIAM Rev., 37 (1995), pp. 297–386.
- [23] C. KELLER, N. I. M. GOULD, AND A. J. WATHEN, *Constraint preconditioning for indefinite linear systems*, SIAM J. Matrix Anal. Appl., 21 (2000), pp. 1300–1317.
- [24] M. KIM, H. PARK, AND L. ELDÉN, *Comparison of algorithms for Toeplitz least squares and symmetric positive definite linear systems*, in Structured Matrices in Mathematics, Computer Science, and Engineering, Vol. II, Contemporary Mathematics, V. Olshevsky, ed., American Mathematical Society, Providence, RI, 2001, pp. 73–99.
- [25] N. LEVINSON, *The Wiener r.m.s. (root-mean-square) error criterion in filter design and prediction*, J. Math. Phys., 25 (1947), pp. 261–278.
- [26] F.-R. LIN, M. K. NG, AND W.-K. CHING, *Factorized banded inverse preconditioners for matrices with Toeplitz structure*, SIAM J. Sci. Comput., 26 (2005), pp. 1852–1870.
- [27] L. LUKŠAN AND J. VLČEK, *Indefinitely preconditioned inexact Newton method for large sparse equality constrained non-linear programming problems*, Numer. Linear Algebra Appl., 5 (1998), pp. 219–247.
- [28] M. K. NG, *Iterative Methods for Toeplitz Systems*, Numerical Mathematics and Scientific Computation, Oxford University Press, New York, 2004.
- [29] M. K. NG, R. H. CHAN, AND W.-C. TANG, *A fast algorithm for deblurring models with Neumann boundary conditions*, SIAM J. Sci. Comput., 21 (1999), pp. 851–866.

- [30] M. NG AND A. YIP, *A fast MAP algorithm for high-resolution image reconstruction with multisensors*, Multidimens. Systems Signal Process., 12 (2001), pp. 143–164.
- [31] H. PARK AND L. ELDÉN, *Stability analysis and fast algorithms for triangularization of Toeplitz matrices*, Numer. Math., 76 (1997), pp. 383–400.
- [32] I. PERUGIA AND V. SIMONCINI, *Block-diagonal and indefinite symmetric preconditioners for mixed finite element formulations*, Numer. Linear Algebra Appl., 7 (2000), pp. 585–616.
- [33] M. ROZLOŽNÍK AND V. SIMONCINI, *Krylov subspace methods for saddle point problems with indefinite preconditioning*, SIAM J. Matrix Anal. Appl., 24 (2002), pp. 368–391.
- [34] Y. SAAD AND M. H. SCHULTZ, *GMRES: A generalized minimal residual algorithm for solving nonsymmetric linear systems*, SIAM J. Sci. Statist. Comput., 7 (1986), pp. 856–869.
- [35] Y. SAAD, *Iterative Methods for Sparse Linear Systems*, 2nd ed., SIAM, Philadelphia, PA, 2003.
- [36] V. SIMONCINI AND M. BENZI, *Spectral properties of the Hermitian and skew-Hermitian splitting preconditioner for saddle point problems*, SIAM J. Matrix Anal. Appl., 26 (2004), pp. 377–389.
- [37] A. TEKALP AND G. PAVLOVIĆ, *Restoration of scanned photographic images*, in Digital Image Restoration, A. Katsaggelos, ed., Springer-Verlag, Berlin, 1991, pp. 209–240.
- [38] A. N. TIKHONOV AND V. Y. ARSENIN, *Solutions of Ill-Posed Problems*, John Wiley and Sons, New York, 1977.
- [39] H. TRUSSELL AND B. HUNT, *Improved methods of maximum a posteriori restoration*, IEEE Trans. Comput., 27 (1979), pp. 57–62.
- [40] H. A. VAN DER VORST, *Bi-CGSTAB: A fast and smoothly converging variant of Bi-CG for the solution of nonsymmetric linear systems*, SIAM J. Sci. Statist. Comput., 13 (1992), pp. 631–644.
- [41] M. ZERVAKIS AND A. VENETSANOPOULOS, *Iterative algorithms with fast convergence rates in nonlinear image restoration*, in SPIE 1452, Image Processing Algorithms and Techniques, SPIE, Bellingham, WA, 1991, pp. 90–103.

A CLASS OF NONSYMMETRIC PRECONDITIONERS FOR SADDLE POINT PROBLEMS*

MIKE A. BOTCHEV[†] AND GENE H. GOLUB[‡]

Dedicated to Henk A. van der Vorst, who has made so many seminal contributions and who has been so supportive to his colleagues and our community, on the occasion of his 60th birthday

Abstract. For the iterative solution of saddle point problems, a nonsymmetric preconditioner is studied which, with respect to the upper-left block of the system matrix, can be seen as a variant of SSOR. An idealized situation where SSOR is taken with respect to the skew-symmetric part plus the diagonal part of the upper-left block is analyzed in detail. Since action of the preconditioner involves solution of a Schur complement system, an inexact form of the preconditioner can be of interest. This results in an inner-outer iterative process. Numerical experiments with solution of linearized Navier–Stokes equations demonstrate the efficiency of the new preconditioner, especially when the upper-left block is far from symmetric.

Key words. saddle point problems, iterative methods, preconditioning methods, nonsymmetric indefinite linear systems, SSOR, constraint preconditioners, skew-symmetric preconditioners, inner-outer iterations, Navier–Stokes equations

AMS subject classifications. 65F10, 65F22, 65F35, 65N22, 65K10

DOI. 10.1137/040618680

1. Introduction. We consider a nonsymmetric preconditioner for the iterative solution of the linear system

$$(1.1) \quad \mathcal{A}u = \begin{bmatrix} A & B^T \\ B & -C \end{bmatrix} \begin{bmatrix} x \\ y \end{bmatrix} = \begin{bmatrix} f \\ g \end{bmatrix} = b,$$

where $A \neq A^T \in \mathbb{R}^{n \times n}$, $B \in \mathbb{R}^{m \times n}$, $C = C^T \in \mathbb{R}^{m \times m}$, and $m < n$ (often $m \ll n$). We assume that the matrix $A + A^T$ is positive definite (i.e., A is a *positive real* matrix) and C is positive semidefinite.

Linear systems of the form (1.1) arise in a number of applications including mixed finite element solution of the Navier–Stokes and the Maxwell equations and constraint optimization [25, 10, 42, 43]. In many cases A , B , and C are large sparse matrices and iterative techniques are preferable for solving (1.1), especially in connection with the discretization of partial differential equations in three dimensions. Since $\mathcal{A}$ is indefinite and often ill-conditioned, preconditioning is in most cases indispensable for iterative solution of (1.1).

Let H and S be symmetric (Hermitian) and skew-symmetric (skew-Hermitian)

*Received by the editors November 9, 2004, accepted for publication (in revised form) by E. Ng January 18, 2006; published electronically March 17, 2006.

<http://www.siam.org/journals/simax/27-4/61868.html>

[†]Department of Applied Mathematics, University of Twente, P.O. Box 217, 7500 AE Enschede, the Netherlands (mbotchev@na-net.ornl.gov). This author's research was supported in part by the Dutch government through the national program BSIK: knowledge and research capacity, in the ICT project BRICKS (<http://www.bsic-bricks.nl>), theme MSV1.

[‡]Scientific Computing and Computational Mathematics Program, Stanford University, Stanford, CA 94305-9025 (golub@scm.stanford.edu). The work of this author was supported in part by the Department of Energy of the United States Government.

parts of A , respectively:

$$H \equiv \frac{1}{2}(A + A^T), \quad S \equiv \frac{1}{2}(A - A^T).$$

To solve (1.1), Golub and Wathen [30] considered a basic iteration of the form

$$(1.2) \quad \mathcal{P}u^{k+1} = (\mathcal{P} - \mathcal{A})u^k + b$$

with symmetric ($P = P^T$) indefinite preconditioner

$$(1.3) \quad \mathcal{P} = \begin{bmatrix} P & B^T \\ B & -C \end{bmatrix}.$$

When A is not far from a symmetric matrix (i.e., $\|S\|/\|H\|$ is a small number), an efficient preconditioner can be obtained by taking P to be the symmetric part of A [30]. In the context of the systems stemming from the Navier–Stokes equations, $\mathcal{P}$ corresponds to a discretization of the Stokes operator and, to compute $\mathcal{P}^{-1}u$ for a given vector u , a number of robust direct and iterative techniques exists. Other choices of $P = P^T$ can also be useful [30].

However, as can be expected, performance of these symmetric preconditioners deteriorates when A is essentially nonsymmetric ($\|S\|/\|H\| \approx 1$ or larger). A need for a good nonsymmetric preconditioner and, in particular, a possibility to extend the approach of [30] to a nonsymmetric case motivated our research.

A variety of preconditioning methods to solve (1.1) iteratively have been significantly extended within the last decade [5]. Following one possible classification [27, 4, 5], we mention block and approximate Schur complement preconditioners [19, 20, 16, 21, 17, 44, 45, 37, 38], preconditioners based on the Uzawa algorithm [26, 18, 8, 9, 59], preconditioners stemming from the classical splitting iterative schemes (where our approach may fall into) [14, 30, 31, 4], preconditioners inspired by analysis of the underlying continuous partial differential operators [35], sparse direct and approximate factorization preconditioners [13, 24, 46], the so-called null-space preconditioners [48, 1, 32], multigrid preconditioners [51], and other approaches. A few of these approaches work well in the nonsymmetric case, among them [19, 16, 44, 35, 4]. Preconditioners for systems (1.1) stemming from the Navier–Stokes equations are the subject of a recent book [22].

We note that preconditioners of the form (1.3) are studied in [30, 36, 32, 11, 12, 49] and sometimes called constraint preconditioners.

For simplicity, without loss of generality, here and throughout the paper we assume that A has ones on its main diagonal, i.e.,

$$\text{Diag}(A) = I,$$

with $\text{Diag}(A)$ and I being, respectively, the diagonal part of matrix A and the identity matrix. This usually can be achieved by a diagonal prescaling. In certain applications, however, one may want to avoid diagonal prescaling due to ill-conditioning of A .

In this paper we consider a nonsymmetric preconditioner which, with respect to A , can be seen as a variant of SSOR. Namely, we take P in (1.3) as

$$(1.4) \quad P := P_{\text{ssor}} \equiv \frac{1}{\omega}(I + \omega L)(I + \omega U), \quad L + U = A - I,$$

where L and U are strictly lower and upper triangular parts of A , respectively. We are not able to provide any rigorous analysis for preconditioner (1.4) and analyze an idealized situation where we take

$$(1.5) \quad P := P_{\text{skew}} \equiv \frac{1}{\omega}(I + \omega L_S)(I + \omega U_S), \quad L_S + U_S = S;$$

i.e., L_S and U_S are, respectively, lower and upper triangular parts of the skew-symmetric part of A . Evidently, when the skew-symmetric part S is large compared to the symmetric part H , P_{skew} appears to be a good approximation to P_{ssor} . Preconditioner (1.5) is thus relevant for understanding the behavior of preconditioner (1.4) when A is strongly nonsymmetric. In our analysis we use a technique similar to [39, 40, 6].

Preconditioners (1.4) and (1.5) coincide when A is a sum of the identity and a skew-symmetric matrix. The idea to consider this special situation to gain understanding in the iterative solution of nonsymmetric problems is not new [28].

Since in (1.3) the preconditioner $\mathcal{P}$ is not a product of (block) triangular matrices, an important question is how to implement action of the preconditioner, i.e., how to find $\mathcal{P}^{-1}u$ when u is given. Unlike for the symmetric Stokes preconditioner, for the preconditioners (1.4), (1.5) there are no standard solvers available. However, the system with the matrix P in our case can easily be solved and, following a straightforward approach, to compute $\mathcal{P}^{-1}u$ one needs to solve a system with the matrix $BP^{-1}B^T + C$ (the negative of the Schur complement matrix). This is an $m \times m$ matrix and in many cases, especially when $m \ll n$, solution by a direct solver would be feasible. An alternative is to apply an inexact form of the preconditioner where, for example, GMRES iteration [47] can be applied to solve the system with $BP^{-1}B^T + C$. We analyze this inexact form of our preconditioning method. Furthermore, our (limited) experience shows that this inexact preconditioner works well, leading to only a moderate increase in the number of the outer iterations as compared to the exact form.

As our numerical experiments suggest, the SSOR preconditioner (1.4) compares favorably to other preconditioning techniques, for a wide range of $\|S\|/\|H\|$, i.e., for matrices close to symmetric as well as for strongly nonsymmetric matrices.

The paper is organized as follows. In section 2.1 analysis for the “idealized” skew preconditioner (1.3), (1.5) is given. First, we obtain general conditions to have convergence in iteration (1.2). (Convergence means that matrix $\mathcal{P}^{-1}\mathcal{A}$ has its eigenvalues on the complex plane inside the unit circle centered at the point $1 + 0i$ ($i^2 = -1$).) Then in section 2.2 we provide bounds for ω that guarantee convergence and further discuss a possible way to optimize convergence by a suitable choice of ω . However, this optimization is usually not efficient in practice since it is based on an estimate which is not sharp. Therefore, in section 2.3 we discuss simple ways to choose ω which work well in practice. This and subsequent sections deal both with the preconditioners (1.3), (1.4) and (1.3), (1.5). A simple model problem for which the eigenvalues of $\mathcal{P}^{-1}\mathcal{A}$ can be computed analytically gives an insight into the effect of the preconditioners in section 2.4. Inexact form of the preconditioners, where at each “outer” iteration the system with $BP^{-1}B^T + C$ is solved by an “inner” iterative process, is studied in section 2.5. Section 2.6 addresses implementation issues and provides estimates of the computational costs. In section 3 we present results of numerical tests. Finally, we make conclusions and give an outlook to future research in the last section.

2. Analysis of the preconditioner.

2.1. Convergence. In this section we analyze the skew preconditioner (1.3), (1.5). Throughout this and the next subsection it is assumed that $P = P_{\text{skew}}$. Following [30], we first rewrite the iteration matrix

$$(2.1) \quad \mathcal{G} \equiv \mathcal{P}^{-1}(\mathcal{P} - \mathcal{A})$$

of the scheme (1.2) as

$$(2.2) \quad \mathcal{G} = \begin{bmatrix} X(P - A) & 0 \\ Y(P - A) & 0 \end{bmatrix},$$

$$(2.3) \quad \begin{aligned} \mathbb{R}^{n \times n} \ni X &= P^{-1} - P^{-1}B^T(BP^{-1}B^T + C)^{-1}BP^{-1}, \\ \mathbb{R}^{m \times n} \ni Y &= (BP^{-1}B^T + C)^{-1}BP^{-1}. \end{aligned}$$

Assume now that P is a positive real matrix. As we will see, for the skew preconditioner (1.5) this will be guaranteed by choosing parameter ω . To get a sufficient condition for convergence, we estimate the spectral radius of $\mathcal{G}$ as

$$(2.4) \quad \rho(\mathcal{G}) = \rho(X(P - A)) \leq \|X(P - A)\|_*,$$

It is convenient here to define the norm $\|\cdot\|_*$ as the Euclidean matrix norm with respect to the symmetric part of P :

$$\|X(P - A)\|_* = \left\| P_H^{1/2} X(P - A) P_H^{-1/2} \right\|_2, \quad P_H \equiv \frac{1}{2}(P + P^T).$$

With this choice of the norm, (2.4) leads to

$$(2.5) \quad \rho(\mathcal{G}) \leq \|X(P - A)\|_* \leq \left\| P_H^{1/2} X P_H^{1/2} \right\|_2 \left\| P_H^{-1/2} (P - A) P_H^{-1/2} \right\|_2,$$

where $P - A$ and, hence, $P_H^{-1/2}(P - A)P_H^{-1/2}$ are symmetric. This is achieved by choosing P in such a way that its skew-symmetric part is that of A . (Nonsymmetric preconditioners with the property $P - P^T = A - A^T$ were introduced in [39].) The following result holds.

LEMMA 2.1. *Let X be defined by (2.3), with positive semidefinite matrix C , and P_H (the symmetric part of P) be positive definite. Then*

$$(2.6) \quad \left\| P_H^{1/2} X P_H^{1/2} \right\|_2 \leq 1,$$

so that (2.5) leads to

$$(2.7) \quad \rho(\mathcal{G}) \leq \left\| P_H^{-1/2} (P - A) P_H^{-1/2} \right\|_2.$$

Proof. We first consider the case when C is positive definite and rewrite matrix X in the form

$$\begin{aligned} X &= P^{-1} [P - B^T(BP^{-1}B^T + C)^{-1}B] P^{-1} \\ &= P^{-1} - P^{-1}B^T C^{-1}(BP^{-1}B^T C^{-1} + I)^{-1}BP^{-1}. \end{aligned}$$

The Sherman–Morrison–Woodbury formula [29] reduces the last expression to

$$X = (P + B^T C^{-1} B)^{-1},$$

so that, with $P_S \equiv \frac{1}{2}(P - P^T)$ being the skew-symmetric part of P ,

$$(2.8) \quad \begin{aligned} & P_H^{1/2} X P_H^{1/2} \\ &= \left(I + \underline{P_H^{-1/2} P_S P_H^{-1/2} + (B P_H^{-1/2})^T C^{-1} (B P_H^{-1/2})} \right)^{-1} = (I + \underline{M})^{-1}, \end{aligned}$$

where the underlined expression is denoted by M . Since C^{-1} is positive definite, the matrix $(B P_H^{-1/2})^T C^{-1} (B P_H^{-1/2})$ is positive semidefinite. Furthermore, $P_H^{-1/2} P_S P_H^{-1/2}$ is skew-symmetric and, therefore, the matrix M is nonnegative real; i.e., $(Mx, x) \geq 0$ for any $x \in \mathbb{R}^n$. Hence, $\|(I + M)^{-1}\|_2 \leq 1$, and the statement is proven.

In the case when C has zero eigenvalues, we write

$$C = U \Lambda U^T,$$

where columns of U are orthonormal eigenvectors of C and $\Lambda = \text{Diag}(\lambda_i^{(c)})$, with $\lambda_i^{(c)}$ being the eigenvalues of C . We introduce, for $\epsilon \geq 0$, matrices

$$\Lambda_\epsilon = \Lambda + \epsilon I, \quad X_\epsilon \equiv P^{-1} - P^{-1} B^T (B P^{-1} B^T + U \Lambda_\epsilon U^T)^{-1} B P^{-1}.$$

Denoting

$$\tilde{B} \equiv U^T B P_H^{-1/2}, \quad \tilde{P}_S \equiv P_H^{-1/2} P_S P_H^{-1/2}, \quad \tilde{\Lambda}_\epsilon^{-1} = \text{Diag} \left(\frac{\epsilon}{\lambda_i^{(c)} + \epsilon} \right) \quad (\epsilon > 0)$$

and using (2.8), we get

$$\begin{aligned} P_H^{1/2} X_\epsilon P_H^{1/2} &= \left(I + \tilde{P}_S + \frac{1}{\epsilon} \tilde{B}^T \tilde{\Lambda}_\epsilon^{-1} \tilde{B} \right)^{-1} \quad (\epsilon > 0), \\ \|P_H^{1/2} X_\epsilon P_H^{1/2}\|_2^2 &= \max_{\|x\|=1} \frac{1}{((I + \tilde{P}_S + \frac{1}{\epsilon} \tilde{B}^T \tilde{\Lambda}_\epsilon^{-1} \tilde{B})x, (I + \tilde{P}_S + \frac{1}{\epsilon} \tilde{B}^T \tilde{\Lambda}_\epsilon^{-1} \tilde{B})x)} \\ &\leq \frac{1}{1 + 2 \min_{\|x\|=1} (\tilde{P}_S x, x) + \frac{1}{\epsilon} \cdot \min_{\|x\|=1} \left[(\tilde{B}^T \tilde{\Lambda}_\epsilon^{-1} \tilde{B} x, x) + \frac{1}{\epsilon} \|\tilde{B}^T \tilde{\Lambda}_\epsilon^{-1} \tilde{B} x\|^2 \right]} \\ &= \frac{1}{1 + 2 \min_{\|x\|=1} (\tilde{P}_S x, x) + \frac{1}{\epsilon} \cdot 0} \leq 1, \end{aligned}$$

because the matrix $\tilde{B}^T \tilde{\Lambda}_\epsilon^{-1} \tilde{B}$ has at least $n - m$ zero eigenvalues. We have

$$\|P_H^{1/2} X_\epsilon P_H^{1/2}\|_2^2 \leq \frac{1}{1 + 2 \min_{\|x\|=1} (\tilde{P}_S x, x)} \leq 1.$$

Letting $\epsilon \rightarrow 0$ in the last estimate yields

$$\|P_H^{1/2} X P_H^{1/2}\|_2 = \lim_{\epsilon \rightarrow 0} \|P_H^{1/2} X_\epsilon P_H^{1/2}\|_2 \leq 1. \quad \square$$

We further proceed similarly to [39, 40, 6] and estimate the norm in (2.7). Recall that the matrix $P - A$ is symmetric, so that $P - A = P_H - H$ and

$$(2.9) \quad \left\| P_H^{-1/2} (P - A) P_H^{-1/2} \right\|_2 = \left\| I - P_H^{-1/2} H P_H^{-1/2} \right\|_2 = \rho(I - P_H^{-1/2} H P_H^{-1/2}).$$

Representing the eigenvalues of $I - P_H^{-1/2} H P_H^{-1/2}$ as Rayleigh quotients, it is easy to see that they are inside the interval $(-1, 1)$ if and only if

$$(2.10) \quad (0 <) \quad (Hx, x) < 2(P_H x, x) \quad \forall x \in \mathbb{R}^n, \quad x \neq 0.$$

We summarize this section with the following result.

THEOREM 2.2. *Let $\mathcal{G}$ be the iteration matrix of (1.2), (1.3), where P is a positive real matrix and $P - P^T = A - A^T$. Then we have that (see Lemma 2.1 and (2.7))*

$$\rho(\mathcal{G}) \leq \left\| P_H^{-1/2} (P - A) P_H^{-1/2} \right\|_2,$$

where P_H is the symmetric part of P . Moreover,

$$\left\| P_H^{-1/2} (P - A) P_H^{-1/2} \right\|_2 < 1$$

if and only if inequality (2.10) holds true.

2.2. Choice of ω . In this section we adopt, with some minor changes, the results from [39, 40, 6] on how the parameter ω should be chosen. So far we have not used the particular form (1.5) of P . It is easy to check that the symmetric part of P is given by

$$(2.11) \quad P_H = \frac{1}{\omega} I + \omega L_S U_S = \frac{1}{\omega} I - \omega L_S L_S^T.$$

Here and elsewhere in this paper we assume that $\omega > 0$. The following obvious lemma follows.

LEMMA 2.3. *The extremal eigenvalues of the symmetric part P_H of preconditioner (1.5) are given by*

$$(2.12) \quad \begin{aligned} \lambda_{\min}(P_H) &= \min_{\|x\|=1} (P_H x, x) = \frac{1}{\omega} - \omega \|L_S\|_2^2, \\ \lambda_{\max}(P_H) &= \max_{\|x\|=1} (P_H x, x) = \frac{1}{\omega}. \end{aligned}$$

Thus, P is positive real, i.e., $\lambda_{\min}(P_H) > 0$, if and only if

$$(2.13) \quad \omega < \frac{1}{\|L_S\|_2}.$$

Taking into account (2.10), we could get conditions on ω (cf. [39, 40, 6]) which would be sufficient for convergence of the iteration (1.2), (1.3), (1.5), provided that (estimates for) the extremum eigenvalues of H and S are known. A similar technique for the classical SSOR, also based on the extremum eigenvalue estimates, has been used in [58, 2], where in particular the norm $\|L_S\|_2$ appears to be an important parameter, too. Indeed, let

$$\lambda_{\min} = \min_{\|x\|_2=1} (Hx, x) = \gamma_1, \quad \lambda_{\max} = \max_{\|x\|_2=1} (Hx, x) = \gamma_2, \quad \|L_S\|_2 = \gamma_3.$$

Note that $\rho(S) < 2\gamma_3$. Then requiring that the maximum of the left-hand side in (2.10) is smaller than the minimum of the right-hand side yields the following condition:

$$(2.10') \quad \gamma_2 < 2 \left(\frac{1}{\omega} - \gamma_3^2 \omega \right),$$

which is sufficient for (2.10) and can easily be solved in ω . Moreover, using the value of γ_1 , by a simple field-of-value technique one can minimize an upper bound for the norm in the right-hand side of (2.7) with respect to ω [39, 40, 6]:

$$\begin{aligned}
 \rho(\mathcal{G}) &\leq \|I - P_H^{-1/2} H P_H^{-1/2}\|_2 \\
 (2.14) \quad &= \max \left\{ |1 - \lambda_{\min}(P_H^{-1/2} H P_H^{-1/2})|, |1 - \lambda_{\max}(P_H^{-1/2} H P_H^{-1/2})| \right\} \\
 &\leq \max \left\{ |1 - \gamma_1 \omega|, \left| 1 - \frac{\gamma_2 \omega}{1 - \gamma_3^2 \omega^2} \right| \right\} \rightarrow \min_{\omega}.
 \end{aligned}$$

It is not difficult to see that the minimum is attained when ω satisfies

$$2 - (\gamma_1 + \gamma_2)\omega - 2\gamma_3^2\omega^2 + \gamma_1\gamma_3^2\omega^3 = 0,$$

and it is the only real root of this polynomial in the interval $0 < \omega < (\sqrt{\gamma_2^2 + 16\gamma_3^2} - \gamma_2)/(4\gamma_3^2)$. However, this “optimal” value of ω is typically useless in practice since it essentially optimizes an upper bound of $\rho(\mathcal{G})$ which is not sharp. (Sharpness is lost in (2.10') and (2.14).)

2.3. Choice of ω in practice. There are simpler ways to choose ω that usually work well in practice. Let

$$(2.15) \quad \omega_* = 1 / \max\{\|L_S\|_{\infty}, \|U_S\|_{\infty}\}.$$

It is easy to check that for $\omega < \omega_*$ the symmetric part P_H of the matrix P has diagonal dominance, and hence P is positive real. This is necessary for the iteration (1.2), (1.3), (1.5) to converge (Theorem 2.2). Another, slightly sharper computable bound on ω (under which P is positive real) is

$$\omega < \left[\sqrt{\|L_S\|_{\infty} \|U_S\|_{\infty}} \right]^{-1} = \left[\sqrt{\|L_S\|_{\infty} \|L_S\|_1} \right]^{-1}.$$

Indeed, this last condition implies (2.13) because $\|L_S\|_2 \leq \sqrt{\|L_S\|_{\infty} \|L_S\|_1}$.

As numerical experiments suggest, fastest in terms of iteration number convergence is typically observed for values of ω usually ≈ 10 % larger than ω_* . This conclusion is made in [6] for linear systems $Ax = f$ stemming from convection-diffusion problems solved by the Richardson and the GMRES methods preconditioned with P from (1.5). For our problem (1.1) we observe the same dependence of convergence on ω for the GMRES method preconditioned by $\mathcal{P}$ with both the SSOR and the skew blocks P (cf. (1.3), (1.4), (1.5)). The typically observed dependence of the number of iterations (to achieve certain residual reduction) on ω is plotted in Figure 2.1.

Figure 2.2 shows how the choice of ω usually influences the eigenvalues of the preconditioned matrix $\mathcal{P}^{-1}\mathcal{A}$. Taking ω slightly larger than ω_* typically leads to a condensation of eigenvalues around point $1 + 0i$ in the complex plane and to a separation of several larger eigenvalues on the real axis. Since clustering is generally beneficial for convergence [55, 56], it is natural to expect a faster convergence for this case. Yet further increasing ω results in the eigenvalues with a negative real part and poor convergence.

We emphasize that the preconditioners (1.3), (1.4) and (1.3), (1.5) usually exhibit very robust convergence behavior with respect to perturbations in ω , especially when applied in combination with a modern Krylov subspace method. Taking ω in the neighborhood of ω_* is normally a very good choice in practice. In the numerical

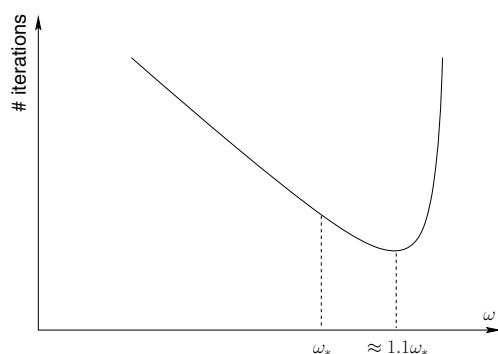


FIG. 2.1. Typical dependence on ω of the iteration number to achieve a certain residual norm reduction for the skew- and SSOR-preconditioned iterative methods (1.2) observed when solving the test problems described in section 3. Similar dependence is observed in [6] for central finite difference discretizations of convection-diffusion problems.

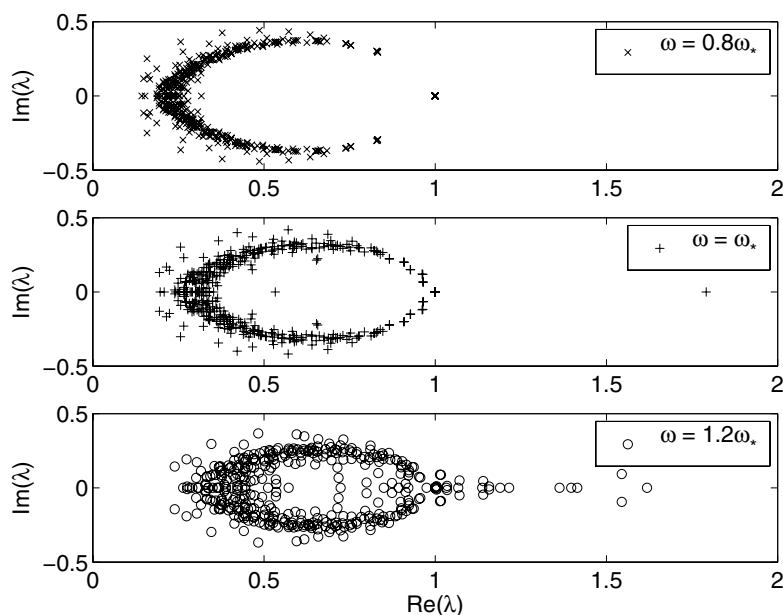


FIG. 2.2. Eigenvalues of the skew-preconditioned matrix $P^{-1}A$ in the complex plane for $\omega = 0.8\omega_*$ (top), $\omega = \omega_*$ (middle), $\omega = 1.2\omega_*$ (bottom); ω_* is given by (2.15). Eigenvalue clustering and faster convergence are observed for $\omega > \omega_*$. A is a diagonally scaled discretized Navier-Stokes operator obtained by the stable $(Q_1\text{-iso-}Q_2)\text{-}P_0$ discretization (see section 3) on a 16×16 mesh ($n = 578$, $m = 64$), viscosity $\nu = 0.01$.

experiments presented in section 3 with the preconditioners (1.3), (1.4) and (1.3), (1.5), ω was chosen as

$$(2.16) \quad \omega := \begin{cases} [0.9 \max\{\|L_S\|_\infty, \|U_S\|_\infty, 1.0\}]^{-1} & \text{if } P = P_{\text{skew}}, \\ [0.9 \max\{\|L\|_\infty, \|U\|_\infty, 1.0\}]^{-1} & \text{if } P = P_{\text{ssor}}, \end{cases}$$

where care is taken that ω does not get too large.

2.4. Preconditioning for a model problem. Here we inspect the preconditioning effect for a simple situation where the eigenvalues of the preconditioned matrix $\mathcal{P}^{-1}\mathcal{A}$ (cf. (1.3), (1.4)) can be computed analytically. More specifically, we make the following assumption.

ASSUMPTION 1. $\mathcal{A}$ is similar to a block-diagonal matrix with 3×3 diagonal blocks

$$(2.17) \quad \begin{bmatrix} 1 & \sigma_i & \hat{b}_{i,1} \\ -\sigma_i & 1 & \hat{b}_{i,2} \\ \hat{b}_{i,1} & \hat{b}_{i,2} & -c_i \end{bmatrix}.$$

When applied to matrix $\mathcal{A}$ in the transformed block-diagonal form with blocks (2.17), preconditioning (1.3) results in matrix $\mathcal{P}^{-1}\mathcal{A}$ of the same block-diagonal structure, so that the effect of the preconditioning can be traced for each of the blocks separately. Note that for block-diagonal matrices with blocks (2.17) the SSOR and skew preconditioners coincide (cf. (1.4) and (1.5)).

The following lemma shows that Assumption 1 holds true for a class of matrices $\mathcal{A}$ if B has a special sparsity structure and C is diagonal.

LEMMA 2.4. *Let*

$$n = 2m, \quad A = \begin{bmatrix} I & G^T \\ -G & I \end{bmatrix}, \quad R = \begin{bmatrix} V & 0 \\ 0 & U \end{bmatrix}, \quad G, U, V \in \mathbb{R}^{m \times m},$$

where the orthogonal matrices U and V define the singular value decomposition of $G = U\Sigma V^T$. Then there exists a permutation matrix $\mathbf{P}$ such that the matrix $(\mathbf{R}\mathbf{P})^T \mathbf{A} \mathbf{R}\mathbf{P}$ is block-diagonal with 2×2 diagonal blocks

$$(2.18) \quad \begin{bmatrix} 1 & \sigma_i \\ -\sigma_i & 1 \end{bmatrix},$$

with σ_i being the singular values of G . Moreover, Assumption 1 holds true if, in addition, C is diagonal and B is such that the matrix $\mathbf{B}\mathbf{R}\mathbf{P}$ has nonzero entries only at the positions $(1, 1)$, $(1, 2)$, $(2, 3)$, $(2, 4)$, $\dots$, $(m, 2m-1)$, $(m, 2m)$.

Proof. The proof is straightforward and shows how, under the assumptions of the lemma, reduction of $\mathcal{A}$ to the block-diagonal form with blocks (2.17) can be made. First, we note that

$$\mathcal{R}^T \mathcal{A} \mathcal{R} = \begin{bmatrix} R^T A R & (BR)^T \\ BR & -C \end{bmatrix} \quad \text{with} \quad \mathcal{R} = \begin{bmatrix} R & 0 \\ 0 & I \end{bmatrix}, \quad R^T A R = \begin{bmatrix} I & \Sigma \\ -\Sigma & I \end{bmatrix}.$$

It is not difficult to see that a permutation matrix $\mathbf{P}$ exists such that $\mathbf{P}(R^T A R)\mathbf{P} = (\mathbf{R}\mathbf{P})^T \mathbf{A} \mathbf{R}\mathbf{P}$ is block-diagonal with diagonal blocks (2.18). Then

$$(\mathbf{R}\mathbf{P}_1)^T \mathcal{A} \mathbf{R}\mathbf{P}_1 = \begin{bmatrix} (\mathbf{R}\mathbf{P})^T \mathbf{A} \mathbf{R}\mathbf{P} & (\mathbf{B}\mathbf{R}\mathbf{P})^T \\ \mathbf{B}\mathbf{R}\mathbf{P} & -C \end{bmatrix}, \quad \mathbf{P}_1 = \begin{bmatrix} \mathbf{P} & 0 \\ 0 & I \end{bmatrix}.$$

Define now another permutation matrix $\mathbf{P}_2 \in \mathbb{R}^{(n+m) \times (n+m)}$ with columns being those of the identity matrix written in the order

$$1, 2, n+1, 3, 4, n+2, \dots, n-1, n, n+m.$$

Matrix $(\mathbf{R}\mathbf{P}_1\mathbf{P}_2)^T \mathcal{A} \mathbf{R}\mathbf{P}_1\mathbf{P}_2$ has the required block-diagonal structure with diagonal blocks (2.17). $\square$

From (2.2), we see that m eigenvalues of $\mathcal{P}^{-1}\mathcal{A}$ are equal to one and the other n are of the form $1 - \lambda_i$, with λ_i being the eigenvalues of the matrix $X(P - A)$. For the blocks (2.17), computations in Maple show that (we omit the subindices i in (2.17))

$$(2.19) \quad X(P - A) = \frac{1}{\hat{b}_1^2 \sigma^2 \omega^3 - (\hat{b}_1^2 + \hat{b}_2^2) \omega - c} \\ \times \begin{bmatrix} (\omega - 1)(-\hat{b}_2^2 \omega + c(\sigma^2 \omega^2 - 1)) & -\omega(\sigma^2 \omega^2 - 1 + \omega)(\sigma c + \hat{b}_1 \hat{b}_2) \\ \omega(\omega - 1)\omega(-\hat{b}_1 \hat{b}_2 + \sigma c) & (\sigma^2 \omega^2 - 1 + \omega)(\hat{b}_1^2 \omega + c) \end{bmatrix}.$$

Without loss of generality we assume that $\sigma > 0$. Since we are mainly interested in the situations for which the skew-symmetric component S is large in norm, we can expect values of σ to be relatively large too; in fact, they are proportional to a norm of S .

For simplicity we consider the case $\hat{b}_1 = \hat{b}_2 = \hat{b}$. It is reasonable to choose σ as a characteristic scale and express ω , $\hat{b}$, and c in terms of σ as

$$\omega = \alpha \left[\frac{1}{\sigma} \right], \quad \hat{b} = \beta \sigma, \quad c = \gamma \sigma,$$

where $\alpha = 1$ corresponds to the important choice $\omega = \omega_* \equiv 1/\sigma$ (cf. (2.15)). Note that $\alpha > 0$, $\gamma \geq 0$.

In the case $c = \gamma = 0$, the eigenvalues of (2.19) take an elegant form

$$(2.20) \quad \lambda_1 = 1 + \frac{2\alpha}{\sigma(\alpha^2 - 2)}, \quad \lambda_2 = 0,$$

delivering, for the following two interesting choices of α ,

$$(2.21) \quad \lambda_1 = 1 - \frac{2}{\sigma} \quad \text{if } \alpha = 1 \ (\omega = \omega_*), \\ \lambda_1 = 1 + \frac{2}{\sigma^2 - 2} \quad \text{if } \alpha = \sigma \ (\omega = 1).$$

Note that $\hat{b}$ has no influence on λ_1 at all. Requirement $|\lambda_1| < 1$ is equivalent to

$$0 < \alpha < \bar{\alpha} \equiv \sqrt{2 + \left(\frac{1}{2\sigma} \right)^2} - \frac{1}{2\sigma},$$

where $\bar{\alpha}$ increases monotonically with σ , and $\lim_{\sigma \rightarrow 0+} \bar{\alpha} = 0$, $\lim_{\sigma \rightarrow \infty} \bar{\alpha} = \sqrt{2}$. Figure 2.3 shows the dependence of $\rho(X(P - A)) = |\lambda_1|$ on α and ω . Here we recognize the familiar dependence of convergence rate on ω (cf. Figure 2.1). The right plot in Figure 2.3 shows the influence of different blocks (2.17) on the spectral radius: choosing the block with the largest $\sigma = \sigma_{\max}$ and setting $\omega := \omega_* \equiv 1/\sigma_{\max}$ will result for the other blocks in

$$(2.22) \quad \alpha = \frac{\sigma}{\sigma_{\max}} \Rightarrow \rho(X(P - A)) = |\lambda_1| = \left| 1 + \frac{2\sigma/\sigma_{\max}}{\sigma((\sigma/\sigma_{\max})^2 - 2)} \right| < 1.$$

For further analysis, where we inspect the effect of the β and γ on the eigenvalues of $\mathcal{P}^{-1}\mathcal{A}$, see the appendix.

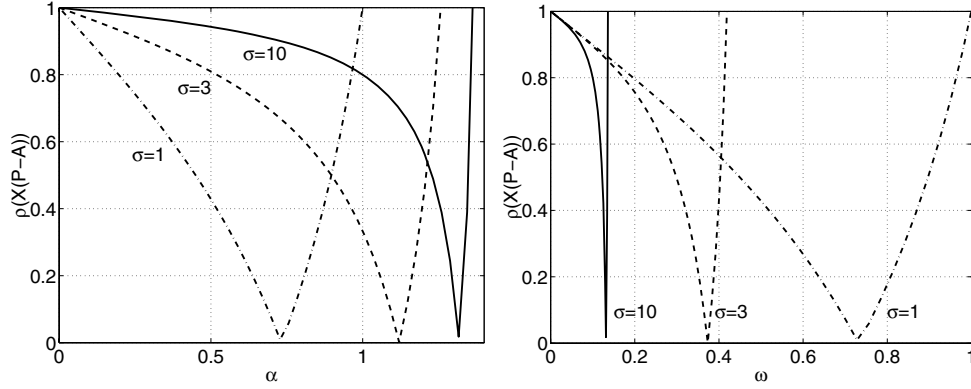


FIG. 2.3. Model problem $c = \gamma = 0$. Left plot: Analytically computed $\rho(X(P-A))$ versus $\alpha = \omega\sigma$ for different values of σ . If σ is large enough, then the fastest convergence ($\rho(X(P-A)) \approx 0$) takes place for $\alpha > 1$ (for $\omega > \omega_*$). Right plot: the same values against ω . Here $\omega_* \equiv 1/\sigma_{\max} = 0.1$.

2.5. Inexact preconditioning. In this section we analyze an inexact form of the preconditioned basic iteration (1.2), (1.3). To compute $v := \mathcal{P}^{-1}u$ (note that we do not compute the matrix $\mathcal{P}^{-1}$ explicitly), an $m \times m$ linear system of the form

$$(2.23) \quad (BP^{-1}B^T + C)y_2 = y_1$$

has to be solved for a given vector y_1 . The inexact method we consider is of interest when solution of (2.23) by a direct method is not feasible and an iterative method is used. This leads to an inner-outer iterative procedure. A proper stopping criterion for the inner iteration should be chosen for which, on the one hand, the outer iteration convergence is not corrupted too much and, on the other hand, not too many inner iterations are done. The inner-outer iterations have been studied in the context of different problems (see, e.g., [28, 18, 23, 3, 57, 52] and the references therein). In particular, it is known that if the residual norm tolerance used in the inner iteration converges to zero, then usually the convergence rate of the exact method is asymptotically recovered [28, 23, 3]. We will show that this is true for the inexact iteration (1.2).

We first adopt some of the known results on the inner-outer iteration to the outer iteration (1.2) written as

$$(2.24) \quad u^{k+1} = u^k + \mathcal{P}^{-1}r^k, \quad r^k \equiv b^k - Au^k.$$

The following simple result provides one possible choice of the inner iteration tolerance for which the convergence rate of the exact method is asymptotically recovered (cf. Theorem 3.3 in [3]).

LEMMA 2.5. Assume that iteration (2.24) converges, $\rho(\mathcal{G}) < 1$, $\mathcal{G} = \mathcal{P}^{-1}(\mathcal{P} - A)$, and let $\|\cdot\|_*$ be a norm such that $\|\mathcal{G}\|_* < 1$. Assume that at each step of iteration (2.24) linear system $\mathcal{P}(u^{k+1} - u^k) = r^k$ is solved inexactly, with a residual $p^k \equiv r^k - \mathcal{P}(u^{k+1} - u^k)$, so that the inexact iteration reads

$$(2.25) \quad u^{k+1} = u^k + \mathcal{P}^{-1}r^k - \mathcal{P}^{-1}p^k.$$

Then the inexact iteration (2.24) converges to the exact solution $\hat{u}$ of (1.1) in the

norm $\|\cdot\|_*$, provided that

$$\begin{aligned} \|p^k\|_* &\leq \epsilon_k \|r^k\|_*, & k = 0, 1, \dots, \quad \text{and} \\ \varsigma + \epsilon_{\max} \theta &< 1, & \epsilon_{\max} = \max_k \epsilon_k, \end{aligned}$$

where $\varsigma \equiv \|\mathcal{G}\|_* < 1$ and $\theta = \|\mathcal{P}^{-1}\|_* \cdot \|\mathcal{A}\|_*$. If, furthermore, the inner iteration tolerance ϵ_k satisfies

$$(2.26) \quad \epsilon_k \leq c\delta^{\tau_k},$$

where $c \geq 0$ and $\delta \in (0, 1)$ are constants and $\tau_k \geq 1$ is a nondecreasing sequence such that $\lim_{k \rightarrow \infty} \tau_k = \infty$, then the convergence rate is asymptotically the same as for the exact iteration (2.24):

$$(2.27) \quad \limsup_{k \rightarrow \infty} \frac{\|u^{k+1} - \hat{u}\|_*}{\|u^k - \hat{u}\|_*} \leq \varsigma.$$

Proof. Existence of the norm $\|\cdot\|_*$ follows from the fact that for any $\varepsilon > 0$ there exists at least one matrix norm such that $\|\mathcal{G}\|_* \leq \rho(\mathcal{G}) + \varepsilon$ (see, e.g., Lemma 5.6.10 in [34]). Subtracting the equality $\hat{u} = \mathcal{G}\hat{u} + \mathcal{P}^{-1}b$ from (2.25) we arrive at

$$u^{k+1} - \hat{u} = \mathcal{G}(u^k - \hat{u}) - \mathcal{P}^{-1}p^k,$$

so that

$$\begin{aligned} \|u^{k+1} - \hat{u}\|_* &\leq \|\mathcal{G}\|_* \|u^k - \hat{u}\|_* + \|\mathcal{P}^{-1}\|_* \|p^k\|_* \\ &\leq \|\mathcal{G}\|_* \|u^k - \hat{u}\|_* + \|\mathcal{P}^{-1}\|_* \epsilon_k \|r^k\|_*. \end{aligned}$$

Since $\|r^k\|_* = \|\mathcal{A}\hat{u} - \mathcal{A}u^k\|_* \leq \|\mathcal{A}\|_* \|u^k - \hat{u}\|_*$ we obtain

$$\|u^{k+1} - \hat{u}\|_* \leq (\varsigma + \epsilon_k \theta) \|u^k - \hat{u}\|_*,$$

which shows convergence, provided that $\varsigma + \epsilon_{\max} \theta < 1$. If ϵ_k satisfies (2.26), then

$$\frac{\|u^{k+1} - \hat{u}\|_*}{\|u^k - \hat{u}\|_*} \leq \varsigma + c\delta^{\tau_k} \theta.$$

Letting $k \rightarrow \infty$ leads to (2.27). Note that for both the inexact and the exact iteration the value of the upper limit in (2.27) depends on the initial guess vector u^0 and belongs to the interval $[\rho(\mathcal{G}), \varsigma]$. If $\mathcal{G}$ is nonsingular, then the value of the limit is exactly ς (see [33, Exercise 3.2.12]). $\square$

We now consider a specific form of the inexact preconditioner (1.2), (1.3), where the system (2.23) is solved approximately. Direct computations show that

$$\mathcal{P}^{-1} = \begin{bmatrix} P^{-1} - P^{-1}B^TW^{-1}BP^{-1} & P^{-1}B^TW^{-1} \\ W^{-1}BP^{-1} & -W^{-1} \end{bmatrix}, \quad W \equiv BP^{-1}B^T + C.$$

Let $v = (\mathcal{P} - \mathcal{A})u^k + b$ be partitioned as $v^T = (x^T, y^T)$, with x consisting of the first n components of v . In (1.2), (1.3) we have

$$(2.28) \quad u^{k+1} = \mathcal{P}^{-1}v = \mathcal{P}^{-1} \begin{bmatrix} x \\ y \end{bmatrix} = \begin{bmatrix} P^{-1}x - P^{-1}B^TW^{-1}(BP^{-1}x - y) \\ W^{-1}(BP^{-1}x - y) \end{bmatrix},$$

where to compute the action of W^{-1} the linear system (2.23) with the right-hand side $y_1 = BP^{-1}x - y$ is solved. In our inexact version of (1.2), (1.3) we allow for an approximate solution of this system with residual $q^k \equiv BP^{-1}x - y - Wy_2$ so that

$$y_2 = W^{-1}(BP^{-1}x - y) - W^{-1}q^k \approx W^{-1}(BP^{-1}x - y).$$

In this inexact iteration we essentially work with an approximate preconditioner $\hat{\mathcal{P}}_k \approx \mathcal{P}$:

$$\begin{aligned} (2.29) \quad u^{k+1} &= \hat{\mathcal{P}}_k^{-1} [(\mathcal{P} - \mathcal{A})u^k + b] \\ &= \hat{\mathcal{P}}_k^{-1} \begin{bmatrix} x \\ y \end{bmatrix} = \begin{bmatrix} P^{-1}x - P^{-1}B^T [W^{-1}(BP^{-1}x - y) - W^{-1}q^k] \\ W^{-1}(BP^{-1}x - y) - W^{-1}q^k \end{bmatrix} \\ &= \begin{bmatrix} P^{-1}x - P^{-1}B^TW^{-1}(BP^{-1}x - y) \\ W^{-1}(BP^{-1}x - y) \end{bmatrix} + \begin{bmatrix} P^{-1}B^TW^{-1}q^k \\ -W^{-1}q^k \end{bmatrix} \\ &= \mathcal{P}^{-1}v + \mathcal{P}^{-1} \begin{bmatrix} 0 \\ q^k \end{bmatrix}. \end{aligned}$$

Substituting $v = (\mathcal{P} - \mathcal{A})u^k + b$ into the last expression, we obtain the following formula for the inexact iteration (1.2), (1.3):

$$(2.30) \quad u^{k+1} = u^k + \mathcal{P}^{-1}r^k + \mathcal{P}^{-1} \begin{bmatrix} 0 \\ q^k \end{bmatrix}.$$

Comparing this last expression with the general inexact form of the Richardson method (2.25), we arrive at the following result.

THEOREM 2.6. *Assume that iteration (1.2), (1.3) converges, $\rho(\mathcal{G}) < 1$, $\mathcal{G} = \mathcal{P}^{-1}(\mathcal{P} - \mathcal{A})$, and let $\|\cdot\|_*$ be a norm such that $\|\mathcal{G}\|_* < 1$. Then the inexact form (2.29), (2.30) of iteration (1.2), (1.3), where at step k the system (2.23) is solved approximately with residual q^k , converges to the exact solution $\hat{u}$ of (1.1) in norm $\|\cdot\|_*$, provided that*

$$\begin{aligned} \|q^k\|_* &\leq \epsilon_k \|r^k\|_*, \quad k = 0, 1, \dots, \quad \text{and} \\ \varsigma + \epsilon_{\max} \theta &< 1, \quad \epsilon_{\max} = \max_k \epsilon_k, \end{aligned}$$

where $\varsigma \equiv \|\mathcal{G}\|_*$ and $\theta = \|\mathcal{P}^{-1}\|_* \cdot \|\mathcal{A}\|_*$. If, furthermore, the inner iteration tolerance ϵ_k satisfies (2.26), then the convergence rate of the inexact iteration (2.29), (2.30) is asymptotically the same as for the exact iteration (1.2), (1.3) and relation (2.27) holds.

Proof. Note that (2.30) is a particular case of (2.25) with

$$p^k = - \begin{bmatrix} 0 \\ q^k \end{bmatrix}$$

and apply Lemma 2.5. Since $\mathcal{G}$ is singular (see (2.2)), the actual convergence rate depends on the initial guess u^0 and can be smaller than ς (see [33, Exercise 3.2.12]). $\square$

By choosing tolerance in the inner stopping criterion carefully one could also aim at minimizing the overall computational work in the inner-outer iteration rather than at preserving the outer convergence rate [23, 28]. Adopting these strategies to the inexact SSOR iteration is left for future work. We emphasize that another approach, aiming at maintaining convergence through the outer iteration process, has recently been developed [52, 53, 54]. This approach is valid for a more general situation of inexact matrix-vector products [53] and explains a heuristical inner iteration relaxation strategy reported in [7].

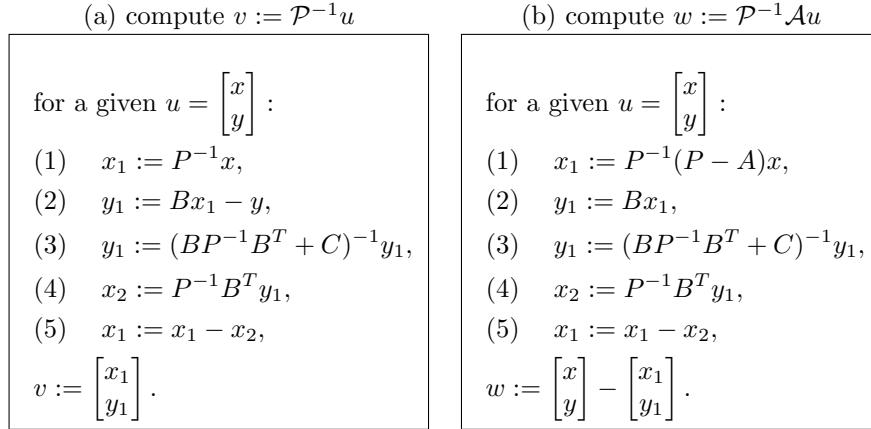


FIG. 2.4. Algorithms for preconditioned matrix-vector products.

2.6. Implementation and costs. The iterative scheme considered in section 2.1 is in fact a stationary Richardson method applied to the left-preconditioned system $\mathcal{P}^{-1}\mathcal{A}u = \mathcal{P}^{-1}b$. When applying preconditioner (1.3) in combination with this or any other iterative method, one needs to repeatedly compute a result of the action of the matrix $\mathcal{P}^{-1}$ on a given vector. In this section we explain how this can be done. In addition, implementation of the matrix-vector multiplication $\mathcal{P}^{-1}\mathcal{A}u$ is considered. As it turns out, it can be organized in such a way that, as compared to computing of $\mathcal{P}^{-1}u$, it requires only a little extra work. We therefore emphasize that in most cases one should not separate steps $v := \mathcal{A}u$, $w := \mathcal{P}^{-1}v$ but rather combine them in $w := \mathcal{P}^{-1}\mathcal{A}u$.

Consider first the matrix-vector multiplication $v := \mathcal{P}^{-1}u$. In view of (2.28), for u partitioned as $u^T = (x^T, y^T)$, $x \in \mathbb{R}^n$, we can write

$$u^{k+1} = \mathcal{P}^{-1}v = \mathcal{P}^{-1} \begin{bmatrix} x \\ y \end{bmatrix} = \begin{bmatrix} P^{-1}x - P^{-1}B^T W^{-1}(BP^{-1}x - y) \\ W^{-1}(BP^{-1}x - y) \end{bmatrix},$$

with $W \equiv BP^{-1}B^T + C$. This leads to the algorithm shown in Figure 2.4(a).

To work out computation of $w := \mathcal{P}^{-1}\mathcal{A}u$, we use (2.1) and (2.2):

$$\mathcal{P}^{-1}\mathcal{A}u = u - \mathcal{G}u = \begin{bmatrix} x \\ y \end{bmatrix} - \begin{bmatrix} X(P - A)x \\ Y(P - A)x \end{bmatrix}.$$

Substituting X and Y from (2.3), we get

$$\begin{aligned} X(P - A)x &= P^{-1}(P - A)x - P^{-1}B^T \underline{Y(P - A)x} \\ &= [I - P^{-1}B^T(BP^{-1}B^T + C)^{-1}B] P^{-1}(P - A)x, \end{aligned}$$

so that, computing $X(P - A)x$, we get $Y(P - A)x$ as a by-product. The resulting procedure to compute $v := \mathcal{P}^{-1}\mathcal{A}u$ is outlined in Figure 2.4(b).

Note that in both algorithms from Figure 2.4 the inverse matrices at steps (1), (3), and (4) do not have to be computed; instead, one solves linear systems. Steps (1) and (4) in both algorithms can be done efficiently since P is a product of triangular matrices. The most expensive part is step (3). One possible way here is to use a direct

TABLE 1

Estimates for the costs of the algorithms for preconditioned matrix-vector products (Figure 2.4). The notation is as follows: L and U have, respectively, at most l and u nonzero entries per row, and B has no more than r nonzero entries in each row or column. The matrix $(BP^{-1}B^T + C)$ is computed and factorized once. It is assumed that operations $x_1 := P^{-1}Ax$ and $x_1 := P^{-1}x$ have roughly the same costs (as is the case when, e.g., Eisenstat's trick is applied [15]). Terms $O(m)$ and $O(n)$ are omitted.

Initialization costs:	
Computation of $(BP^{-1}B^T + C)$	$2n(l + u + m) + 2m^2r$
LU factorization of $(BP^{-1}B^T + C)$	$\frac{2}{3}m^3$
Total:	$\frac{2}{3}m^3 + 2m^2r + 2n(l + u + m)$
Costs per matrix-vector product $v := \mathcal{P}^{-1}u$ or $w := \mathcal{P}^{-1}Au$:	
Step (see Figure 2.4)	Costs
(1)	$2n(l + u)$
(2),(3)	$2m^2 + 2m(r - 1)$
(4)	$2n(l + u + r)$
(5)	n
Total:	$2m^2 + 2n[2(l + u) + r] + 2mr$ (terms $O(m)$ and $O(n)$ are neglected)

linear solver, computing the LU factorization of the $m \times m$ Schur complement once and then reusing it at every step (3). The costs of the preconditioner for this case are given in Table 1. If these costs are not feasible, solution in step (3) can be done iteratively. This leads to the inner-outer iterative procedure analyzed in section 2.5. We further discuss implementation issues for this method in section 3.1.

Compared with other known preconditioners used for systems (1.1), our preconditioning is not too expensive. For example, one matrix-vector product with the BFB^T preconditioning [16] involves solving a linear system with the matrix A . In addition, solving the eigenvalue problem for BB^T (to rearrange the unknowns) may be necessary. In Table 2, the costs of the BFB^T preconditioner are given for the case when the matrix A has bandwidth $\sqrt{n}$, the system with A is solved by a band LU direct solver, and the eigenvalue problem for BB^T is solved. Comparing the costs of the SSOR and BFB^T preconditioners (Tables 1 and 2), we see that application of the preconditioner matrix $v := \mathcal{P}^{-1}u$ for BFB^T is approximately three times more expensive than for SSOR, even if one neglects the costs of the matrix A solved in BFB^T . Indeed, these costs can often be reduced by using an (inner) iterative solver. We emphasize that the costs in BFB^T can be reduced by making it inexact [16], just as can be done for the SSOR preconditioner presented in this paper.

Another, block-triangular preconditioner (see [19] and section 3.3) requires, for the matrix A having bandwidth $\sqrt{n}$, about $2n^2$ operations at its initialization stage for the band LU factorization of A and about $4n\sqrt{n} + 2nr$ operations at each iteration for $w := \mathcal{P}^{-1}v$ (Table 3). Thus, this preconditioner is much cheaper than either BFB^T or SSOR when m is large (cf. Tables 1 and 2).

In section 3.3 we also demonstrate the performance of a simple constraint preconditioner (1.3) with P taken to be the identity matrix, which is the diagonal of A if the diagonal prescaling is applied. The costs of this preconditioner are significantly reduced as compared to general constraint preconditioners (Table 1) because of a simpler structure of the matrix $BP^{-1}B^T + C = BB^T + C$. If this matrix is a sparse band matrix with bandwidth $\sqrt{n}$ (as is the case in section 3.3), the initialization costs are approximately $4mn$ operations and every matrix-vector multiplication requires about $4m\sqrt{n} + 2(m + n)r$ operations (cf. Tables 1 and 2).

TABLE 2

Estimates for the costs of the exact BFB^T preconditioner matrix-vector products. It is assumed that the matrix A has bandwidth $\sqrt{n}$ as is the case for the linearized Navier–Stokes problem of section 3. The notation is the same as in Table 1, and s is the maximal number of nonzero entries per row in C . Terms $O(m)$ and $O(n)$ are omitted.

Initialization costs:	
LU band factorization of A	$\approx 2n\sqrt{n}\sqrt{n} = 2n^2$
Computation of BB^T	$2m^2(2r-1)$
Solving eigenproblem for BB^T	$\frac{8}{3}m^3$
Total:	$\frac{8}{3}m^3 + 2m^2(2r-1) + 2n^2$
Costs per matrix-vector product:	
Operation	Costs
$v := \mathcal{A}u$	$2n(l+u+r) + 2m(r+s-1)$
$w := \mathcal{P}^{-1}v$	$6m^2 + 2n(l+u+2r) + 2m(r-1) + 2m-3 + 4n\sqrt{n}$
Total for $w := \mathcal{P}^{-1}\mathcal{A}u$	$6m^2 + 2n[2(l+u)+3r] + 2m(2r+s) + 4n\sqrt{n}$ (terms $O(m)$ and $O(n)$ are neglected)

TABLE 3

Estimates for the costs of the exact block-triangular preconditioner matrix-vector products. For notation and assumptions see the caption of Table 2.

Initialization costs:	
LU band factorization of A	$\approx 2n\sqrt{n}\sqrt{n} = 2n^2$
Costs per matrix-vector product:	
Operation	Costs
$v := \mathcal{A}u$	$2n(l+u+r) + 2m(r+s-1)$
$w := \mathcal{P}^{-1}v$	$2nr + 4n\sqrt{n}$
Total for $w := \mathcal{P}^{-1}\mathcal{A}u$	$2n(l+u+2r) + 2m(r+s) + 4n\sqrt{n}$ (terms $O(m)$ and $O(n)$ are neglected)

In the symmetric preconditioners [30], one needs to solve a linear system with a symmetric $n \times n$ matrix at each preconditioned matrix vector multiplication; this can often be done with fast direct solvers.

3. Numerical experiments. We have carried out numerical experiments for systems (1.1) coming from the finite element discretization of the two-dimensional linearized Navier–Stokes equations (the Oseen equations; see, e.g., [22, 20, 21, 17]):

$$\begin{cases} -\nu\Delta\mathbf{u} + (\mathbf{v} \cdot \nabla)\mathbf{u} + \nabla p = \mathbf{f}, \\ \nabla \cdot \mathbf{u} = 0, \end{cases}$$

where velocity $\mathbf{u}$ is the unknown, $\mathbf{v}$ is the known velocity from the previous (Picard) iteration, p is pressure, and $\nu > 0$ is viscosity. We have chosen this problem for numerical experiments because it is a widely known and well-understood test problem. We emphasize that there are a lot of powerful preconditioners which work very well for the systems stemming from the Navier–Stokes equations [22, 35, 17, 21]; see also the recent survey [5]. Our aim here is to show that our preconditioning approach based on purely algebraic considerations, though apparently not being the best possible choice for this particular problem, may be competitive with other well-known preconditioners.

The test problem is the leaky-lid driven cavity problem, as generated by the MATLAB software of Silvester and Elman [50], with the wind field $\mathbf{v} = (v_1, v_2)$ chosen as

$$v_1(x, y) = 2y(1 - x^2), \quad v_2(y) = -2x(1 - y^2), \quad -1 \leq x, y \leq 1.$$

The software can produce two types of discretizations: the stable (Q₁-iso-Q₂)-P₀ discretization and the stabilized Q₁-P₀ discretization (in the former case $C = 0$); see, e.g., [20, 51, 35] and the references therein. The stabilization parameter for the stabilized Q₁-P₀ discretization was $\beta = 0.25$.

Throughout this section, ω in preconditioners (1.3), (1.4) and (1.3), (1.5) was chosen according to (2.16). For the SSOR, skew, and Stokes preconditioners, the two-sided diagonal prescaling was used to get $\text{Diag}(A) = I$. We used full GMRES [47] as the (outer) iterative solver. All the runs were done on a PC with a 2.5 GHz processor and 2 Gb memory.

3.1. Implementation of the inexact iteration. Analysis of section 2.5 gives convergence conditions for the Richardson iteration (1.2) when action of the Schur complement inverse is computed approximately by another, inner iterative process. Since the obtained conditions (Theorem 2.6) are based on the norm estimates that are not sharp, we expect them to be too strict in practice. These conditions should also be relaxed when a modern Krylov subspace method, instead of the simplest Richardson method, is employed in the outer iteration. Therefore, we interpret Theorem 2.6 qualitatively rather than quantitatively: the residual norm in the inner iteration should be proportional to the outer iteration residual norm.

In our numerical experiments, we have used the following stopping criterion for the inner iterations:

$$(3.1) \quad \|q^k\|_2 \leq \begin{cases} 10^{-6} & \text{if } \|r^k\|_2 > 0.01, \\ \|r^k\|_2 & \text{otherwise,} \end{cases}$$

where q^k and r^k are inner and outer residuals, respectively. Here the strict tolerance on $\|q^k\|_2$ for large outer residual norm is not caused by the convergence requirements but rather by accuracy requirements of this specific test problem. Without this condition, convergence of the outer process is not corrupted. However, the obtained solution is much less accurate, as compared to the known exact solution. We do not have an exact explanation for this. For the inner iterative solver full GMRES was taken with the incomplete LU factorization of the matrix $BB^T + C$ as the preconditioner. Note that in case $BB^T + C$ is guaranteed to be positive definite the incomplete Cholesky factorization could be used as a preconditioner. The ILU preconditioner is useful but by no means crucial for the overall performance. The maximum number of iterations was taken (15 and 50, respectively) for the stable (Q₁-iso-Q₂)-P₀ and for the stabilized Q₁-P₀ discretizations. A lower value of the maximum iteration number for the stable discretization was taken for efficiency reasons, because of a very robust outer iteration convergence behavior observed in this case.

An example showing how the chosen inner stopping criterion affects the outer iteration convergence can be seen in Figure 3.1, where, in addition to the stopping criterion (3.1), convergence plots for a much stricter criterion are given. As we see, for the stable discretization the outer iteration convergence is hardly affected by the choice of the inner solver. This robust convergence behavior was observed for this discretization in almost all runs (see the results reported in section 3.3).

3.2. Skew and SSOR preconditioners. The convergence results of section 2.1 were obtained for the skew preconditioner (1.3), (1.5) rather than for the SSOR preconditioner (1.3), (1.4) we are aiming at. As it has been already stated, the skew preconditioner should be an increasingly better approximation to SSOR as the skew-symmetric component S of A grows in norm. In practice, we do observe the similar

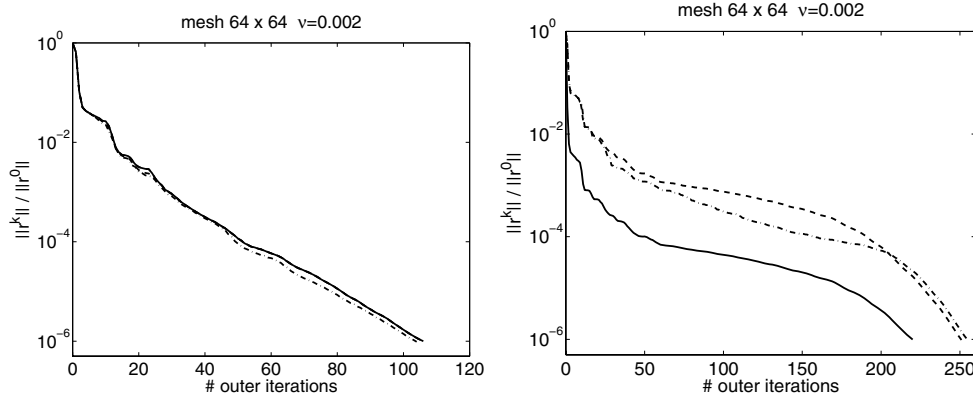


FIG. 3.1. Convergence plots for preconditioned GMRES with three different Schur complement solvers: direct solver with LU factorization (solid line), inner iteration with the strict stopping criterion $\|q^k\|_2 \leq 10^{-6} \|q^0\|_2$ (dashed line), and inner iteration with stopping criterion (3.1) (dash-dotted line). Left plot: the stable $(Q_1\text{-iso-}Q_2)\text{-}P_0$ discretization. Right plot: the stabilized $Q_1\text{-}P_0$ discretization. In both cases $\nu = 0.002$ and 64×64 mesh is used. For the stable discretization, accuracy of the inner solver has almost no influence on the outer iteration convergence.

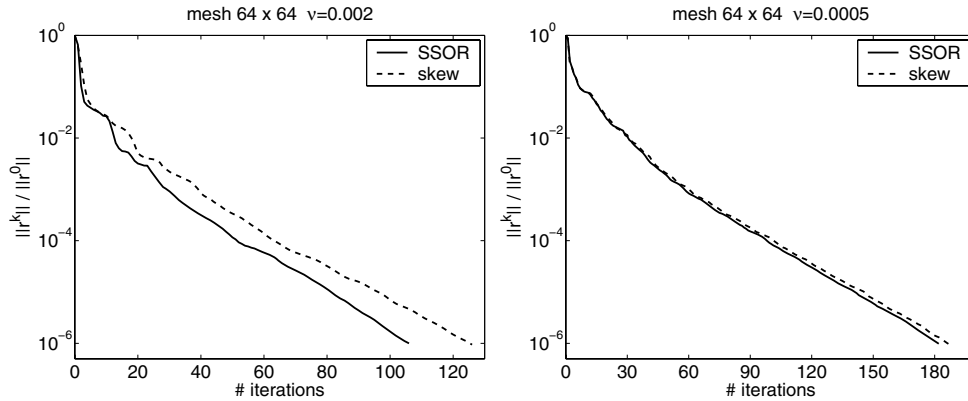


FIG. 3.2. Convergence plots for GMRES preconditioned with the skew and SSOR preconditioners. Left plot: 64×64 mesh, $\nu = 0.002$, right plot: 64×64 mesh, $\nu = 0.0005$. The stable $(Q_1\text{-iso-}Q_2)\text{-}P_0$ discretization is used.

convergence behavior of both preconditioners already for not so small viscosity values ν (note that $\|S\| \sim 1/\nu$). This can be seen in Figure 3.2 and in the results reported in section 3.3.

3.3. Comparison with other preconditioners. Here we present results of the comparison of the SSOR and skew preconditioners with the BFB^T preconditioner [16], the block-triangular preconditioner [19], and the Stokes preconditioner [30]. Together with the “exact” version of the SSOR preconditioner, we test the inexact SSOR preconditioner implemented as explained in section 3.1.

We emphasize that comparison of inexact forms of SSOR with inexact forms of other preconditioners is beyond the scope of the paper. We restrict ourselves to demonstrating that

- (i) the new preconditioning technique in its exact form can be very competitive

with other techniques for the cases where the exact implementations are feasible and require comparable work,

(ii) the inexact form of the new preconditioner (presented in section 3.1) is robust and leads to only a moderate increase of the number of the outer iterations.

In the Stokes preconditioner, $\mathcal{P}$ is taken in the same way as in (1.3), with P being the symmetric component H of A . Therefore, we expect this preconditioner to work well only for weakly nonsymmetric systems (for large viscosity values). We implement the Stokes preconditioner by computing the Cholesky factorization of H and then following the same procedure as for the SSOR and skew preconditioners (see section 2.6).

The BFB^T and the block-triangular preconditioners were used for, respectively, the stable (Q₁-iso-Q₂)-P₀ and the stabilized Q₁-P₀ discretizations. These preconditioners were selected for comparison as the most efficient preconditioners provided by the software [50] for each of these discretizations.

The costs of the SSOR, BFB^T , and block-triangular preconditioners, as they are implemented for this test, are reported in Tables 1–3. For this model problem the parameters appearing in the cost estimations have the following values: $l = u = 4$, $r = 10$, and $s = 0$ for the stable discretization and $s = 3$ for the stabilized discretization. For the values of m and n see Tables 4 and 5.

Both the BFB^T and the block-triangular preconditioners involve action of A^{-1} on a given vector; this was implemented by the LU factorization (which was computed once and reused every iteration; see Tables 2 and 3). This factorization and corresponding back/forward substitutions (with costs of $2n^2$ and $4n\sqrt{n}$ operations, respectively) were done very fast, taking a hardly noticeable part of the total CPU time.

The SSOR and BFB^T preconditioners both required $O(m^3)$ floating point operations at the initialization stage, and an iteration of BFB^T was approximately a factor three more expensive than an SSOR iteration. The initialization stages for both preconditioners took up a significant, sometimes dominant part of the total CPU time. For this reason the reported CPU time is not proportional to the number of iterations.

We have also made tests with a simple constraint preconditioner (1.3) with P taken to be the identity matrix which, due to the applied diagonal prescaling, is the diagonal of A . The matrix $BP^{-1}B^T + C = BB^T + C$ is a sparse band matrix with bandwidth $\sqrt{n}$, and the costs of this preconditioner scale linearly with m (see section 2.6).

The results of the comparisons are presented in Tables 4 and 5. As we see in Table 4 the SSOR preconditioner competes quite well with the other techniques, especially for the small values of ν , when the matrix A is far from symmetric. For this stable discretization the inexact form is much slower than the exact form (even though the maximum number of the inner iterations was restricted to 15; see section 3.1). This is because of the high efficiency of direct solvers in MATLAB software, two-dimensionality of the test problem, and its size. The situation is different for the stabilized discretization (see Table 5): in the SSOR, skew, and Stokes preconditioners the costs for the Schur complement system solution are increased due to the larger values of m , and the simple constraint preconditioner with $P = I$ appears to be the most efficient in this case. The performance of this preconditioner, however, deteriorates strongly as ν decreases and A becomes more nonsymmetric. We also see that the inexact SSOR preconditioner is more competitive for this problem.

4. Conclusions and an outlook to future research. As our analysis and experiments suggest, the nonsymmetric preconditioner approach (1.3), (1.4) appears

TABLE 4

The CPU time (seconds) and number of iterations (given in brackets) for different preconditioners, mesh sizes, and viscosity parameters ν . The stable $(Q_1\text{-iso-}Q_2)\text{-}P_0$ discretization. One BFB^T iteration is approximately a factor three more expensive than one SSOR or skew iteration (for estimation of costs per iteration see Tables 1 and 2). We emphasize that the reported CPU times are obtained for MATLAB codes and, thus, give only an indication of the performance.

$n = 2178, m = 256$ (mesh 32×32)						
ν	SSOR	InexSSOR	Skew	BFB^T	Stokes	$P = I$
0.1	1.4(24)	3.9(24)	2.4(64)	1.6(29)	2.1(16)	2.4(74)
0.01	1.9(41)	4.7(41)	2.3(60)	1.7(32)	3.3(85)	6.8(198)
0.005	2.0(43)	8.5(43)	2.1(50)	2.0(37)	8.7(141)	6.6(198)
0.002	2.6(69)	12(70)	2.7(71)	9.5(111)	16(261)	9.8(277)
$n = 8450, m = 1024$ (mesh 64×64)						
ν	SSOR	InexSSOR	Skew	BFB^T	Stokes	$P = I$
0.1	12(50)	27(50)	18(130)	21(42)	25(15)	13(143)
0.01	14(82)	51(83)	23(150)	22(46)	37(91)	51(437)
0.005	17(116)	68(117)	23(150)	22(45)	53(176)	70(546)
0.002	16(106)	61(104)	20(126)	23(50)	75(277)	54(494)
0.001	20(129)	73(126)	21(137)	46(124)	130(467)	65(568)
0.0005	26(182)	96(182)	27(187)	127(344)	209(693)	101(771)

TABLE 5

The CPU time (seconds) and number of iterations (given in brackets) for different preconditioners, mesh sizes, and viscosity parameters ν . The stabilized $Q_1\text{-}P_0$ discretization. “—” means that a preconditioner has not been tried for this case.

For estimation of costs per iteration see Tables 1 and 3. We emphasize that the reported CPU times are obtained for MATLAB codes and, thus, give only an indication of the performance.

$n = 2178, m = 1024$ (mesh 32×32)						
ν	SSOR	InexSSOR	Skew	Block-tr	Stokes	$P = I$
0.1	7.3(23)	5.2(23)	10(51)	0.7(28)	9.0(15)	1.8(76)
0.01	10(55)	19(55)	12(70)	12(286)	21(89)	6.2(231)
0.005	13(84)	28(85)	14(96)	35(549)	29(145)	8.2(326)
0.002	20(154)	50(163)	21(161)	78(796)	35(246)	16(486)
$n = 8450, m = 4096$ (mesh 64×64)						
ν	SSOR	InexSSOR	Skew	Block-tr	Stokes	$P = I$
0.1	272(48)	70(48)	470(120)	3.3(25)	280(15)	17(128)
0.01	296(69)	176(90)	414(162)	58(283)	384(90)	81(443)
0.005	509(139)	276(144)	622(159)	193(615)	708(159)	127(665)
0.002	684(220)	476(255)	561(276)	>1075(>1500)	654(270)	285(1002)
0.001	706(371)	705(386)	749(400)	>2500(>2000)	1264(425)	433(1348)
0.0005	963(574)	1097(589)	995(592)	—	—	718(1822)

to be an interesting alternative to other preconditioning techniques, especially when A is strongly nonsymmetric and when the Schur complement system can be efficiently solved (for example, because of its size or structure). When solution of the Schur complement system is expensive, inexact forms of the preconditioner can be employed. Our experiments show that the chosen simple strategy for the inner-outer iteration (namely, keeping the inner residual norm bounded by the outer iteration residual norm) usually works well in practice. However, if necessary, more can be done to minimize the overall work in the inner-outer iteration (see, e.g., [23]).

The framework introduced in section 2.1 can be applied to analyze any preconditioner (1.3) with P having the same skew-symmetric part as A . In fact, other choices of P are possible. For example, one could define a class of *skew incomplete LU factorizations* of A (cf. (1.4))

$$P := P_{\text{skewLU}} \equiv (D + L_S)D^{-1}(D + U_S),$$

where D is a diagonal matrix chosen such that $\text{Diag}(A) = \text{Diag}(P)$ or, for the modified version of ILU, such that the row sums in A and P are identical. Another class of skew preconditioners for the discretized Navier–Stokes problems can be obtained by using the rotation form of the equations [44, 45, 41]. These skew preconditioners will be a subject of future research.

Appendix. Here we further analyze the model problem of section 2.4. For the realistic choice $\omega = \omega_* \equiv 1/\sigma$, we inspect the effect of the β and γ on the eigenvalues of (2.19). This awkward expression reduces in this case to

$$(2.19') \quad X(P - A) = \begin{bmatrix} \frac{\sigma - 1}{\sigma(\gamma + \beta^2)}\beta^2 & \frac{1}{\sigma} \\ \frac{\sigma - 1}{\sigma(\gamma + \beta^2)}(\gamma - \beta^2) & -\frac{1}{\sigma} \end{bmatrix},$$

whose eigenvalues are (cf. Figure 4.1)

$$(4.1) \quad \lambda_{1,2} = \frac{1}{2} \left[\frac{\sigma - 1}{\sigma(\gamma + \beta^2)}\beta^2 - \frac{1}{\sigma} \pm \sqrt{\left(\frac{\sigma - 1}{\sigma(\gamma + \beta^2)}\beta^2 - \frac{1}{\sigma} \right)^2 + 4 \frac{\sigma - 1}{\sigma^2(\gamma + \beta^2)}\gamma} \right].$$

Analysis of the eigenvalues yields the following lemma.

LEMMA 4.1. *Let the matrix $X(P - A)$ given by (2.19') result from the action of the SSOR preconditioner (1.3), (1.4) on the blocks (2.17), $\sigma > 1$, and $\omega = \omega_* \equiv 1/\sigma$. Then for the spectral radius of $X(P - A)$ the following hold.*

1. $\rho(X(P - A)) = |1 - 2/\sigma|$ if $\gamma = 0$.
2. For $\gamma > 0$,

$$(4.2) \quad \rho(X(P - A)) = \begin{cases} |\lambda_2| & \text{if } 0 \leq |\beta| < \bar{\beta}, \\ |\lambda_1| & \text{otherwise,} \end{cases}$$

$$\bar{\beta} = \begin{cases} \sqrt{\frac{\gamma}{\sigma - 2}} & \text{if } \sigma > 2, \\ \infty & \text{otherwise.} \end{cases}$$

Furthermore, $\rho(X(P - A))$ decreases monotonically with $|\beta|$ whenever $\rho(X(P - A)) = |\lambda_2|$ or $\sigma \leq 3$. If $\rho(X(P - A)) = |\lambda_1|$, then it is a constant (monotonically increasing) function in $|\beta|$ for $\sigma = 3$ (respectively, for $\sigma > 3$). Finally,

$$(4.3) \quad \rho(X(P - A)) \leq \max \left\{ \frac{1 + \sqrt{4\sigma - 3}}{2\sigma}, 1 - \frac{2}{\sigma} \right\} \quad \text{for any } \beta, \gamma > 0, \sigma > 1.$$

Proof. For $\gamma = 0$, it follows directly from (2.20) that $\rho(X(P - A)) = |1 - 2/\sigma|$. For $\gamma > 0$ we analyze the eigenvalues (4.1) as functions of $|\beta|$. Since β appears in $\lambda_{1,2}$ only as β^2 , assume, without loss of generality, that $\beta > 0$. From (4.1), we have

$$\lambda_{1,2} = \frac{1}{2} (f(\beta) \pm \sqrt{f^2(\beta) + g(\beta)}),$$

$$f(\beta) = \frac{\sigma - 1}{\sigma(\gamma + \beta^2)}\beta^2 - \frac{1}{\sigma}, \quad g(\beta) = 4 \frac{\sigma - 1}{\sigma^2(\gamma + \beta^2)}\gamma.$$

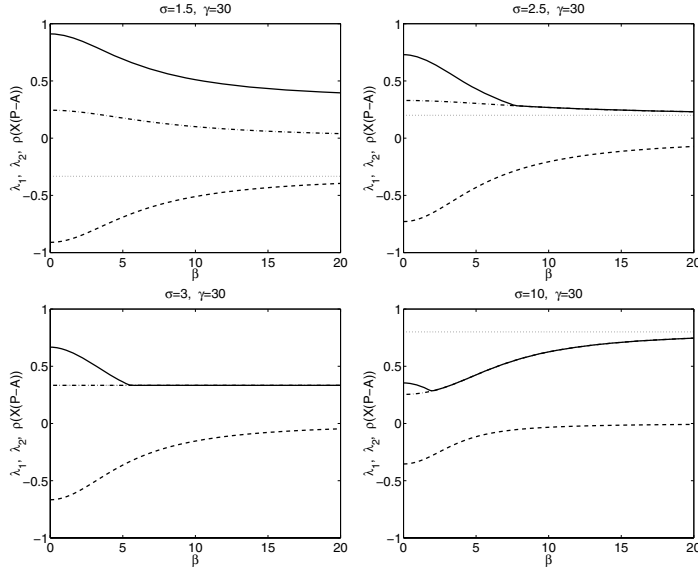


FIG. 4.1. An illustration of Lemma 4.1: the eigenvalues λ_1 (dash-dotted line), λ_2 (dashed line), and the spectral radius (solid line) of the matrix $X(P-A)$ against β for different values of σ . The dotted line is the asymptote $1 - \frac{2}{\sigma}$. The value of γ is taken arbitrarily.

If $\sigma > 1$, then $f'(\beta) > 0$ and derivatives of $\lambda_{1,2}$ with respect to β are positive if and only if

$$(4.4) \quad \sqrt{f^2(\beta) + g(\beta)} \pm \left(f(\beta) + \frac{g'(\beta)}{2f'(\beta)} \right) = \sqrt{f^2(\beta) + g(\beta)} \pm \left(f(\beta) + \frac{-2}{\sigma} \right) > 0.$$

The second of these inequalities (with the minus sign) corresponds to λ_2 and is always true. Hence, λ_2 monotonically increases with β . Multiplying the two inequalities (4.4) with each other, we obtain

$$g(\beta) - \frac{4}{\sigma^2} + \frac{4}{\sigma}f(\beta) > 0 \quad \Leftrightarrow \quad (\sigma - 3)\gamma > -(\sigma - 3)\beta^2,$$

which holds if and only if $\sigma > 3$. If $\sigma = 3$, then $\lambda_1 \equiv 1/3$. Furthermore, it is easy to see that $\lambda_{1,2}$ have different signs and that for $\beta = 0$ it holds $-\lambda_2 = \frac{1+\sqrt{4\sigma-3}}{2\sigma} > \lambda_1$. Moreover,

$$\lim_{\beta \rightarrow \infty} \lambda_1 = \max \left\{ 0, 1 - \frac{2}{\sigma} \right\}, \quad \lim_{\beta \rightarrow \infty} \lambda_2 = \min \left\{ 0, 1 - \frac{2}{\sigma} \right\}.$$

This completes the proof. $\square$

The eigenvalues and the spectral radius of $X(P-A)$ from (2.19') for the different situations described in Lemma 4.1 are plotted in Figure 4.1.

Lemma 4.1 provides information on the action of the SSOR preconditioner (1.3), (1.4) for the choice $\omega := \omega_* \equiv 1/\sigma$ which is made only for the blocks (2.17) with $\sigma = \sigma_{\max}$. For the other blocks this choice of ω will result in $\omega = \alpha/\sigma$ with $\alpha = \sigma/\sigma_{\max}$. Then the eigenvalues of these blocks either are given by (2.22) for $\gamma = 0$ or, for $\gamma > 0$, can be obtained in the same way as done in (2.19'), (4.1). The precise analysis of these eigenvalues is rather complicated and beyond the scope of this paper.

Acknowledgments. The first author thanks Henk van der Vorst and Gerard Sleijpen for stimulating discussions at early stages of this work. We thank the anonymous referees for helping us to improve this paper and Michele Benzi for useful comments and, in particular, for bringing to our attention [44, 45, 41].

REFERENCES

- [1] M. ARIOLI AND L. BALDINI, *A backward error analysis of a null space algorithm in sparse quadratic programming*, SIAM J. Matrix Anal. Appl., 23 (2001), pp. 425–442.
- [2] O. AXELSSON, *A generalized SSOR method*, Nordisk Tidskr. Informationsbehandling (BIT), 12 (1972), pp. 443–467.
- [3] Z.-Z. BAI, G. H. GOLUB, AND M. K. NG, *Hermitian and skew-Hermitian splitting methods for non-Hermitian positive definite linear systems*, SIAM J. Matrix Anal. Appl., 24 (2003), pp. 603–626.
- [4] M. BENZI AND G. H. GOLUB, *A preconditioner for generalized saddle point problems*, SIAM J. Matrix Anal. Appl., 26 (2004), pp. 20–41.
- [5] M. BENZI, G. H. GOLUB, AND J. LIESEN, *Numerical solution of saddle point problems*, Acta Numer., 14 (2005), pp. 1–137.
- [6] M. A. BOCHEV AND L. A. KRUKIER, *Iterative solution of strongly nonsymmetric systems of linear algebraic equations*, Russian Comput. Mathematics and Math. Physics, 37 (1997), pp. 1241–1251.
- [7] A. BOURAS AND V. FRAYSSÉ, *A Relaxation Strategy for Inexact Matrix-Vector Products for Krylov Methods*, Report TR/PA/00/15, CERFACS, Toulouse, France, 2000.
- [8] J. H. BRAMBLE, J. E. PASCIAK, AND A. T. VASSILEV, *Analysis of the inexact Uzawa algorithm for saddle point problems*, SIAM J. Numer. Anal., 34 (1997), pp. 1072–1092.
- [9] J. H. BRAMBLE, J. E. PASCIAK, AND A. T. VASSILEV, *Uzawa type algorithms for nonsymmetric saddle-point problems*, Math. Comp., 69 (2000), pp. 667–689.
- [10] S. C. BRENNER AND L. R. SCOTT, *The Mathematical Theory of Finite Element Methods*, Springer-Verlag, New York, 2002.
- [11] E. DE STURLER AND J. LIESEN, *Block-diagonal and constraint preconditioners for nonsymmetric indefinite linear systems. Part I: Theory*, SIAM J. Sci. Comput., 26 (2005), pp. 1598–1619.
- [12] H. S. DOLLAR AND A. J. WATHEN, *Approximate factorization constraint preconditioners for saddle-point matrices*, SIAM J. Sci. Comput., 27 (2006), pp. 1555–1572.
- [13] I. S. DUFF, N. I. M. GOULD, J. K. REID, J. A. SCOTT, AND K. TURNER, *The factorization of sparse symmetric indefinite matrices*, IMA J. Numer. Anal., 11 (1991), pp. 181–204.
- [14] N. DYN AND W. E. FERGUSON, JR., *The numerical solution of equality constrained quadratic programming problems*, Math. Comp., 41 (1983), pp. 165–170.
- [15] S. C. EISENSTAT, *Efficient implementation of a class of preconditioned conjugate gradient methods*, SIAM J. Sci. Statist. Comput., 2 (1981), pp. 1–4.
- [16] H. C. ELMAN, *Preconditioning for the steady-state Navier–Stokes equations with low viscosity*, SIAM J. Sci. Comput., 20 (1999), pp. 1299–1316.
- [17] H. C. ELMAN, *Preconditioners for saddle-point problems arising in computational fluid dynamics*, Appl. Numer. Math., 43 (2002), pp. 75–89.
- [18] H. C. ELMAN AND G. H. GOLUB, *Inexact and preconditioned Uzawa algorithms for saddle point problems*, SIAM J. Numer. Anal., 31 (1994), pp. 1645–1661.
- [19] H. ELMAN AND D. SILVESTER, *Fast nonsymmetric iterations and preconditioning for Navier–Stokes equations*, SIAM J. Sci. Comput., 17 (1996), pp. 33–46.
- [20] H. C. ELMAN, D. J. SILVESTER, AND A. J. WATHEN, *Iterative methods for problems in computational fluid dynamics*, in *Iterative Methods in Scientific Computing* (Hong Kong, 1995), Springer-Verlag, Singapore, 1997, pp. 271–327.
- [21] H. C. ELMAN, D. J. SILVESTER, AND A. J. WATHEN, *Performance and analysis of saddle-point preconditioners for the discrete steady-state Navier–Stokes equations*, Numer. Math., 90 (2002), pp. 665–688.
- [22] H. C. ELMAN, D. J. SILVESTER, AND A. J. WATHEN, *Finite Elements and Fast Iterative Solvers with Applications in Incompressible Fluid Dynamics*, Oxford University Press, New York, 2005.
- [23] E. GILADI, G. H. GOLUB, AND J. B. KELLER, *Inner and outer iterations for the Chebyshev algorithm*, SIAM J. Numer. Anal., 35 (1998), pp. 300–319.
- [24] P. E. GILL, W. MURRAY, D. B. PONCELEÓN, AND M. A. SAUNDERS, *Preconditioners for indefinite systems arising in optimization*, SIAM J. Matrix Anal. Appl., 13 (1992), pp. 292–311.

- [25] V. GIRAULT AND P.-A. RAVIART, *Finite Element Approximation of the Navier–Stokes Equations*, Springer-Verlag, Berlin, New York, 1979.
- [26] R. GLOWINSKI, *Numerical Methods for Nonlinear Variational Problems*, Springer Series in Computational Physics, Springer-Verlag, New York, 1984.
- [27] G. H. GOLUB AND C. GREIF, *On solving block-structured indefinite linear systems*, SIAM J. Sci. Comput., 24 (2003), pp. 2076–2092.
- [28] G. H. GOLUB AND M. L. OVERTON, *The convergence of inexact Chebyshev and Richardson iterative methods for solving linear systems*, Numer. Math., 53 (1988), pp. 571–593.
- [29] G. H. GOLUB AND C. F. VAN LOAN, *Matrix Computations*, 3rd ed., The Johns Hopkins University Press, Baltimore, MD, London, 1996.
- [30] G. H. GOLUB AND A. J. WATHEN, *An iteration for indefinite systems and its application to the Navier–Stokes equations*, SIAM J. Sci. Comput., 19 (1998), pp. 530–539.
- [31] G. H. GOLUB, X. WU, AND J.-Y. YUAN, *SOR-like methods for augmented systems*, BIT, 41 (2001), pp. 71–85.
- [32] N. I. M. GOULD, M. E. HRIBAR, AND J. NOCEDAL, *On the solution of equality constrained quadratic programming problems arising in optimization*, SIAM J. Sci. Comput., 23 (2001), pp. 1376–1395.
- [33] W. HACKBUSCH, *Iterative Solution of Large Sparse Systems of Equations*, Springer-Verlag, New York, 1994.
- [34] R. A. HORN AND C. R. JOHNSON, *Matrix Analysis*, Cambridge University Press, Cambridge, UK, 1985.
- [35] D. KAY, D. LOGHIN, AND A. J. WATHEN, *A preconditioner for the steady-state Navier–Stokes equations*, SIAM J. Sci. Comput., 24 (2002), pp. 237–256.
- [36] C. KELLER, N. I. M. GOULD, AND A. J. WATHEN, *Constraint preconditioning for indefinite linear systems*, SIAM J. Matrix Anal. Appl., 21 (2000), pp. 1300–1317.
- [37] A. KLAWONN, *An optimal preconditioner for a class of saddle point problems with a penalty term*, SIAM J. Sci. Comput., 19 (1998), pp. 540–552.
- [38] A. KLAWONN AND G. STARKE, *Block triangular preconditioners for nonsymmetric saddle-point problems: Field-of-values analysis*, Numer. Math., 81 (1999), pp. 577–594.
- [39] L. A. KRUKIER, *Implicit difference schemes and an iterative method for solving them for a certain class of systems of quasi-linear equations*, Izv. Vyssh. Uchebn. Zaved. Mat., 1979, no. 7, pp. 41–52 (in Russian); Soviet Math. (Iz. VUZ), 23 (1979), pp. 43–55 (in English).
- [40] L. A. KRUKIER, *Convergence acceleration of triangular iterative methods based on the skew symmetric part of the matrix*, Appl. Numer. Math., 30 (1999), pp. 281–290.
- [41] G. LUBE AND M. A. OLSHANSKII, *Stable finite-element calculation of incompressible flows using the rotation form of convection*, IMA J. Numer. Anal., 22 (2002), pp. 437–461.
- [42] P. MONK, *Finite Element Methods for Maxwell’s Equations*, Oxford University Press, New York, 2003.
- [43] J. NOCEDAL AND S. WRIGHT, *Numerical Optimization*, Springer-Verlag, New York, 1999.
- [44] M. A. OLSHANSKII, *An iterative solver for the Oseen problem and numerical solution of incompressible Navier–Stokes equations*, Numer. Linear Algebra Appl., 6 (1999), pp. 353–378.
- [45] M. A. OLSHANSKII AND A. REUSKEN, *Navier–Stokes equations in rotation form: A robust multigrid solver for the velocity problem*, SIAM J. Sci. Comput., 23 (2002), pp. 1683–1706.
- [46] W.-Q. REN AND J.-X. ZHAO, *Iterative methods with preconditioners for indefinite systems*, J. Comput. Math., 17 (1999), pp. 89–96.
- [47] Y. SAAD AND M. H. SCHULTZ, *GMRES: A generalized minimal residual algorithm for solving nonsymmetric linear systems*, SIAM J. Sci. Statist. Comput., 7 (1986), pp. 856–869.
- [48] V. SARIN AND A. SAMEH, *An efficient iterative method for the generalized Stokes problem*, SIAM J. Sci. Comput., 19 (1998), pp. 206–226.
- [49] C. SIEFERT AND E. DE STURLER, *Preconditioners for Generalized Saddle-Point Problems*, Report UIUCDCS-R-2004-2448, University of Illinois at Urbana-Champaign, 2004.
- [50] D. J. SILVESTER AND H. C. ELMAN, *Software for the Workshop “Theoretical and Practical Aspects of Incompressible CFD,”* Utrecht University, Mathematical Institute, Utrecht, The Netherlands, 1997.
- [51] D. J. SILVESTER, H. C. ELMAN, D. KAY, AND A. J. WATHEN, *Efficient preconditioning of the linearized Navier–Stokes equations for incompressible flow*, J. Comput. Appl. Math., 128 (2001), pp. 261–279.
- [52] V. SIMONCINI AND D. B. SZYLD, *Flexible inner-outer Krylov subspace methods*, SIAM J. Numer. Anal., 40 (2003), pp. 2219–2239.
- [53] J. VAN DEN ESHOF AND G. L. G. SLEIJPEN, *Inexact Krylov subspace methods for linear systems*, SIAM J. Matrix Anal. Appl., 26 (2004), pp. 125–153.
- [54] J. VAN DEN ESHOF, G. L. G. SLEIJPEN, AND M. B. VAN GIJZEN, *Iterative linear system solvers*

- with approximate matrix-vector products*, in QCD and Numerical Analysis III, Lect. Notes Comput. Sci. Eng. 47, A. Boriçi, A. Frommer, B. Joó, A. D. Kennedy, and B. Pendleton, eds., Springer-Verlag, Berlin, Heidelberg, New York, 2005, pp. 133–142.
- [55] A. VAN DER SLUIS AND H. A. VAN DER VORST, *The rate of convergence of conjugate gradients*, Numer. Math., 48 (1986), pp. 543–560.
 - [56] H. A. VAN DER VORST AND C. VUIK, *The superlinear convergence of GMRES*, J. Comput. Appl. Math., 48 (1993), pp. 327–341.
 - [57] H. A. VAN DER VORST AND C. VUIK, *GMRESR: A family of nested GMRES methods*, Numer. Linear Algebra Appl., 1 (1994), pp. 369–386.
 - [58] D. M. YOUNG, *Iterative Solution of Large Linear Systems*, Academic Press, New York, London, 1971.
 - [59] W. ZULENHER, *Analysis of iterative methods for saddle point problems: A unified approach*, Math. Comp., 71 (2002), pp. 479–505.

KERNEL PRESERVING MULTIGRID METHODS FOR CONVECTION-DIFFUSION EQUATIONS*

RANDOLPH E. BANK[†], JUSTIN W. L. WAN[‡], AND ZHENPENG QU[§]

Abstract. We propose a kernel preserving multigrid approach for solving convection-diffusion equations. The multigrid methods use Petrov–Galerkin coarse grid correction and linear interpolation. The restriction operator is constructed by preserving the kernel of the convection-diffusion operator. The construction considers constant and variable coefficient problems as well as cases where the convection term is not known explicitly. For constant convection-diffusion problems, we prove that the resulting Petrov–Galerkin coarse grid correction has small phase errors and the coarse grid matrix is almost an M-matrix. We demonstrate numerically the effectiveness of the multigrid methods by solving a constant convection problem, a recirculating flow problem, and a real application problem for pricing Asian options.

Key words. convection-diffusion equations, multigrid, kernel preserving, interpolation, restriction

AMS subject classifications. 65D05, 65F10, 65F50, 65M12, 65M55

DOI. 10.1137/040619533

1. Introduction. We consider multigrid methods to solve convection-diffusion equations of the form

$$-\nabla \cdot (\nabla u + \beta(x)u) = f, \quad x \in \Omega,$$

where $\beta(x) \gg 1$. The equation can be discretized by upwinding finite difference methods [19], finite element methods [20], and finite volume methods [1]. In any case, the result is a linear system of the form

$$(1.1) \quad A^h u^h = f^h.$$

When $\beta(x) \equiv 0$, the linear system can be solved efficiently using, for instance, fast Fourier transform, preconditioned conjugate gradient, multigrid, and domain decomposition methods. For large $\beta(x)$, however, A^h is a highly nonsymmetric matrix, and most of these solvers become less efficient.

Multigrid methods are widely used for solving elliptic partial differential equations (PDEs) because the convergence rates typically are independent of mesh size. For nonelliptic PDEs, in particular convection-diffusion equations, multigrid convergence often deteriorates with increasing convection. One reason is that simple elliptic multigrid approaches do not take into account the hyperbolic nature of the convection-diffusion operator. Thus modifications must be made in the smoother and coarse grid

*Received by the editors November 23, 2004; accepted for publication (in revised form) by M. Benzi May 11, 2005; published electronically March 17, 2006.

<http://www.siam.org/journals/simax/27-4/61953.html>

[†]Department of Mathematics, University of California-San Diego, La Jolla, CA 92093-0112 (rbank@ucsd.edu). The work of this author was supported by the National Science Foundation under contracts DMS-9973276 and DMS-0208449.

[‡]School of Computer Science, University of Waterloo, Waterloo, ON, N2L 3G1, Canada (jwlwan@uwaterloo.ca). This author was supported by the Natural Sciences and Engineering Research Council of Canada.

[§]Department of Combinatorics and Optimization, University of Waterloo, Waterloo, ON, N2L 3G1, Canada (zqu@uwaterloo.ca). This author was supported by the Natural Sciences and Engineering Research Council of Canada.

correction processes to improve convergence. Several smoothing techniques have been proposed to improve multigrid convergence. One approach is to apply Gauss–Seidel with the so-called downwind ordering [4, 7, 13, 18, 27]. Another approach is to use time-stepping schemes as smoothers for pure hyperbolic problems [15, 16, 17, 21]. Here the smoothers do not just reduce the high frequency errors, but also propagate the errors along the flow directions. The solution on the coarsest grid is typically done by a few smoothing steps. In this case, the multigrid process can be interpreted as speeding up the wave propagation by taking larger time-step sizes on the coarse grids. As a result, errors can be removed rapidly by propagating them out of the boundary. Other possibilities include the use of line smoothers [22] or ILU smoothers [28, 29]. Although often effective, all these smoothers are relatively expensive to compute and apply, especially in three dimensions. Thus, in this paper we primarily consider relaxation smoothers and focus on the choice and construction of the interpolation and restriction operators.

Coarse grid correction leads to more subtle issues. Applying a first differential approximation (FDA) analysis and local mode analysis, one can show that for a model convection diffusion with constant coefficients, if the convection is not aligned with the grid and direct discretization is used to form the coarse grid matrix, the resulting two-level convergence factor is at best 0.5 [8]. The recent phase error analysis [25] leads to the same conclusion. These negative results pose the fundamental challenge of multigrid for solving convection-diffusion equations.

One remedy is to apply artificial viscosity on the coarse grid matrices if the amount of viscosity can be determined a priori [33]. Another approach is to use Galerkin coarsening [32]. However, the Galerkin coarse grid matrices tend to central difference operators, leading to stability problems on coarse grids [12, 33]. In a recent analysis [30], it is shown that more accurate discrete coarse grid operators should be used for the direct discretization approach, and more accurate prolongation and restriction operators for the Galerkin approach. In another analysis [23], it is proved for a model problem that a Galerkin coarse grid matrix is an M-matrix for certain matrix-dependent interpolation. The Petrov–Galerkin approach has also been considered, where the restriction and interpolation operators are allowed to be different. In [6, 31], different combinations of restriction (downstream residual transfer, full weighting) and interpolation (bilinear interpolation, upstream interpolation) operators are tested. In [10], the restriction and interpolation operators are constructed by applying the black box interpolation idea [9] to $(A^h)^T$ and the symmetric part of A^h , respectively. In [30], bilinear interpolation is also considered, but different restrictions are used for forming the coarse grid operator and performing the intergrid transfer.

In this paper, we consider the Petrov–Galerkin coarsening approach. We employ linear interpolation unless otherwise stated. The goal is to design a good restriction operator that exploits the hyperbolic nature of the PDEs and is flexible enough to be applied on regular Cartesian grids as well as general unstructured grids. The latter is not addressed much in the approaches mentioned above. We also note that the construction of the interpolation and restriction operators in this paper is closely related to the construction of appropriate test and trial spaces in the hierarchical basis multigrid method [2, 3].

In section 2, we describe the construction of our kernel preserving restriction operator. We first assume the convection $\beta(x)$ is known explicitly, and then we generalize the kernel preserving restriction operator to the case when $\beta(x)$ is not given explicitly. In section 3, we justify theoretically the importance of preserving the kernel in the construction of the restriction operator for the two-grid case. In particular, we

prove that the coarse grid matrix is near an M-matrix and the phase error of the coarse grid correction process is negligibly small. In section 4, we demonstrate the effectiveness of our multigrid methods by solving model problems as well as a real application problem: Asian option pricing. We make some concluding remarks in section 5.

2. Kernel preserving multigrid. For simplicity, we assume $\Omega = (0, 1) \times (0, 1)$, the fine grid Ω^h is a regular triangular mesh with mesh size $h = 1/N$, and the grid points are given by

$$\{(x_i, y_j) : x_i = ih, y_j = jh, 0 \leq i, j \leq N\}.$$

Discrete functions $v^h = (v_{i,j}^h)$ are grid functions defined on Ω^h . The discrete problem (1.1) is solved using K grids, $\{\Omega^l\}_{l=0}^{K-1}$, where the finest grid is Ω^{K-1} , and Ω^{l-1} is obtained from Ω^l by standard full coarsening [24]. Denote by P^l the interpolation (or prolongation) operator from Ω^{l-1} to Ω^l , and by R^l the restriction operator from Ω^l to Ω^{l-1} , $l = 1, 2, \dots, K-1$. Also, denote the smoothing operator on Ω^l by S^l . A standard multigrid V(q_1, q_2)-cycle algorithm [5, 12] consists of q_1 steps of presmoothing, q_2 steps of postsmoothing, and coarse grid corrections performed recursively on each coarse grid. On the coarsest grid, the discrete problem is solved exactly.

In this paper, we consider simple relaxation smoothers, e.g., damped Jacobi and Gauss-Seidel with natural ordering. We do not employ any special ordering in the smoothing, which can be quite complicated in general, especially in high dimensions. For coarse grid correction, we consider the Petrov-Galerkin approach, i.e., in the two-grid setting, the coarse grid operator is obtained from

$$A^H = RA^hP,$$

where P and R are the interpolation and restriction operators, respectively, with $R \neq P^T$ in general. The focus of the paper is on constructing P and R such that A^H is an accurate, and, more important, stable approximation to A^h . Accuracy and stability issues will be made more precise in section 3.

Our key idea for constructing P and R is to preserve the kernel of appropriate PDE operators. Specifically, we require the interpolation operator to preserve exactly certain kernel functions of the Laplace operator $\nabla \cdot \nabla$, namely constant and linear functions. A natural choice is linear interpolation and its construction is standard. For the restriction operator, we require it to preserve certain kernel functions of the convection-diffusion operator

$$\nabla \cdot (\nabla u + \beta u).$$

The details require further explanation. In the following, we discuss the construction of R , starting with simple cases, followed by more general situations.

2.1. Convection is known. When $\beta(x)$ is given analytically, conditions required to preserve the kernel functions are sufficient to determine the restriction operator when the fine mesh is obtained from regular refinement of a coarse triangular mesh. Here each noncoarse grid point is connected to precisely two coarse grid points.

2.1.1. Constant β . When $\beta(x)$ is constant, the kernel of the PDE operator

$$\nabla \cdot (\nabla u + \beta u)$$

consists of constant and exponential functions of the following form:

$$u(x) = \text{constant} \quad \text{and} \quad u(x) = e^{-\beta \cdot x}.$$

For simplicity, we describe the construction first in one dimension.

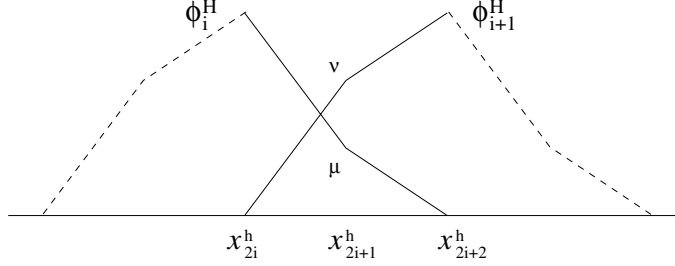


FIG. 2.1. 1D (one-dimensional) basis.

Consider the basis functions ϕ_i^H, ϕ_{i+1}^H as shown in Figure 2.1. Let $\mu = \phi_i^H(x_{2i+1}^h)$ and $\nu = \phi_{i+1}^H(x_{2i+1}^h)$. Thus, the stencil of the restriction operator is given by $[\nu \ 1 \ \mu]$. To determine μ and ν , we require that ϕ_i^H and ϕ_{i+1}^H locally preserve the kernel exactly at x_{2i+1}^h . Specifically, for constant kernel functions, we require

$$(2.1) \quad \mu + \nu = 1.$$

For the exponential kernel functions, we require

$$(2.2) \quad e^{\beta h} \mu + e^{-\beta h} \nu = 1,$$

where h is the fine mesh size. Equations (2.1)–(2.2) can be easily solved as follows:

$$(2.3) \quad \mu = \frac{1}{1 + e^{\beta h}}, \quad \nu = \frac{1}{1 + e^{-\beta h}}.$$

These are the restriction weights for the multigrid algorithm. It is easy to construct once β is known. Note that when $\beta = 0$, i.e., pure diffusion, we recover full-weighting restriction (i.e., transpose of the linear interpolation). When β is large, however, it becomes an upwind biased piecewise constant restriction. One may view this as a generalization of the full-weighting restriction to convection-diffusion equations. We remark that the interpolation weights given in (2.3) have also been considered in [3] for constant coefficient PDEs. In this paper, we explore this idea further for the case where the PDE coefficients are not constants or are even unspecified (and hence the kernel functions are not known) and for the case where some of the noncoarse grid points are connected to more than two coarse grid points.

In two dimensions (and similarly in higher dimensions), consider a generic coarse grid basis function ϕ_i^H defined on a standard uniform triangulation of a square domain as shown in Figure 2.2. Circles indicate coarse grid points and crosses fine grid points. We compute the restriction weights at the noncoarse grid points in the same manner as in the 1D case, and hence the derivation is omitted here. The resulting 7-point stencil of the restriction operator is then given by

$$(2.4) \quad \begin{bmatrix} 0 & \frac{1}{1+e^{-\beta \cdot (0,h)}} & \frac{1}{1+e^{-\beta \cdot (h,h)}} \\ \frac{1}{1+e^{-\beta \cdot (-h,0)}} & 1 & \frac{1}{1+e^{-\beta \cdot (h,0)}} \\ \frac{1}{1+e^{-\beta \cdot (-h,-h)}} & \frac{1}{1+e^{-\beta \cdot (0,-h)}} & 0 \end{bmatrix}.$$

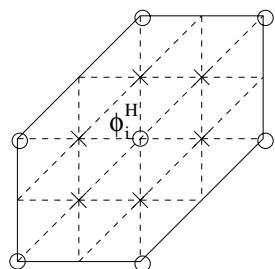


FIG. 2.2. The support of a 2D (two-dimensional) coarse basis on a triangular grid. The circles denote the coarse grid connections and the crosses the noncoarse grid connections.

As in the 1D case, when $\beta = 0$, (2.4) reduces to linear restriction on triangular grids, as desired. When β is large, it becomes an upwind biased piecewise constant restriction.

Remark. If the Cartesian grid is used instead, together with standard coarsening, then some of the noncoarse grid points will be connected to four coarse grid points. Thus, the above construction will fail. See section 2.2 for the general case.

2.1.2. Variable β . The analytic form of the kernel of the convection term with variable coefficients is unknown in general. However, since restriction is a local process, we approximate the convection coefficient $\beta(x)$ locally by a constant. One approach is to pick the value of $\beta(x)$ at the noncoarse grid point and then apply the previous restriction. Specifically, consider Figure 2.3(a), which shows two generic coarse grid points in two dimensions. Let β be the value of the convection at $x_{2i+1,2j}$, i.e.,

$$(2.5) \quad \beta = \beta(x_{2i+1,2j}).$$

Then, the restriction weights are given by (2.4).

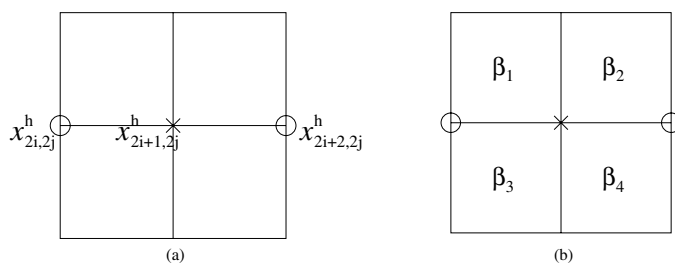


FIG. 2.3. (a) Coarse connection of a noncoarse grid point at $x_{2i+1,2j}$. (b) Values of $\beta(x)$ near a noncoarse grid point.

One may consider averaging the nearby values of $\beta(x)$. Consider Figure 2.3(b). We can define β as

$$(2.6) \quad \beta = \frac{\beta_1 + \beta_2 + \beta_3 + \beta_4}{4}.$$

The numerical results in section 4 show that these two approaches lead to essentially the same multigrid convergence.

Another approach is to preserve the exponential kernel function $e^{-\beta \cdot x}$ for non-constant β locally. Although it is no longer in the kernel of the variable coefficient

PDE it remains a good local approximation. Let μ and ν be the restriction weights from the left and right coarse grid points, respectively, and let β_W , β_E , and β_M be the corresponding values of $\beta(x)$ at the two coarse grid points (x_W, x_E) and at the noncoarse grid point (x_M) in between. As in the 1D case, we impose two constraints as follows:

$$\begin{aligned}\mu + \nu &= 1, \\ e^{-\beta_W \cdot x_W} \mu + e^{-\beta_E \cdot x_E} \nu &= e^{-\beta_M \cdot x_M}.\end{aligned}$$

After some simple calculations, we obtain the restriction weights

$$(2.7) \quad \mu = \frac{e^{-\beta_M \cdot x_M} - e^{-\beta_E \cdot x_E}}{e^{-\beta_W \cdot x_W} - e^{-\beta_E \cdot x_E}}, \quad \nu = \frac{e^{-\beta_W \cdot x_W} - e^{-\beta_M \cdot x_M}}{e^{-\beta_W \cdot x_W} - e^{-\beta_E \cdot x_E}}.$$

The coordinates of the grid points are also used in this case. It is noted that the weights in (2.7) equal those in (2.4) in the case of $\beta_M = \beta_W = \beta_E$.

2.2. Convection not known. In the previous section, the construction of the restriction operators made use of the explicit knowledge of $\beta(x)$; more precisely, the values of β at the grid points. While many applications provide such information, there are also situations where this information is not easily available. Furthermore, all these constructions assume the noncoarse grid points are connected to precisely two coarse grid points, which is hardly the case for unstructured grids, for example. In this section, we describe an algebraic approach that requires only the discretization matrix and exploits the elliptic and hyperbolic aspects of the PDE via an energy minimization formulation.

2.2.1. Constant β . For the moment, assume the convection is constant and known, so the kernel of the convection operator is also known. Eventually β is not needed, but here we treat it as known to explain how a constrained minimization is formulated as described below. Now, however, we allow noncoarse grid points to have more than two coarse connections. As a result, there may be more degrees of freedom than constraints preserving the kernel functions. We formulate a minimization problem to select the “optimal” choice of restriction weights. The formulation is motivated by the elliptic PDE case. For elliptic PDEs, the energy minimization approach [26] defines an interpolation by utilizing coarse grid basis functions which satisfy

$$\begin{aligned}\min_{\phi_i^H \in V^H} \quad & \sum_{i=1}^M \|\phi_i^H\|_A^2 \\ \text{subject to} \quad & \sum_{i=1}^M \phi_i^H(x_j^h) = 1, \quad j = 1, \dots, N,\end{aligned}$$

where $\|\cdot\|_A$ is the energy norm associated with the matrix A , and V^H consists of piecewise linear functions $v^H(x)$ such that $v^H(x_i) = 1$ and $v^H(x_j) = 0$ if x_j is not a neighboring node of x_i . The idea of the energy minimization approach is based on the classical convergence theory of multigrid, which requires the interpolation to have stability and approximation properties. The minimal energy norm of the coarse grid basis is used to satisfy the stability property, whereas the constant preserving constraint is used for the approximation property. Then an interpolation operator can be constructed from $\{\phi_i^H\}$. The resulting multigrid method has been shown to

be robust and efficient, and its convergence is independent of the grid size h and the size of jumps in the PDE coefficients for a wide variety of elliptic problems [26].

Unfortunately, the convergence theory of multigrid for convection-diffusion equations is not as well established. Nevertheless, we generalize the energy minimization formulation to convection dominated problems by a joint minimization formulation. Let $\{\phi_i^H\}$ and $\{\psi_i^H\}$ be the coarse grid basis functions corresponding to interpolation and restriction, respectively. Our approach is to compute $\{\phi_i^H\}$ and $\{\psi_i^H\}$ at the same time using the minimization problem

$$(2.8) \quad \min \sum_{i=1}^M (\psi_i^H, \phi_i^H)_A$$

subject to
$$\begin{cases} \sum_{i=1}^M \phi_i^H(x_j^h) = 1, \\ \sum_{i=1}^M e^{-\beta \cdot x_i^H} \psi_i^H(x_j^h) = e^{-\beta \cdot x_j^h}, \end{cases} \quad j = 1, \dots, N.$$

The operator $(\cdot, \cdot)_A$ is defined as follows. Let the vector representation of ψ_i^H and ϕ_i^H be $\phi^i = (\phi_1^i, \dots, \phi_N^i)$ and $\psi^i = (\psi_1^i, \dots, \psi_N^i)$, respectively. Then

$$(\psi_i^H, \phi_i^H)_A \equiv (\psi^i)^T A^h \phi^i.$$

Remarks.

1. When $\beta = 0$, A^h becomes the standard Laplace operator and is symmetric positive definite. Then $\{\psi_i^H\} = \{\phi_i^H\}$ and $(\psi_i^H, \phi_i^H)_A = \|\phi_i^H\|_A^2$, and hence the latter reduces to the elliptic formulation. Thus, the new formulation (2.8) generalizes the energy minimization approach to the convection-diffusion case.
2. The formulation has no assumptions on the geometry or dimension. Thus, it can be directly applied to problems defined on unstructured grids in any dimension.
3. In one dimension, the restriction operator constructed from $\{\psi_i^H\}$ is identical to the formula given in (2.3). However, the interpolation operator constructed from $\{\phi_i^H\}$ is not linear interpolation; in fact, it coincides with the restriction operator. As shown in section 4, although the multigrid convergence is somewhat slower than that for linear interpolation, it does show mesh and convection-independent convergence. We note, however, that one may simply use linear interpolation and disregard the one constructed from $\{\phi_i^H\}$.
4. In two dimensions, even the restriction is different from the formula given in (2.4), where only the kernel preserving is used. The interpolation is different from the restriction and it resembles a downwind piecewise constant interpolation for large β .

2.2.2. Nonconstant β . For variable and unspecified β , the kernel is not known analytically. Moreover, we cannot use local values of $\beta(x)$ to approximate the kernel, as $\beta(x)$ is not known explicitly either. Our approach is based on the observation that the kernel of the PDE is related closely to the zero eigenvectors of the stiffness matrix. (Note that we do not need the stiffness matrix in our algorithm.) More precisely, the right zero eigenvector corresponds to the constant null space and the left zero eigenvector corresponds to the exponential kernel functions. Thus, we want the interpolation to preserve the right zero eigenvector and the restriction to preserve the left zero eigenvector. For standard finite difference or finite element discretizations,

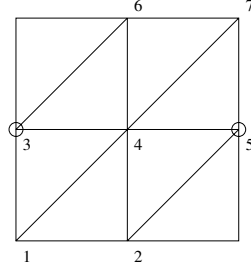


FIG. 2.4. Extraction of a local matrix at the noncoarse grid point 4 for the computation of left and right zero eigenvectors.

the right zero eigenvector of the discretization matrix is a vector of all 1's. Thus, it is consistent with the first constraint in our minimization formulation (2.8).

However, the left zero eigenvector is generally unknown, so we approximate it locally. Consider Figure 2.4 and, in particular, the noncoarse grid point 4. For a 7-point stencil matrix, we extract the local 7×7 submatrix corresponding to the neighbors of point 4. (For a 9-point stencil matrix, the submatrix will be of size 9×9 .) Replace the diagonal of the local matrix by the negative of the sum of the off-diagonal entries. The resulting matrix then resembles an element stiffness matrix.

Now compute the left and right zero eigenvectors of the local submatrix. Let $v^L = (v_1^L, \dots, v_7^L)$ be the left zero eigenvector. To preserve this vector locally at grid point 4, we impose the constraint

$$v_3^L \mu + v_5^L \nu = v_4^L$$

on the restriction weights μ, ν . This will be a constraint for the restriction operator. Similarly, the constraints for the interpolation operator can be set up using the right zero eigenvector. We note, however, that the right zero eigenvector is a constant vector, and hence they need not be computed and the corresponding constraints are simply set to preserve the constant 1.

We remark that for constant β , the left zero eigenvector coincides with the discrete exponential kernel function, and hence the minimization problem reduces to (2.8). However, the restriction is different from those given by (2.4) and (2.7), as mentioned in Remark 4 in the previous section.

2.2.3. Solution to minimization problem. We assume $\beta(x)$ is constant in the following. The procedure can be easily modified to the general case. For each coarse grid point i , write the coarse grid basis for interpolation as $\phi_i^H = \sum_{j=1}^n \varphi_j^i \phi_j^h$ and let $\varphi^i = (\varphi_1^i, \dots, \varphi_n^i)^T$. Similarly, we write the coarse grid basis for restriction as $\psi_i^H = \sum_{j=1}^n \psi_j^i \phi_j^h$ and $\psi^i = (\psi_1^i, \dots, \psi_n^i)^T$. Because of compact support, they are sparse vectors. Let $\Phi = [\varphi^1; \dots; \varphi^m]$ be an $mn \times 1$ vector obtained by concatenating all the φ 's. Do the same for Ψ . Then, (2.8) can be written as the following equivalent discrete linear constrained minimization problem:

$$(2.9) \quad \begin{aligned} & \min \quad \Psi^T \mathcal{A}^h \Phi \\ & \text{subject to} \quad \begin{cases} B_\Phi^T \Phi = \mathbf{1}, \\ B_\Psi^T \Psi = \mathbf{1}. \end{cases} \end{aligned}$$

The symbol $\mathbf{1}$ denotes a vector of all 1's. The $mn \times mn$ matrix $\mathcal{A}^h$ is block diagonal with each block equal to $\mathcal{A}_i^h$, which is defined as follows:

$$(\mathcal{A}_i^h)_{kl} = \begin{cases} A_{kl}^h & \text{if } \varphi_k^i \neq 0 \text{ and } \varphi_l^i \neq 0, \\ \delta_{kl} & \text{otherwise.} \end{cases}$$

The $n \times mn$ rectangular matrix $B_\Phi^T = [J_1 \cdots J_m]$ is defined as follows. J_i is a matrix corresponding to the injection operator that maps a vector v to v_i such that $(v_i)_k = (v)_k$ on $\text{supp}(\phi_i^H)$ and $(v_i)_k = 0$ otherwise. More precisely,

$$(J_i)_{kl} = \begin{cases} 1 & \text{if } k = l \text{ and } \varphi_k^i \neq 0, \\ 0 & \text{otherwise.} \end{cases}$$

The $n \times mn$ rectangular matrix $B_\Psi^T = [K_1 \cdots K_m]$ is defined similarly, where

$$(K_i)_{kl} = \begin{cases} e^{-\beta \cdot (x_k^h - x_i^H)} & \text{if } k = l \text{ and } \psi_k^i \neq 0, \\ 0 & \text{otherwise.} \end{cases}$$

Now, we solve the discrete linear constrained minimization problem (2.9) by the Lagrange multiplier formulation, which is equivalent to

$$(2.10) \quad \begin{bmatrix} \mathcal{A}^T & 0 & B_\Phi & 0 \\ 0 & \mathcal{A} & 0 & B_\Psi \\ B_\Psi^T & 0 & 0 & 0 \\ 0 & B_\Phi^T & 0 & 0 \end{bmatrix} \begin{bmatrix} \Psi \\ \Phi \\ \Lambda_\Phi \\ \Lambda_\Psi \end{bmatrix} = \begin{bmatrix} \mathbf{0} \\ \mathbf{0} \\ \mathbf{1} \\ \mathbf{1} \end{bmatrix},$$

where Λ_Φ and Λ_Ψ are $n \times 1$ vectors of the Lagrange multipliers. By eliminating Φ and Ψ from (2.10), we obtain the following equations for Λ_Φ and Λ_Ψ :

$$\begin{aligned} (B_\Psi^T \mathcal{A}^{-T} B_\Phi) \Lambda_\Phi &= -\mathbf{1}, \\ (B_\Phi^T \mathcal{A}^{-1} B_\Psi) \Lambda_\Psi &= -\mathbf{1}. \end{aligned}$$

Note that B_Φ , B_Ψ , and $\mathcal{A}^{-1}$ are sparse matrices. We can solve the linear systems by a Krylov subspace method, e.g., BiCGStab.

Once Λ_Φ , Λ_Ψ are known, Φ and Ψ can be computed by solving

$$\begin{aligned} \mathcal{A}\Phi &= -B_\Psi \Lambda_\Psi, \\ \mathcal{A}^T \Psi &= -B_\Phi \Lambda_\Phi. \end{aligned}$$

Since $\mathcal{A}$ is block diagonal and inverting each block corresponds to solving a small matrix (e.g., the matrix size is 7×7 for a 7-point stencil matrix), it is straightforward to compute Φ and Ψ .

3. Theoretical justification. It is well known that if direct discretization is used as the coarse grid operator, the convergence rate of multigrid can be very poor. One reason is that the phase error can be relatively large. If the Galerkin coarse grid operator is used instead, the phase error is significantly smaller. However, it is known that Galerkin coarse grid operators can be unstable on coarse grids [12, 33]. In this section, we prove that the Petrov–Galerkin coarse grid operator has small phase error and is stable on the coarse grid.

In the following analysis, we assume the velocity $\beta(x)$ is the constant vector (β, β) so that it is not aligned with the computational grid. Also, the convergence rate of

multigrid is usually poorest when the direction of the flow is at 45 degrees with the x-axis. Periodic boundary conditions are used, and we consider only two-grid analysis for the schemes described in section 2.1. Although the algorithm is applicable to matrices arising from different discretization methods, in the following analysis we consider primarily the Scharfetter–Gummel method [1]. We remark that unlike some other discretization methods such as streamline-upwinding Petrov–Galerkin [14], the Scharfetter–Gummel discretization gives rise to an M-matrix if the triangulation is a Delaunay triangulation. The M-matrix property is of particular interest since it will guarantee nonoscillatory numerical solutions. Given that $\beta(x) = (\beta, \beta)$, the 5-point stencil of the Scharfetter–Gummel discretization matrix is given by

$$\begin{bmatrix} & -\mathcal{B}(-\beta h) & \\ -\mathcal{B}(\beta h) & d & -\mathcal{B}(-\beta h) \\ & -\mathcal{B}(\beta h) & \end{bmatrix},$$

where $\mathcal{B}(x)$ is the Bernoulli function, defined as $\mathcal{B}(x) = x/(e^x - 1)$, and $d = 2\mathcal{B}(\beta h) + 2\mathcal{B}(-\beta h)$.

3.1. Phase error analysis. In [25], it was shown that the phase error of the coarse grid correction process is useful in determining the convergence rate of the corresponding multigrid method. It was proved that coarse grid matrices obtained from direct discretization lead to large phase errors, whereas the Galerkin coarse grid matrices lead to significantly smaller phase errors. In the following, we show that the phase error of the Petrov–Galerkin coarse grid matrix is also small.

The phase error analysis is based on a Fourier analysis of a two-grid method. Let the discrete Fourier function, $\psi_{\mu,\nu}^h$, be written as follows:

$$\psi_{\mu,\nu}^h(x_{j,k}^h) = \frac{1}{2} e^{i\mu\pi x_j^h} e^{i\nu\pi y_k^h}, \quad -N \leq \mu, \nu \leq N-1.$$

Thus, $\psi_{\mu,\nu}^h$, with $|\mu|, |\nu| \approx 0$, correspond to smooth or less oscillatory (low) modes, whereas $\psi_{\mu,\nu}^h$, $|\mu|, |\nu| \approx N$ correspond to the most oscillatory (high) modes. For two-grid analysis, it is customary to pair up the low-low, high-low, low-high, and high-high modes together, and hence the Fourier transform matrix Q_h is given by the following:

$$Q_h = [\cdots \psi_{\mu,\nu}^h \ \psi_{\mu,\nu'}^h \ \psi_{\mu',\nu}^h \ \psi_{\mu',\nu'}^h \cdots],$$

where $\mu' = \mu - N$, $\nu' = \nu - N$. Denote the Fourier transform of a matrix B by $\hat{B} \equiv Q_h^{-1} B Q_h$, with entries $\hat{B}_{\mu,\nu}$. In multigrid literature, $\hat{B}_{\mu,\nu}$ is often referred to as the Fourier symbol of B .

The iteration matrix of the $V(q_1, q_2)$ -cycle can be written as

$$M = S^{q_2} C S^{q_1},$$

where C and S are the coarse grid correction matrix and the iteration matrix of the smoother, respectively. Then $\hat{M} = \hat{S}^{q_2} \hat{C} \hat{S}^{q_1}$ is a block diagonal matrix with each block a 4×4 matrix $\hat{M}_{\mu,\nu}$, where

$$\hat{M}_{\mu,\nu} = \hat{S}_{\mu,\nu}^{q_2} \hat{C}_{\mu,\nu} \hat{S}_{\mu,\nu}^{q_1}, \quad \mu, \nu = -N/2, \dots, N/2 - 1.$$

Hence the convergence rate is determined by the spectral radii of $\hat{M}_{\mu,\nu}$, which in turn depend on

$$\hat{C}_{\mu,\nu} = I - \hat{P}_{\mu,\nu} (\hat{A}_{\mu,\nu}^H)^{-1} \hat{R}_{\mu,\nu} \hat{A}_{\mu,\nu}^h,$$

where P , R , and A^H are the prolongation, restriction, and coarse grid operators, respectively. In phase error analysis, one assumes the smoothers are effective at reducing high frequency errors. Thus, $\hat{S}_{\mu,\nu}(1,1)$, the $(1,1)$ entry of $\hat{S}_{\mu,\nu}$, is dominant and the other entries are either zero or negligible. Consequently, $\hat{M}_{\mu,\nu}$ is essentially determined by $\hat{C}_{\mu,\nu}(1,1)$, which represents the low frequency-low frequency interaction. In other words, it represents how the smooth waves are changed by the coarse grid correction. In theory, the ideal case is $\hat{C}_{\mu,\nu}(1,1) = 0$. In general, $\hat{C}_{\mu,\nu}(1,1)$ is a nonzero complex number. The phase error analysis is to analyze how $\hat{C}_{\mu,\nu}(1,1)$, in particular the phase angle of $\hat{C}_{\mu,\nu}(1,1)$, deviates from 0.

To analyze $\hat{C}_{\mu,\nu}(1,1)$, one first needs to compute the Fourier symbols $\hat{A}_{\mu,\nu}^h$, $\hat{A}_{\mu,\nu}^H$, $\hat{P}_{\mu,\nu}$, and $\hat{R}_{\mu,\nu}$.

LEMMA 3.1. *The symbols of $\hat{A}^h$, $\hat{A}^H$, $\hat{P}$, $\hat{R}$ are given by*

$$\hat{A}_{\mu,\nu}^h = \text{diag} \left(\begin{bmatrix} \frac{\beta h}{1-e^{\beta h}}(e^{-\mu\pi hi} + e^{\beta h}e^{\mu\pi hi} + e^{-\nu\pi hi} + e^{\beta h}e^{\nu\pi hi} - 2 - 2e^{\beta h}) \\ \frac{\beta h}{1-e^{\beta h}}(e^{-\mu\pi hi} + e^{\beta h}e^{\mu\pi hi} - e^{-\nu\pi hi} - e^{\beta h}e^{\nu\pi hi} - 2 - 2e^{\beta h}) \\ \frac{\beta h}{1-e^{\beta h}}(-e^{-\mu\pi hi} - e^{\beta h}e^{\mu\pi hi} + e^{-\nu\pi hi} + e^{\beta h}e^{\nu\pi hi} - 2 - 2e^{\beta h}) \\ \frac{-\beta h}{1-e^{\beta h}}(e^{-\mu\pi hi} + e^{\beta h}e^{\mu\pi hi} + e^{-\nu\pi hi} + e^{\beta h}e^{\nu\pi hi} - 2 - 2e^{\beta h}) \end{bmatrix} \right),$$

$$\hat{A}_{\mu,\nu}^H = \frac{1}{4} \left[\frac{b\omega^2 + \omega + 2}{2} \frac{e^{-2\mu\pi hi}}{\omega^2 + 1} + \frac{a}{2} \frac{2\omega^2 + \omega + 1}{\omega^2 + 1} e^{2\mu\pi hi} \right. \\ + \frac{b\omega^2 + \omega + 2}{2} \frac{e^{-2\nu\pi hi}}{\omega^2 + 1} + \frac{a}{2} \frac{2\omega^2 + \omega + 1}{\omega^2 + 1} e^{2\nu\pi hi} \\ + \frac{b(1-\omega)}{(\omega+1)(\omega^2+1)} e^{-2\mu\pi hi} e^{-2\nu\pi hi} + \frac{\omega^2(\omega-1)a}{(\omega+1)(\omega^2+1)} e^{2\mu\pi hi} e^{2\nu\pi hi} \\ \left. - b \frac{\omega^2 + \omega + 2}{\omega^2 + 1} - a \frac{2\omega^2 + \omega + 1}{\omega^2 + 1} - b \frac{1-\omega}{(\omega+1)(\omega^2+1)} - a \frac{\omega^2(\omega-1)}{(\omega+1)(\omega^2+1)} \right],$$

$$\hat{P}_{\mu,\nu} = \begin{bmatrix} c_\mu^2 c_\nu^2 \\ -s_\mu^2 c_\nu^2 \\ -c_\mu^2 s_\nu^2 \\ s_\mu^2 s_\nu^2 \end{bmatrix},$$

$$\hat{R}_{\mu,\nu}^T = \frac{1}{4} \left[\frac{\frac{1}{1+e^{\beta h}}(e^{-\mu\pi hi} + e^{\beta h}e^{\mu\pi hi} + e^{-\nu\pi hi} + e^{\beta h}e^{\nu\pi hi}) + \frac{1}{1+e^{2\beta h}}(e^{-(\mu+\nu)\pi hi} + e^{\beta h}e^{(\mu+\nu)\pi hi}) + 1}{\frac{1}{1+e^{\beta h}}(e^{-\mu\pi hi} + e^{\beta h}e^{\mu\pi hi} - e^{-\nu\pi hi} - e^{\beta h}e^{\nu\pi hi}) + \frac{1}{1+e^{2\beta h}}(e^{-(\mu+\nu)\pi hi} + e^{\beta h}e^{(\mu+\nu)\pi hi}) + 1} \right. \\ \left. \frac{\frac{1}{1+e^{\beta h}}(-e^{-\mu\pi hi} - e^{\beta h}e^{\mu\pi hi} + e^{-\nu\pi hi} + e^{\beta h}e^{\nu\pi hi}) + \frac{1}{1+e^{2\beta h}}(e^{-(\mu+\nu)\pi hi} + e^{\beta h}e^{(\mu+\nu)\pi hi}) + 1}{\frac{1}{1+e^{\beta h}}(e^{-\mu\pi hi} + e^{\beta h}e^{\mu\pi hi} + e^{-\nu\pi hi} + e^{\beta h}e^{\nu\pi hi}) + \frac{1}{1+e^{2\beta h}}(e^{-(\mu+\nu)\pi hi} + e^{\beta h}e^{(\mu+\nu)\pi hi}) + 1} \right],$$

where $c_\mu = \cos(\mu\pi h/2)$, $s_\mu = \sin(\mu\pi h/2)$, $\omega = e^{\beta h}$, and $\text{diag}(d)$ denotes a diagonal matrix whose diagonal is d .

Proof. The symbols are obtained by direct computation; the details are omitted. $\square$

From Lemma 3.1, it is clear that

$$\hat{C}_{\mu,\nu}(1,1) = 1 - c_\mu^2 c_\nu^2 \left[\frac{1}{1+e^{\beta h}}(e^{-\mu\pi hi} + e^{\beta h}e^{\mu\pi hi} + e^{-\nu\pi hi} + e^{\beta h}e^{\nu\pi hi}) \right. \\ \left. + \frac{1}{1+e^{2\beta h}}(e^{-(\mu+\nu)\pi hi} + e^{\beta h}e^{(\mu+\nu)\pi hi}) + 1 \right] \frac{\hat{A}_{\mu,\nu}(1,1)}{\hat{A}_{\mu,\nu}^H}.$$

To gain more insight into the formula of $\hat{C}_{\mu,\nu}(1, 1)$, we consider two special and yet important cases: the frequency components in the characteristic direction, i.e., (μ, ν) such that

$$\mu - \nu = 0,$$

and the cross-characteristic direction¹ [6, 8, 30], i.e., (μ, ν) such that

$$\mu + \nu = 0.$$

THEOREM 3.2. *For the components in the characteristic direction,*

$$\hat{C}_{\mu,\nu}(1, 1) = 1 - c_\mu^4 + O(\mu\pi h)^2.$$

For the components in the cross-characteristic direction,

$$\hat{C}_{\mu,\nu}(1, 1) = 1 - c_\mu^4.$$

Proof. We first consider the frequency components in the characteristic direction, i.e., $\nu = \mu$. Let $a = -\mathcal{B}(-\beta h)$ and $b = -\mathcal{B}(\beta h)$. By Lemma 3.1, we have

$$\begin{aligned} \hat{A}_{\mu,\mu}^h(1, 1) &= 2be^{-\mu\pi hi} + 2\omega be^{\mu\pi hi} - 2b - 2\omega b \\ &= 2b \left(-\mu\pi hi - \frac{(\mu\pi h)^2}{2} \right) + 2\omega b \left(\mu\pi hi - \frac{(\mu\pi h)^2}{2} \right) + O(\mu\pi h)^3. \end{aligned}$$

Similarly, $\hat{A}_{\mu,\mu}^H$ can be written as

$$\begin{aligned} \hat{A}_{\mu,\mu}^H &= \frac{b}{4(\omega^2 + 1)} \left[(\omega^2 + \omega + 2)e^{-2\mu\pi hi} + \omega(2\omega^2 + \omega + 1)e^{2\mu\pi hi} + \frac{1 - \omega}{1 + \omega}e^{-4\mu\pi hi} \right. \\ &\quad \left. + \frac{\omega^3(\omega - 1)}{\omega + 1}e^{4\mu\pi hi} - \frac{\omega^3 + 2\omega^2 + 2\omega + 3}{\omega + 1} - \frac{\omega(3\omega^3 + 2\omega^2 + 2\omega + 1)}{\omega + 1} \right] \\ &= \frac{b}{4(\omega^2 + 1)(\omega + 1)} \left[(\omega + 1)(\omega^2 + \omega + 2)(-2\mu\pi hi - 2(\mu\pi h)^2) \right. \\ &\quad \left. + \omega(\omega + 1)(2\omega^2 + \omega + 1)(2\mu\pi hi - 2(\mu\pi h)^2) + (1 - \omega)(-4\mu\pi hi - 8(\mu\pi h)^2) \right. \\ &\quad \left. + \omega^3(\omega - 1)(4\mu\pi hi - 8(\mu\pi h)^2) \right] + O(\mu\pi h)^3 \\ &= \frac{-\mu\pi hb}{(\omega^2 + 1)(\omega + 1)} \left[(\omega + 1)((1 - \omega^3)i + (\omega^3 + \omega^2 + \omega + 1)(\mu\pi h)) \right. \\ &\quad \left. + (1 - \omega)((\omega^3 + 1)i + 2(1 - \omega^3)(\mu\pi h)) \right] + O(\mu\pi h)^3 \\ &= \frac{-\mu\pi hb}{(\omega^2 + 1)(\omega + 1)} \left[2(1 - \omega^4)i + (3\omega^4 + 2\omega^2 + 3)(\mu\pi h) \right] + O(\mu\pi h)^3. \end{aligned}$$

Combining these two formulae, we have

$$\begin{aligned} \frac{\hat{A}_{\mu,\mu}^h(1, 1)}{\hat{A}_{\mu,\mu}^H} &= \frac{2(\omega + 1)(\omega^2 + 1)(i + \frac{\mu\pi h}{2}) + 2\omega(\omega + 1)(\omega^2 + 1)(-i + \frac{\mu\pi h}{2})}{2(1 - \omega^4)i + (3\omega^4 + 2\omega^2 + 3)(\mu\pi h)} + O(\mu\pi h)^2 \\ &= 1 - \frac{\omega^3 - 1}{(\omega + 1)(\omega^2 + 1)}\mu\pi hi + O(\mu\pi h)^2. \end{aligned}$$

¹These components are also called characteristic components in the literature.

We then make a Taylor expansion of $\hat{R}_{\mu,\mu}(1, 1)$ and obtain

$$(3.1) \quad \hat{R}_{\mu,\mu}(1, 1) = 1 + \frac{\omega^3 - 1}{(\omega + 1)(\omega^2 + 1)} \mu\pi h i + O(\mu\pi h)^2.$$

Finally,

$$\begin{aligned} \hat{C}_{\mu,\mu}(1, 1) &= 1 - c_\mu^4 \left(1 + \frac{\omega^3 - 1}{(\omega + 1)(\omega^2 + 1)} \mu\pi h i \right) \left(1 - \frac{\omega^3 - 1}{(\omega + 1)(\omega^2 + 1)} \mu\pi h i \right) + O(\mu\pi h)^2 \\ &= 1 - c_\mu^4 + O(\mu\pi h)^2. \end{aligned}$$

In the cross-characteristic direction, where $\nu = -\mu$,

$$\begin{aligned} \hat{A}_{\mu,-\mu}^h(1, 1) &= -b(2 - e^{-\mu\pi h i} - \omega e^{-\mu\pi h i} - e^{\mu\pi h i} - \omega e^{\mu\pi h i} + 2\omega) \\ &= -2b(\omega + 1)(1 - \cos(\mu\pi h)) \\ &= -4b(\omega + 1)s_\mu^2. \end{aligned}$$

Similarly,

$$\begin{aligned} \hat{A}_{\mu,-\mu}^H &= \frac{b}{8(\omega^2 + 1)} (\omega^2 + \omega + 2 + 2\omega^3 + \omega^2 + \omega)(e^{-2\mu\pi h i} + e^{2\mu\pi h i} - 2) \\ &= \frac{-b(\omega^3 + \omega^2 + \omega + 1)\sin^2(\mu\pi h)}{\omega^2 + 1}. \end{aligned}$$

For the restriction operator, we have

$$\hat{R}_{\mu,-\mu} = \frac{1}{4} \left(\frac{1}{\omega + 1} + \frac{\omega}{\omega + 1} \right) (e^{-\mu\pi h i} + e^{\mu\pi h i} + 2) = c_\mu^2.$$

Combining all the formulae, we obtain

$$\begin{aligned} \hat{C}_{\mu,-\mu}(1, 1) &= 1 - c_\mu^6 \frac{-4b(\omega + 1)s_\mu^2}{\frac{-b(\omega^3 + \omega^2 + \omega + 1)\sin^2(\mu\pi h)}{\omega^2 + 1}} \\ &= 1 - c_\mu^6 \frac{4(\omega + 1)(\omega^2 + 1)s_\mu^2}{(\omega^3 + \omega^2 + \omega + 1)4s_\mu^2 c_\mu^2} \\ &= 1 - c_\mu^4. \quad \square \end{aligned}$$

It is interesting to note that the phase error in the characteristic direction of the coarse grid matrix relative to the fine grid matrix can be quite significant. More precisely, for $\omega \rightarrow \infty$ (i.e., $\beta \rightarrow \infty$),

$$\frac{\hat{A}_{\mu,\nu}^h(1, 1)}{\hat{A}_{\mu,\nu}^H} \approx 1 - \mu\pi h i + O(\mu\pi h)^2 = e^{-\mu\pi h i} + O(\mu\pi h)^2.$$

If one ignores the effect of prolongation and restriction, then the coarse grid correction has the effect of shifting waves of any frequency by 1 grid point to the right. For $\omega \rightarrow 0$ (i.e., $\beta \rightarrow -\infty$),

$$\frac{\hat{A}_{\mu,\nu}^h(1, 1)}{\hat{A}_{\mu,\nu}^H} \approx e^{\mu\pi h i} + O(\mu\pi h)^2.$$

In this case, the convection has an opposite direction and the phase error has the effect of shifting waves by 1 grid point to the left. As a result, the coarse grid error is off by 1 grid point compared to the fine grid error, leading to at most first order accurate approximation. Fortunately, the restriction operator has the same effect with opposite sign, as can be seen by its symbol where the sign is positive for the term $\mu\pi h i$; see (3.1). It counteracts the shifting caused by the Petrov–Galerkin coarse grid matrix. As a result, one can easily derive from $\hat{C}_{\mu,\nu}(1, 1)$ that the phase error of the Petrov–Galerkin coarse grid correction is only $O(\mu\pi h)^2$. For comparison, the phase errors of the direct discretization and Galerkin coarse grid correction approaches are $1/2$ and $O(\mu\pi h)^3$, respectively [25]. Although Petrov–Galerkin is not as accurate as the Galerkin approach, it is certainly much better than direct discretization. Moreover, in the cross-characteristic direction, $\hat{C}_{\mu,\nu}(1, 1)$ is close to zero, in contrast with the direct discretization approach where $\hat{C}_{\mu,\nu}(1, 1) = 1/2$ [8, 30], which leads to a convergence rate of at most 0.5.

The phase error of Petrov–Galerkin coarse grid correction is not as small as that of Galerkin coarse grid correction. However, Galerkin coarse grid operators are often not stable. In the next section, we shall show the stability of Petrov–Galerkin coarse grid operators.

3.2. M-matrix property. If the fine grid discretization matrix is an M-matrix, there will be no spurious oscillation in the numerical solution. For stability, it is desirable that the coarse grid matrix inherits the M-matrix property of the fine grid matrix. If direct discretization is used to construct the coarse grid matrix, it would clearly be an M-matrix, but then it leads to poor coarse grid correction and large phase errors. On the other hand, if a Galerkin coarse grid matrix is used, it may not necessarily be an M-matrix.

Here, we prove that our Petrov–Galerkin coarse grid matrix with linear interpolation and exponential fitting restriction is almost an M-matrix. More precisely, it approaches an M-matrix asymptotically as $|\beta|$ approaches infinity. We first compute the entries of the coarse grid matrix.

LEMMA 3.3. *Denote the stencil of the coarse grid matrix A^H by*

$$\begin{bmatrix} & A_N^H & A_{NE}^H \\ A_W^H & A_C^H & A_E^H \\ A_{SW}^H & A_S^H & \end{bmatrix},$$

where A^H is obtained by linear interpolation and exponential fitting restriction (2.4). Then

$$\begin{aligned} A_N^H &= \frac{2\omega^2 + \omega + 1}{2(\omega^2 + 1)}a, & A_{NE}^H &= \frac{\omega^2(\omega - 1)}{(\omega + 1)(\omega^2 + 1)}a, \\ A_W^H &= \frac{\omega^2 + \omega + 2}{2(\omega^2 + 1)}b, & A_E^H &= \frac{2\omega^2 + \omega + 1}{2(\omega^2 + 1)}a, \\ A_{SW}^H &= \frac{1 - \omega}{(\omega + 1)(\omega^2 + 1)}b, & A_S^H &= \frac{\omega^2 + \omega + 2}{2(\omega^2 + 1)}b, \\ A_C^H &= -(A_N^H + A_{NE}^H + A_W^H + A_E^H + A_{SW}^H + A_S^H), \end{aligned}$$

where a , b , and ω are defined as before.

Proof. The computation of all the entries of A^H is similar so we show only the formula for A_W^H . Let I be the index of a coarse grid unknown and i be the index of the

corresponding unknown on the fine grid. Then $i = 2I$, assuming standard coarsening. Also, denote the number of fine grid points in the x-direction by nx . Let P and R be the linear interpolation and exponential fitting restriction operators, respectively. By the definition of the Petrov–Galerkin coarse grid matrix,

$$\begin{aligned}
A_W^H &= A_{I,I-1}^H = (RA^hP)_{I,I-1} \\
&= \sum_i R_{I,i} \sum_j A_{i,j}^h P_{j,I-1} \\
&= \sum_i R_{I,i} \left[\frac{1}{2} A_{i,2I-nx-3}^h + \frac{1}{2} A_{i,2I-nx-2}^h + \frac{1}{2} A_{i,2I-3}^h + A_{i,2I-2}^h + \frac{1}{2} A_{i,2I-1}^h \right. \\
&\quad \left. + \frac{1}{2} A_{i,2I+nx-2}^h + \frac{1}{2} A_{i,2I+nx-1}^h \right] \\
&= \frac{1}{\omega^2 + 1} \left(\frac{1}{2} A_{2I-nx-1,2I-nx-2}^h + \frac{1}{2} A_{2I-nx-1,2I-1}^h \right) \\
&\quad + \frac{1}{\omega + 1} \left(A_{2I-1,2I-2}^h + \frac{1}{2} A_{2I-1,2I-1}^h + \frac{1}{2} A_{2I-1,2I+nx-1}^h \right) \\
&\quad + \frac{1}{2} A_{2I,2I-1}^h + \frac{\omega}{\omega + 1} \left(\frac{1}{2} A_{2I+nx,2I+nx-1}^h \right) \\
&= \frac{1}{\omega^2 + 1} \left(\frac{1}{2} b + \frac{1}{2} a \right) + \frac{1}{\omega + 1} \left(b + \frac{1}{2} (-2a - 2b) + \frac{1}{2} a \right) \\
&\quad + \frac{1}{2} b + \frac{\omega}{\omega + 1} \left(\frac{1}{2} b \right) \\
&= \frac{\omega(1 - \omega)}{2(\omega^2 + 1)(\omega + 1)} a + \frac{2\omega^3 + \omega^2 + 3\omega + 2}{2((\omega^2 + 1)(\omega + 1))} b.
\end{aligned}$$

The result follows from the fact that $a = \beta h e^{\beta h} / (1 - e^{\beta h}) = \omega b$. $\square$

THEOREM 3.4. *The Petrov–Galerkin coarse grid matrix can be written as follows:*

$$A^H = \bar{A}^H + E^H,$$

where $\bar{A}^H$ is an M-matrix and $E^H = O(e^{-2|\beta|h})$ as $|\beta|$ approaches infinity.

Proof. Note that $a < 0$, $b < 0$, and $\omega \equiv e^{\beta h} > 0$. Thus, by Lemma 3.3, all of the off-diagonal entries of A^H are negative, except for A_{NE}^H and A_{SW}^H . Consider

$$A_{SW}^H = \frac{1 - \omega}{(\omega + 1)(\omega^2 + 1)} b.$$

If $\beta \leq 0$, then $\omega \leq 1$ and thus $A_{SW}^H \leq 0$. If $\beta > 0$, however, then $A_{SW}^H > 0$. Similarly, for

$$A_{NE}^H = \frac{\omega^2(\omega - 1)}{(\omega + 1)(\omega^2 + 1)} a,$$

it is nonpositive if $\beta \geq 0$ and positive if $\beta < 0$. Now, consider the case when $\beta \geq 0$. Let $\bar{A}^H$ have the same stencil as A^H with A_{SW}^H replaced by 0. Also, the diagonal entries of $\bar{A}^H$ are defined as the negative sum of the off-diagonal entries. Then it is clear that $\bar{A}^H$ is an M-matrix. For the other case when $\beta \leq 0$, we let $\bar{A}^H$ have

the same stencil as A^H with A_{NE}^H replaced by 0. Then, again, $\bar{A}^H$ is an M-matrix. Furthermore, when $\beta \rightarrow \infty$,

$$A_{SW}^H \rightarrow \frac{-1}{\omega^2} b = O(e^{-2\beta h}).$$

When $\beta \rightarrow -\infty$,

$$A_{NE}^H \rightarrow -\omega^2 a = O(e^{2\beta h}).$$

Thus $E^H = A^H - \bar{A}^H = O(e^{-2|\beta|h})$. $\square$

We note that A^H is not an M-matrix in exact arithmetic; however, the error term decays exponentially and becomes the order of roundoff error for sufficiently large β . So for convection dominated problems, we should expect numerically stable Petrov–Galerkin coarse grid matrices.

4. Numerical results. In this section, we present results of numerical experiments in two dimensions that demonstrate the effectiveness of the proposed multigrid methods. In all the numerical examples, the computational domain is $\Omega = [0, 1] \times [0, 1]$ with homogeneous Dirichlet boundary conditions. We discretize the PDE on a regular Cartesian mesh and finite difference (upwinding or otherwise stated) methods are used. In the multigrid procedure, a V-cycle is used with two pre- and two postsmoothings, unless stated otherwise. Damped Jacobi and Gauss–Seidel (GS) smoothers are considered. The coarse grid operators are obtained from Petrov–Galerkin coarsening with linear interpolation and restriction as specified in each example. Since some of the numerical results are compared with those reported in other papers, the stopping criterion varies. Typically, the iteration was terminated when the relative residual l_2 -norm was less than 10^{-6} or 10^{-8} .

Example 1. In this example, we consider the following constant coefficient convection-diffusion equation [10, Example 1]:

$$-\delta h \Delta u + u_x + u_y = f.$$

In [10], the smoother used is Kaczmarz with one presmoothing and one postsmoothing. Linear interpolation is used together with restriction obtained from applying the black box interpolation [9] idea to A^T . The coarse grid operator is formed by the Petrov–Galerkin approach. We used a simple damped Jacobi iteration as a smoother. As the cost of one Kaczmarz iteration is approximately equal to two Jacobi iterations, the number of pre- and postsmoothings is two instead. We used linear interpolation but exponential fitting restriction as defined in sections 2.1 and 2.2. The stopping criterion is 10^{-6} .

In order to compare with the results given in [10], we select $h = 1/32$ and vary δ . The convergence results are shown in Table 4.1. BMG denotes the black box multigrid and EMG denotes our multigrid method. The bracket specifies the smoother used. For relatively large δ , the two multigrid methods are comparable. However, as δ becomes smaller, the convergence of BMG deteriorates while EMG(Jac) remains efficient. In fact, EMG(Jac) shows somewhat better convergence for small δ . Hence, the kernel preserving restriction results in convection-independent multigrid convergence.

Table 4.1 also shows the multigrid convergence results for GS smoothers with different ordering. GS_1 denotes the ordering first in the x-direction (left to right) and then in the y-direction (bottom to top), which is just the natural ordering. GS_2 denotes the ordering first in the y-direction (top to bottom) and then in the x-direction

TABLE 4.1

Number of multigrid V-cycles for a convection-diffusion equation with convection term $\beta(x) = -[1, 1]$ and varying diffusion coefficient δh .

δ	BMG	EMG(Jac)	EMG(GS_1)	EMG(GS_2)	EMG(GS_3)	EMG(GS_4)
1	11	11	6	6	8	6
1/2	13	10	5	7	9	7
1/4	20	10	4	6	11	6
1/8	> 20	9	3	8	15	8
1/16	> 20	9	3	10	18	10

(left to right). GS_3 denotes the ordering first in the x-direction (right to left) and then in the y-direction (top to bottom). Finally, GS_4 denotes the ordering first in the y-direction (bottom to top) and then in the x-direction (right to left). The results are generally worse than those for damped Jacobi smoothing, except for EMG(GS_1), where the ordering is consistent with the flow direction. The worst results are given by EMG(GS_3), whose ordering is opposite to the flow direction. Thus, in contrast with diffusion-dominated problems, GS shows a poorer smoothing effect compared to damped Jacobi, but it can still be an effective smoother if the ordering happens to be consistent with the flow direction.

We also tested our multigrid method by fixing the diffusion term, i.e., $\delta h = 10^{-3}$, and varying the mesh size. The results are shown in Table 4.2. EMG(Jac) and EMG(GS_1) perform the best and their convergence rates are independent of the mesh size, h . The convergence rates of other EMG methods are much slower and are generally dependent on the mesh size, similar to the results shown in Table 4.1. We also note that their convergence behavior is somewhat irregular with respect to the mesh size; they converge faster when $h = 1/128$ than when $h = 1/64$. This may be due to the poor smoothing effect.

TABLE 4.2

Number of multigrid V(2,2)-cycles for a convection-diffusion equation with convection term $\beta(x) = -[1, 1]$, diffusion coefficient $\delta h = 10^{-3}$, and varying mesh size h .

h	EMG(Jac)	EMG(GS_1)	EMG(GS_2)	EMG(GS_3)	EMG(GS_4)
1/16	9	2	6	11	6
1/32	9	2	10	18	10
1/64	9	3	15	28	15
1/128	9	3	12	14	12

Example 2. We consider a variable coefficient problem, often known as the recirculating flow problem, as follows:

$$\epsilon \Delta u + au_x + bu_y = f,$$

where $a = 4x(x-1)(1-2y)$, $b = -4y(y-1)(1-2x)$. In this case, the kernel of the PDE operator is unknown and so we use a locally defined value of $\beta(x)$ to construct the restriction, as described in section 2.1.2. Denote the multigrid using the midpoint value of $\beta(x)$, (2.5), by $EMG(\beta_{midpt})$; the average of nearby values of $\beta(x)$, (2.6), by $EMG(\beta_{ave})$; and multigrid with restriction locally preserving $e^{-\beta \cdot x}$, (2.7), by $EMG(\beta_{weighted})$. In this example, we use GS with ordering depending on the sign of a and b ; see [30]. Basically, we sweep over the variables four times in the appropriate order, and in each sweep relax roughly one quarter of the variables as follows: in the first sweep only variables at locations where both $a(x)$ and $b(x)$ are nonnegative are

relaxed; in the second sweep only variables corresponding to locations where $a(x)$ is nonnegative and $b(x)$ is nonpositive are relaxed; and similarly for the third and fourth sweeps. ϵ is taken as 10^{-3} . The stopping criterion is 10^{-8} .

TABLE 4.3

Number of multigrid $V(2,1)$ -cycles for the recirculating flow problem. The multigrid methods obtained from different choices of β are compared.

h	$\text{EMG}(\beta_{\text{midpt}})$	$\text{EMG}(\beta_{\text{ave}})$	$\text{EMG}(\beta_{\text{weighted}})$
1/16	6	6	4
1/32	7	7	6
1/64	10	10	8
1/128	12	12	10

Table 4.3 shows the number of V-cycles of different multigrid methods. The choices of using the local midpoint $\beta(x)$ and using the average of local values of $\beta(x)$ give rise to two multigrid methods with similar convergence. All these methods, however, show a small logarithmic dependence on the mesh size.

We also compare our results with those of Yavneh [30], who developed an efficient multigrid method for solving the recirculating flow problems. He tested two types of coarse grid correction, non-Galerkin and Petrov–Galerkin. For the non-Galerkin approach, the coarse grid operator is obtained from two discretizations of different accuracy. He used the same GS smoothing as ours, $V(2,1)$ -cycle, bilinear interpolation, and full-weighted restriction. For the Petrov–Galerkin approach, the coarse grid operator is obtained from a linear interpolation and a restriction operator such that the scheme remains “upstream on all grids.” (Note: these operators are different from the intergrid transfer operators.) Artificial viscosity is also added to the coarse grid operator. The stopping criterion is 10^{-8} .

TABLE 4.4

Number of multigrid $V(2,1)$ -cycles for the recirculating flow problem. The multigrid methods are compared using different ϵ and mesh size h . The non-Galerkin and Petrov–Galerkin columns show the multigrid results given in [30]. N/A denotes unavailable results.

h	ϵ	$\text{EMG}(\beta_{\text{weighted}})$	$\text{EMG}(\beta_{\text{midpt}})$	Non-Galerkin	Petrov–Galerkin
1/64	∞	9	6	N/A	N/A
1/64	10^{-2}	11	8	N/A	N/A
1/64	10^{-3}	8	10	N/A	N/A
1/64	10^{-5}	8	11	N/A	N/A
1/128	∞	10	6	8	8
1/128	10^{-1}	10	6	8	8
1/128	10^{-3}	10	12	27	10
1/128	10^{-5}	10	12	19	13
1/128	10^{-7}	9	12	17	14
1/128	10^{-9}	9	12	N/A	N/A
1/256	∞	9	7	N/A	N/A
1/256	10^{-3}	10	12	N/A	N/A
1/256	10^{-5}	10	13	N/A	N/A
1/256	10^{-7}	10	13	24	12
1/256	10^{-9}	10	13	N/A	N/A

The results are shown in Table 4.4 (N/A denotes results not available in [30]). First, we see that the iteration numbers of EMG are insensitive to mesh size and ϵ . $\text{EMG}(\beta_{\text{weighted}})$ seems generally better than $\text{EMG}(\beta_{\text{midpt}})$ but the difference is

not very significant. The non-Galerkin approach discussed by Yavneh shows poorer convergence and stronger dependence on h and ϵ . The Galerkin approach discussed by Yavneh shows similar convergence behavior. However, the coarse grid operator is somewhat more complicated to construct and it is unclear how to construct the linear and restriction operators in general. Also, artificial viscosity was needed but it was not discussed how much one should add. For our multigrid approach, we use the same interpolation and restriction operators for both intergrid transfer and the construction of the coarse grid matrices. Also, we do not require artificial viscosity.

Example 3. We consider the same equation as in Example 1, but the multigrid method uses the restriction and interpolation operators constructed by solving the minimization problem described in section 2.2.3. Both damped Jacobi and GS smoothings are tested. The results are shown in Table 4.5.

TABLE 4.5

Number of multigrid $V(2,2)$ -cycles for a convection-diffusion equation with convection term $\beta(x) = -[1, 1]$, diffusion coefficient $\delta h = 10^{-3}$, and varying mesh size h . The restriction and interpolation operators are constructed from solving a minimization problem.

h	EMG(Jac)	EMG(GS_1)	EMG(GS_2)	EMG(GS_3)	EMG(GS_4)
1/16	8	2	6	9	6
1/32	9	2	11	17	11
1/64	11	3	16	28	16
1/128	12	3	16	28	16

This multigrid method does not take advantage of the Cartesian geometry and does not make explicit use of the knowledge of the convection term, and hence it is completely algebraic. Since it is more general, it is not expected to be as efficient as the multigrid method considered in Example 1. The results shown in Table 4.5, however, are comparable to those in Table 4.2. EMG(Jac) here is a bit slower than in Example 1 and also shows a slight dependence on the mesh size. On the whole, nonetheless, there is no significant difference between the results of the two EMG approaches. It would be interesting to see the results of this general approach to solving more general problems, e.g., unstructured grids. This will be considered in future research.

Example 4. We apply our EMG multigrid method to solve a real application problem: pricing Asian options. The price of an Asian option, V , can be found by solving a PDE in two dimensions based on the Black-Scholes equations:

$$(4.1) \quad \frac{\partial V}{\partial \tau} = \frac{1}{2} \sigma^2 S^2 \frac{\partial^2 V}{\partial S^2} + rS \frac{\partial V}{\partial S} + \frac{1}{T - \tau} (S - A) \frac{\partial V}{\partial A} - rV,$$

where S is the stock price, A is the arithmetic average of the stock price over some time interval T , τ is the time from the expiration date T , and r and σ are constants representing the risk-free interest rate and volatility. It is not within the scope of this paper to derive this equation; we refer the interested readers to [11, 34]. For fixed strike call options, the terminal boundary condition is

$$V(S(T), A(T), T) = \max(A(T) - K, 0),$$

where K is the strike price of an option. The boundary condition at $S = 0$ is

$$\frac{\partial V}{\partial \tau} = \frac{-A}{T - \tau} \frac{\partial V}{\partial A} - rV.$$

For boundary condition at $S \rightarrow \infty$, we make the standard assumption that $\frac{\partial^2 V}{\partial S^2} \rightarrow 0$. The equation is discretized using an implicit finite volume method.

In each time step, one needs to solve the implicit equation corresponding to the convection-diffusion equation (4.1). It is important to note that there is no diffusion in the A direction, which is the source of many numerical difficulties. It is a particularly interesting test case for multigrid since it is very convection dominated (in fact, pure hyperbolic) in the A direction and the convection is not constant. We apply our multigrid of section 2.2.2 with point GS smoothing to solve the zero strike call option (i.e., $K = 0$), a case where the analytic solution is known. We use the same parameters as in [11], where $r = 0.1$ and the initial stock price is $S_0 = 100$. The mesh size $h = 1/64$ and the time step size $\Delta\tau = 0.025$. The stopping criterion is 10^{-6} .

TABLE 4.6

Analytical and numerical solutions for an Asian call option with zero strike and the average number of multigrid $V(2,2)$ -cycles over all time steps for solving the Asian option model equation.

σ	T	Analytic	Numerical	EMG(GS)
0.1	0.25	98.763	98.7604	12
0.1	0.50	97.547	97.5411	9
0.1	1.00	95.175	95.1626	6
0.2	0.25	98.763	98.7574	12
0.2	0.50	97.547	97.5309	9
0.2	1.00	95.175	95.1785	6
0.4	0.25	98.763	98.7673	12
0.4	0.50	97.547	97.5447	9
0.4	1.00	95.175	95.1633	6

Table 4.6 shows the numerical results with different practical values of σ , leading to different strengths of diffusion. The numerical solutions are in good agreement with the analytic solutions. Moreover, our EMG method converges between 6 and 12 multigrid iterations on average in each time step. More precisely, the method consistently converges in about 3 iterations when τ is small and in about 16 iterations when τ approaches the final time T (strong convection). Thus, the average iteration numbers tend to be larger for T small and smaller for T large.

5. Conclusions. We have demonstrated that Petrov–Galerkin coarse grid correction together with linear interpolation and kernel preserving restriction will lead to a fast convergent multigrid method for solving convection-diffusion equations. The main reason is that the coarse grid correction process is accurate and stable. Although the coarse grid matrix has a phase error of $1/2$ which shifts waves of any frequency by one grid point, the kernel preserving restriction has the same effect with opposite sign that counteracts the shifting caused by the coarse grid matrix. As a result, the coarse grid correction is accurate. Moreover, we have proved that the Petrov–Galerkin coarse grid matrix is stable in the sense that it is asymptotically an M-matrix for large β .

We have also considered the general case where $\beta(x)$ is not given explicitly and proposed a joint minimization formulation to construct the restriction operator. The convergence rate of the resulting multigrid method has been shown to be comparable to that given by exponential fitting restriction. However, we note that one needs to solve a minimization problem which may be costly. The algorithms are implemented in MATLAB, so accurate timings for constructing the restriction operator are not available. However, we refer the interested reader to [26] for more discussions on the cost of the minimization problem. This general approach is purely algebraic in nature and thus applicable to complicated geometries such as unstructured grids, but these cases are not discussed in the present paper.

REFERENCES

- [1] R. E. BANK, W. M. COUGHRAN, AND L. C. COWSAR, *The finite volume Scharfetter-Gummel method for steady convection diffusion equations*, Comput. Vis. Sci., 1 (1998), pp. 123–136.
- [2] R. E. BANK, T. F. DUPONT, AND H. YSERENTANT, *The hierarchical basis multigrid method*, Numer. Math., 52 (1988), pp. 427–458.
- [3] R. E. BANK AND S. GUTSCH, *Hierarchical basis for the convection-diffusion equation on unstructured meshes*, in Proceedings of the Ninth International Conference on Domain Decomposition Methods, P. Bjorstad, M. Espedal, and D. Keyes, eds., John Wiley, New York, 1998, pp. 251–265.
- [4] J. BEY AND G. WITTUM, *Downwind numbering: Robust multigrid for convection-diffusion problems*, Appl. Numer. Math., 23 (1997), pp. 177–192.
- [5] A. BRANDT, *Multi-level adaptive solutions to boundary-value problems*, Math. Comp., 31 (1977), pp. 333–390.
- [6] A. BRANDT, *Multigrid Solvers for Non-elliptic and Singular-Perturbation Steady-State Problems*, Department of Applied Mathematics, The Weizmann Institute of Science, Rehovot, Israel, 1981.
- [7] A. BRANDT AND I. YAVNEH, *On multigrid solution of high-Reynolds incompressible entering flows*, J. Comput. Phys., 101 (1992), pp. 151–164.
- [8] A. BRANDT AND I. YAVNEH, *Accelerated multigrid convergence and high-Reynolds recirculating flows*, SIAM J. Sci. Comput., 14 (1993), pp. 607–626.
- [9] J. E. DENDY, JR., *Black box multigrid*, J. Comput. Phys., 48 (1982), pp. 366–386.
- [10] J. E. DENDY, JR., *Black box multigrid for nonsymmetric problems*, Appl. Math. Comput., 13 (1983), pp. 261–283.
- [11] Y. D’HALLUIN, P. A. FORSYTH, AND G. LABAHN, *A semi-Lagrangian approach for American Asian options under jump diffusion*, SIAM J. Sci. Comput., 27 (2005), pp. 315–345.
- [12] W. HACKBUSCH, *Multi-grid Methods and Applications*, Springer-Verlag, Berlin, 1985.
- [13] W. HACKBUSCH AND T. PROBST, *Downwind Gauss-Seidel smoothing for convection dominated problems*, Numer. Linear Algebra Appl., 4 (1997), pp. 85–102.
- [14] T. HUGHES AND A. BROOKS, *Streamline-upwind Petrov-Galerkin formulations for convection dominated flows with particular emphasis on the incompressible Navier-Stokes equations*, Methods Appl. Mech. Engrg., 32 (1982), pp. 199–259.
- [15] A. JAMESON, *Solution of the Euler equations for two dimensional transonic flow by a multigrid method*, Appl. Math. Comput., 13 (1983), pp. 327–355.
- [16] A. JAMESON, *Computational transonics*, Comm. Pure Appl. Math., 41 (1988), pp. 507–549.
- [17] D. C. JESPERSEN, *A Time-Accurate Multiple-Grid Algorithm*, Paper 85-1493-CP, Amer. Inst. of Aeronautics and Astronautics, Reston, VA, 1985.
- [18] K. JOHANNSSEN, *Robust smoothers for convection-diffusion problems with closed characteristics*, Computing, 65 (2000), pp. 203–225.
- [19] R. LEVEQUE, *Numerical Methods for Conservation Laws*, 2nd ed., Birkhäuser Verlag, Basel, 1992.
- [20] K. W. MORTON, *Numerical Solution of Convection-Diffusion Problems*, Chapman and Hall, London, 1996.
- [21] R. H. NI, *A multiple-grid scheme for solving the Euler equations*, AIAA J., 20 (1982), pp. 1565–1571.
- [22] C. W. OOSTERLEE, F. J. GASPAR, T. WASHIO, AND R. WIENANDS, *Multigrid line smoothers for higher order upwind discretization of convection-dominated problems*, J. Comput. Phys., 139 (1998), pp. 274–307.
- [23] A. REUSKEN, *Multigrid with matrix-dependent transfer operators for a singular perturbation problem*, Computing, 50 (1993), pp. 199–211.
- [24] U. TROTTEMBERG, C. OOSTERLEE, AND A. SCHÜLLER, *Multigrid*, Academic Press, New York, 2001.
- [25] W. L. WAN AND T. F. CHAN, *A phase error analysis of multigrid methods for hyperbolic equations*, SIAM J. Sci. Comput., 25 (2003), pp. 857–880.
- [26] W. L. WAN, T. F. CHAN, AND B. SMITH, *An energy-minimizing interpolation for robust multigrid methods*, SIAM J. Sci. Comput., 21 (2000), pp. 1632–1649.
- [27] F. WANG AND J. XU, *A crosswind block iterative method for convection-dominated problems*, SIAM J. Sci. Comput., 21 (1999), pp. 620–645.
- [28] P. WESSELING, *An Introduction to Multigrid Methods*, John Wiley, Chichester, 1992.
- [29] G. WITTUM, *On the robustness of ILU smoothing*, SIAM J. Sci. Stat. Comput., 10 (1989), pp. 699–717.

- [30] I. YAVNEH, *Coarse-grid correction for nonelliptic and singular perturbation problems*, SIAM J. Sci. Comput., 19 (1998), pp. 1682–1699.
- [31] I. YAVNEH, C. H. VENNER, AND A. BRANDT, *Fast multigrid solution of the advection problem with closed characteristics*, SIAM J. Sci. Comput., 19 (1998), pp. 111–125.
- [32] P. M. D. ZEEUW, *Matrix-dependent prolongations and restrictions in a blackbox multigrid solver*, J. Comput. Appl. Math., 33 (1990), pp. 1–27.
- [33] P. M. D. ZEEUW AND E. J. VAN ASSELT, *The convergence rate of multi-level algorithms applied to the convection-diffusion equation*, SIAM J. Sci. Stat. Comput., 6 (1985), pp. 492–503.
- [34] R. ZVAN, P. A. FORSYTH, AND K. R. VETZAL, *Robust numerical methods for PDE models of Asian options*, J. Comp. Finance, 1 (1998), pp. 39–78.

$\mathcal{H}$ -MATRIX PRECONDITIONERS IN CONVECTION-DOMINATED PROBLEMS*

SABINE LE BORNE[†] AND LARS GRASEDYCK[‡]

Abstract. Hierarchical matrices provide a data-sparse way to approximate fully populated matrices. In this paper we exploit $\mathcal{H}$ -matrix techniques to approximate the LU -decompositions of stiffness matrices as they appear in (finite element or finite difference) discretizations of convection-dominated elliptic partial differential equations. These sparse $\mathcal{H}$ -matrix approximations may then be used as preconditioners in iterative methods. Whereas the approximation of the matrix inverse by an $\mathcal{H}$ -matrix requires some modification in the underlying index clustering when applied to convection-dominant problems, the $\mathcal{H}$ - LU -decomposition works well in the standard $\mathcal{H}$ -matrix setting even in the convection dominant case. We will complement our theoretical analysis with some numerical examples.

Key words. hierarchical matrices, data-sparse approximation, preconditioning, convection-dominant problems

AMS subject classifications. 65F05, 65F30, 65F50

DOI. 10.1137/040615845

1. Introduction. In a series of papers, the technique of hierarchical matrices, or $\mathcal{H}$ -matrices for short, has been introduced [3, 4, 6, 7, 8]. $\mathcal{H}$ -matrices provide an inexpensive but sufficiently accurate approximation to fully populated matrices as they appear in boundary element methods or finite element methods. In finite element methods, it is the inverse of the stiffness matrix as well as the matrices L and U in an LU -decomposition, which are typically fully populated.

An $\mathcal{H}$ -matrix approximation is based on a certain hierarchical block structure of the matrix in which some (preferably large) off-diagonal blocks are represented by low-rank approximations. Such a block structure and the resulting approximations for (inverse) matrices arising in the discretization of uniformly elliptic PDEs can be computed and stored in almost linear complexity, i.e., $\mathcal{O}(n \log_2^\alpha n)$ with moderate parameter α [1]. The constants in these complexity estimates depend on the structure of the involved $\mathcal{H}$ -matrices and have been computed exactly in [4].

In a previous paper [11], the approximation quality for standard $\mathcal{H}$ -matrices was analyzed when applied to the inverse matrices that stem from convection-dominated problems with constant convection. Some modifications were derived in which the direction and magnitude of the convection influence the block structure of the resulting $\mathcal{H}$ -matrix, leading to significant improvements in the approximation by an $\mathcal{H}$ -matrix compared to the standard setting. In the case of a *nonconstant* dominant convection,

*Received by the editors September 27, 2004; accepted for publication (in revised form) by Y. Saad May 23, 2005; published electronically March 17, 2006. This work was performed by an employee of the U.S. Government or under U.S. contract. The U.S. Government retains a nonexclusive, royalty-free license to publish or reproduce the published form of this contribution, or allow others to do so, for U.S. Government purposes. Copyright is owned by SIAM to the extent not limited by these rights.

<http://www.siam.org/journals/simax/27-4/61584.html>

[†]Department of Mathematics, Box 5054, Tennessee Technological University, Cookeville, TN 38505 (sleborne@tntech.edu). A part of this material is based upon work supported by the National Science Foundation under grant DMS-0408950.

[‡]Max-Planck-Institute for Mathematics in the Sciences, Inselstr. 22-26, Leipzig 04103, Germany (lgr@mis.mpg.de).

the standard $\mathcal{H}$ -matrix turns out to provide a satisfactory approximation, due to the numerical diffusion induced by the discretization [12].

While an approximate $\mathcal{H}$ -matrix inverse provides a robust preconditioner, its construction is rather expensive due to the large constants in the complexity estimates. Here, we will pursue the construction of a preconditioner based upon an approximate $\mathcal{H}$ - LU factorization of the matrix, which can be obtained much faster. In [13], an analysis is provided for the $\mathcal{H}$ -Cholesky factorization of the Laplace equation in two spatial dimensions.

In this paper, after an introduction to $\mathcal{H}$ -matrices in section 2, we will use $\mathcal{H}$ -matrices to approximate the matrix factors in an LU -decomposition of the stiffness matrix. In section 3, we will describe the recursive construction of an LU -decomposition using $\mathcal{H}$ -matrix arithmetic, and then we focus on $\mathcal{H}$ - LU -decompositions for the stiffness matrix of the convection-dominant convection-diffusion equation, for both constant and nonconstant convection directions. We will use the $\mathcal{H}$ -matrix approximations of the LU -decomposition as preconditioners in Krylov-accelerated iterative methods. In section 4, we will provide some numerical tests to illustrate the efficiency of the resulting method.

2. A brief introduction to $\mathcal{H}$ -matrices. An $\mathcal{H}$ -matrix approximation to a given (dense) matrix is obtained by replacing certain blocks of the matrix by matrices of a low rank k , stored in so-called Rk -format, as will be further explained below. Given such an $\mathcal{H}$ -matrix, the standard matrix operations such as matrix-vector multiplication, matrix-matrix addition and multiplication, and (approximate) matrix inversion can be defined for this $\mathcal{H}$ -matrix format. The construction of $\mathcal{H}$ -matrices is reviewed in subsection 2.1, and their arithmetic is reviewed in subsection 2.2.

2.1. Construction of $\mathcal{H}$ -matrices. The formal definition of an $\mathcal{H}$ -matrix depends on appropriate hierarchical partitionings of the index set and also of the product index set, which are organized in (block) cluster trees as defined next. Instead of fixed partitionings, these trees will provide hierarchies of partitionings, which gives a hierarchical matrix its name.

DEFINITION 2.1 (cluster tree). *Let I be a finite index set, and let $T_I = (V, E)$ be a tree with vertex set V and edge set E . For a vertex $v \in V$ we define the set of sons of v as $S(v) := \{w \in V \mid (v, w) \in E\}$. The tree T_I is called a cluster tree of I if its vertices consist of subsets of I and satisfy the following conditions:*

1. $I \in V$ is the root of T_I and $v \subset I$, $v \neq \emptyset$, for all $v \in V$.
2. For all $v \in V$ there either holds $S(v) = \emptyset$ or $v = \bigcup_{w \in S(v)} w$.

In the following we identify V and T_I ; i.e., we write $v \in T_I$ instead of $v \in V$. The nodes $v \in V$ are called clusters.

For regular grids one can construct the cluster tree T_I in a cardinality balanced way; i.e., an index cluster is divided into a certain number of sons of approximately the same size with respect to the number of indices (see [6, 8]). In this paper we restrict our attention to the case of two (or no) sons per cluster, which is the easiest with respect to analysis and implementation, and we are not aware of any disadvantages compared to an arbitrary number of sons. For locally refined grids, the results from [4] indicate that the cardinality balanced clustering is *not* optimal with respect to the complexity of adaptive $\mathcal{H}$ -matrix updates. Instead, one can use a geometrically balanced approach as described in [5], which is used in the remainder of this paper, including the numerical tests. In the case of regular grids, however, both approaches yield very similar if not identical cluster trees.

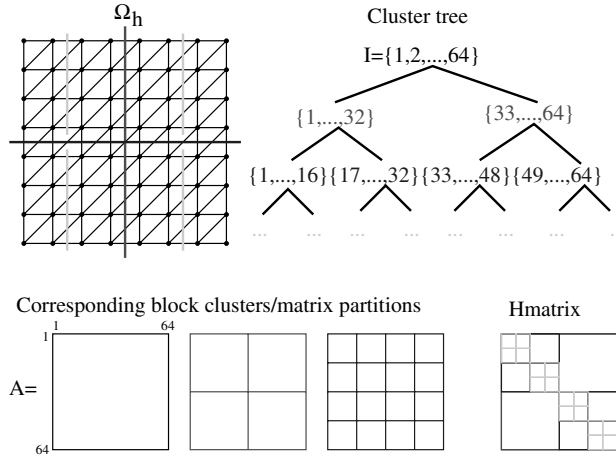


FIG. 2.1. A mesh Ω_h (top left), a corresponding cluster tree (top right), resulting hierarchy of matrix partitions (bottom left), and an example of an $\mathcal{H}$ -matrix (bottom right).

A simple example for the construction of a cluster tree is given in Figure 2.1. In this example, the geometrical and cardinality balanced approaches result in the same cluster tree. Here, the regular grid Ω_h contains 64 vertices, i.e., $I = \{1, \dots, 64\}$ (assuming continuous piecewise linear elements). The index set is subdivided into two subsets of size 32 each along the middle vertical line. The resulting subsets are both subdivided horizontally, resulting in a total of four subsets of size 16. This process is continued until 64 subsets (or clusters) of size 1 are obtained.

DEFINITION 2.2 (leaf, father, level, depth). Let T_I be a cluster tree. The set of leaves of the tree T_I is $\mathcal{L}(T_I) = \{v \in T_I \mid S(v) = \emptyset\}$. The uniquely determined predecessor (father) of a nonroot vertex $v \in T_I$ is denoted by $\mathcal{F}(v)$. The levels of the tree T_I are defined by

$$T_I^{(0)} := \{I\}, \quad T_I^{(\ell)} := \{v \in T_I \mid \mathcal{F}(v) \in T_I^{(\ell-1)}\} \quad \text{for } \ell \in \mathbb{N},$$

and we write $\text{level}(v) = \ell$ if $v \in T_I^{(\ell)}$. The depth of T is defined as $d(T) := \max\{\ell \in \mathbb{N}_0 \mid T_I^{(\ell)} \neq \emptyset\}$.

For any cluster tree T_I there holds $I = \dot{\bigcup} \{v \mid v \in \mathcal{L}(T_I)\}$. A hierarchy of block partitionings of the product index set $I \times I$ is based upon a cluster tree T_I and is organized in a block cluster tree, as follows.

DEFINITION 2.3 (block cluster tree). Let T_I be a cluster tree of the index set I . A cluster tree $T_{I \times I}$ is called a block cluster tree (based upon T_I) if for all $v \in T_{I \times I}^{(1)}$ there exist $t, s \in T_I^{(1)}$ such that $v = t \times s$. The nodes $v \in T_{I \times I}$ are called block clusters.

Analogously to the cluster tree, for any block cluster tree $T_{I \times I}$ there holds $I \times I = \dot{\bigcup} \{v \mid v \in \mathcal{L}(T_{I \times I})\}$; i.e., the leaves of the block cluster tree provide a disjoint block partition of the product index set $I \times I$.

The objective is to construct a block cluster tree from a given cluster tree such that the leaves (of the block cluster tree) correspond to (preferably large) matrix blocks with “smooth” data that can be approximated by low rank matrices in the following Rk-matrix representation.

DEFINITION 2.4 (Rk-matrix representation). Let $k, n, m \in \mathbb{N}_0$. Let $M \in \mathbb{R}^{n \times m}$

be a matrix of at most rank k . A representation of M in factorized form,

$$(2.1) \quad M = AB^T, \quad A \in \mathbb{R}^{n \times k}, B \in \mathbb{R}^{m \times k},$$

with A and B stored in full matrix representation, is called an *Rk-matrix representation* of M , or, in short, we call M an *Rk-matrix*.

If the rank k is small compared to the matrix size given by n and m , we obtain considerable savings in the storage and work complexities of an Rk-matrix compared to a full matrix [4].

In the following construction, we will build a block cluster tree iteratively starting from $I \times I$ and refining the block clusters if they do not satisfy a certain admissibility condition. The choice of the admissibility condition depends on the underlying continuous problem (i.e., the elliptic partial differential equation, in particular its associated Green's function) and shall ensure that all admissible blocks allow a sufficiently accurate Rk-approximation. A typical admissibility condition for uniformly elliptic problems is as follows:

$$(2.2) \quad \text{block cluster } s \times t \text{ is admissible} \Leftrightarrow \min(\text{diam}(s), \text{diam}(t)) \leq \eta \text{ dist}(s, t).$$

Here, “diam” and “dist” denote the Euclidean diameter/distance of the (union of the) supports of the basis functions with indices in s, t , respectively. A given cluster tree along with an admissibility condition allows the following canonical construction of a block cluster tree:

Let the cluster tree T_I be given. We define the block cluster tree $T_{I \times I}$ by $\text{root}(T) := I \times I$, and each vertex $s \times t \in T$ has the set of successors

$$(2.3) \quad S(s \times t) := \begin{cases} \emptyset & \text{if } s \times t \text{ admissible,} \\ \emptyset & \text{if } \min\{\#t, \#s\} \leq n_{\min}, \\ \{s' \times t' \mid s' \in S(s), t' \in S(t)\} & \text{otherwise.} \end{cases}$$

The parameter $n_{\min}$ ensures that blocks do not become too small where the matrix arithmetic of a full matrix is more efficient than any further subdivision. It is typically set to $n_{\min} = 32$ or even $n_{\min} = 64$. The leaves of a block cluster tree obtained through this construction will be used in the following definition of an $\mathcal{H}$ -matrix.

DEFINITION 2.5 ($\mathcal{H}$ -matrix). *Let $k, n_{\min} \in \mathbb{N}_0$. The set of $\mathcal{H}$ -matrices induced by a block cluster tree $T := T_{I \times I}$ with blockwise rank k and minimum block size $n_{\min}$ is defined by*

$$\mathcal{H}(T, k) := \{M \in \mathbb{R}^{I \times I} \mid \forall t \times s \in \mathcal{L}(T) : \text{rank}(M|_{t \times s}) \leq k \text{ or } \min\{\#t, \#s\} \leq n_{\min}\}.$$

A matrix $M \in \mathcal{H}(T, k)$ is said to be given in $\mathcal{H}$ -matrix representation if the blocks $M|_{t \times s}$ with $\text{rank}(M|_{t \times s}) \leq k$ are in Rk-matrix representation and the remaining blocks with $\min\{\#t, \#s\} \leq n_{\min}$ are stored as full matrices. The set of indices $(i, j) \in I \times I$ that belong to Rk-matrix blocks is called the *farfield*, while the complement is called the *nearfield*.

Whereas the classical $\mathcal{H}$ -matrix uses a fixed rank for the Rk-blocks, it is possible to replace it by *variable (or adaptive) ranks* in order to enforce a desired accuracy within the individual blocks. In particular, for a given admissible block $s \times t$, we set the rank k of the corresponding matrix block $M|_{s \times t}$ as follows:

$$(2.4) \quad k(M|_{s \times t}) := \min\{k' \mid \sigma_{k'} \leq \delta \sigma_1\},$$

where σ_i denotes the i th largest singular value and $0 < \delta < 1$ denotes the desired relative accuracy within each block. We will demonstrate the performance of both approaches in the numerical tests in section 4.

An example for the block structure of an $\mathcal{H}$ -matrix is shown in Figure 2.1 (bottom right). Both the accuracy and (storage) complexity of an $\mathcal{H}$ -matrix approximation to a given matrix depend on the construction of an appropriate cluster tree, i.e., a hierarchy of index set partitionings. Details regarding approximation errors for blocks that satisfy the admissibility condition, i.e., for blocks that have a large distance compared to their diameters, as well as storage requirements for full, Rk-, and $\mathcal{H}$ -matrices are given in [4]. The intuitive objective in the construction of a cluster tree, given the standard admissibility condition (2.2), is to partition the index set into well separated clusters of vertices that are geometrically close to each other. As a result, relatively large blocks become admissible and we obtain an accurate $\mathcal{H}$ -matrix approximation that is inexpensive to store.

2.2. Arithmetic of $\mathcal{H}$ -matrices. Given two $\mathcal{H}$ -matrices $A, B \in \mathcal{H}(T, k)$ based on the same block cluster tree T , i.e., with the same block structure, the exact sum or product of these two matrices will typically not belong to $\mathcal{H}(T, k)$. In the case of matrix addition, we have $A + B \in \mathcal{H}(T, 2k)$; the rank of an exact matrix product is less obvious. We will use a truncation operator $\mathcal{T}_{k \leftarrow k'}^{\mathcal{H}}$ to define the $\mathcal{H}$ -matrix addition $C := A \oplus_{\mathcal{H}} B$ and $\mathcal{H}$ -matrix multiplication $C := A \otimes_{\mathcal{H}} B$, where again $C \in \mathcal{H}(T, k)$.

A truncation of a rank k matrix R to rank $k' < k$ is defined as the best approximation with respect to the Frobenius (or spectral) norm in the set of rank k' matrices. In the context of $\mathcal{H}$ -matrices, we use such truncations for all admissible (rank k) blocks. Using truncated versions of the QR -decomposition and singular value decomposition, the truncation of a rank k matrix $R \in \mathbb{R}^{n,m}$ (given in the form $R = AB^T$, where $A \in \mathbb{R}^{n,k}$ and $B \in \mathbb{R}^{m,k}$) to a lower rank can be computed with complexity $\mathcal{O}(5k^2(n+m) + 23k^3)$; further details are provided in [4].

We can now define the $\mathcal{H}$ -matrix addition and multiplication as follows:

$$\begin{aligned} A \oplus_{\mathcal{H}} B &= \mathcal{T}_{k \leftarrow 2k}^{\mathcal{H}}(A + B), \\ A \otimes_{\mathcal{H}} B &= \mathcal{T}_{k \leftarrow k'}^{\mathcal{H}}(A * B), \end{aligned}$$

where $k' \leq c(p+1)k$ is the rank of the exact matrix product, c denotes some constant (which depends on the block cluster tree T), and p denotes the depth of the tree (see Definition 2.2). Estimates show that the $\mathcal{H}$ -matrix addition and multiplication have almost optimal complexity and are provided in [4] along with efficient implementations of these operations.

3. $\mathcal{H}$ -matrix approximations to the LU -decomposition. Typically, the factors L and U in an exact LU -decomposition of a sparse matrix A suffer from significant fill-in, as illustrated in Figure 3.1. Here, the stiffness matrix A (left) and its LU -factors (middle) are presented where the matrix A results from an upwind finite element discretization of the convection-diffusion equation $-\Delta u + u_x = f$ with Dirichlet boundary conditions on a regular triangulation of the square $\Omega = [-1, 1] \times [-1, 1]$. In this example, the problem size has been set to $n = 10000$. A count of the nonzero entries in the LU -decomposition for various problem sizes yields a nonoptimal complexity of $\mathcal{O}(n^{\frac{3}{2}})$ for the number of nonzero entries. The ordering of the unknowns results from the clustering process in the $\mathcal{H}$ -matrix construction. It is well known that the ordering of the unknowns affects the sparsity pattern of the LU -decomposition as well as the approximation quality of an incomplete LU -decomposition (ILU).

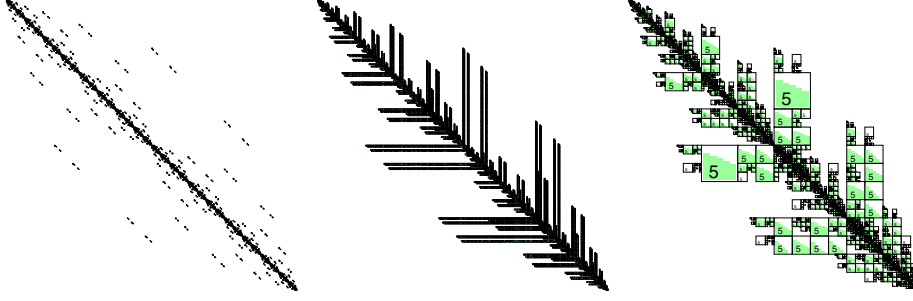


FIG. 3.1. Sparsity pattern of a typical stiffness matrix A (left), its LU -factors (middle), and an $\mathcal{H}$ -matrix approximation to its LU -factors (right).

Figure 3.1 (right) shows an $\mathcal{H}$ -matrix approximation to the LU -factors. Here, the admissibility condition (2.2) has been applied with $\eta = 4$. Admissible off-diagonal blocks are represented by Rk -matrices with ranks at most five. For the larger off-diagonal blocks, one can observe the decay of the first five singular values on a logarithmic scale; i.e., they decrease exponentially since their logarithms appear to decrease linearly.

3.1. Construction of the $\mathcal{H}$ -matrix- LU -decomposition. The computation of $\mathcal{H}$ -matrix approximations to the LU -factors is done recursively (with respect to the block clustering) and is of complexity $\mathcal{O}(n \log^2 nk^2)$, where n denotes the problem size and k denotes the maximum rank of the Rk -blocks. It requires a triangular solve $TX = B$ for an unknown $\mathcal{H}$ -matrix X with a given (upper or lower) triangular $\mathcal{H}$ -matrix T and a given right-hand-side $\mathcal{H}$ -matrix B , which we will describe first. We distinguish between three types of matrix blocks X and B :

1. Rk -blocks (which correspond to admissible block clusters),
2. full blocks (which correspond to nonadmissible, small blocks of size $n \leq n_{\min}$), and
3. refined blocks (which are further subdivided).

In the first case of Rk -blocks X and B , we will use the Rk -representation $B = Y_1 Y_2^T$ in order to compute the matrices X_1 and X_2 in $X = X_1 X_2^T$ by

$$X_2 := Y_2, \quad X_1 = T^{-1} Y_1.$$

Since T is assumed to be triangular and X_1 is assumed to have k columns, the computation of X_1 requires k backward (or forward) solves with T .

In the second case of full blocks, we define $X := T^{-1} B$, where we use the standard (dense) matrix arithmetic. For the third case of refined blocks, we assume a 2×2 block structure of the matrix which is induced by the hierarchy of index partitionings. We assume that the triangular solve is already defined on all coarser levels (which is granted since it is defined on the coarsest levels by 1 and 2 above). We will solve for the unknown blocks X_{ij} in

$$\begin{pmatrix} L_{11} & \\ L_{21} & L_{22} \end{pmatrix} \begin{pmatrix} X_{11} & X_{12} \\ X_{21} & X_{22} \end{pmatrix} = \begin{pmatrix} B_{11} & B_{12} \\ B_{21} & B_{22} \end{pmatrix}$$

in the following four steps:

1. Solve $L_{11} X_{11} = B_{11}$ for X_{11} by a triangular solve (with L_{11}) on the next coarser level.

2. Solve $L_{11}X_{12} = B_{12}$ for X_{12} by a triangular solve (with L_{11}) on the next coarser level.
3. Solve $L_{22}X_{21} = B_{21} - L_{21}X_{11}$ for X_{21} by a triangular solve (with L_{22}) on the next coarser level.
4. Solve $L_{22}X_{22} = B_{22} - L_{21}X_{12}$ for X_{22} by a triangular solve (with L_{22}) on the next coarser level.

Any matrix arithmetic in the previous four steps will be replaced by (approximate) $\mathcal{H}$ -matrix arithmetic $\oplus_{\mathcal{H}}$, $\ominus_{\mathcal{H}}$, and $\otimes_{\mathcal{H}}$ (see section 2.2), which will ensure almost optimal complexity for the computational cost as well as storage for the resulting matrix blocks X_{ij} . The triangular solve involving an upper triangular matrix follows analogously.

To compute an $\mathcal{H}$ -LU-decomposition $A = LU$, we use a standard (dense) LU-decomposition $A = LU$ on the coarsest level (where $n \leq n_{\min}$), and we assume that the $\mathcal{H}$ -LU-decomposition is defined already on all coarser levels. We will solve for the unknown blocks L_{ij}, U_{ij} in

$$\begin{pmatrix} A_{11} & A_{12} \\ A_{21} & A_{22} \end{pmatrix} = \begin{pmatrix} L_{11} & \\ L_{21} & L_{22} \end{pmatrix} \begin{pmatrix} U_{11} & U_{12} \\ & U_{22} \end{pmatrix}$$

in the following four steps:

1. Solve $A_{11} = L_{11}U_{11}$ for L_{11}, U_{11} by an $\mathcal{H}$ -LU-decomposition on the next coarser level.
2. Solve $A_{12} = L_{11}U_{12}$ for U_{12} by a triangular solve (with L_{11}).
3. Solve $A_{21} = L_{21}U_{11}$ for L_{21} by a triangular solve (with U_{11}).
4. Solve $L_{22}U_{22} = A_{22} - L_{21}U_{12}$ for L_{22}, U_{22} by an $\mathcal{H}$ -LU-decomposition on the next coarser level.

Again, any matrix arithmetic will be replaced by (approximate) $\mathcal{H}$ -matrix arithmetic $\oplus_{\mathcal{H}}$, $\ominus_{\mathcal{H}}$, and $\otimes_{\mathcal{H}}$ (see section 2.2), which will ensure almost optimal complexity for the computational cost as well as storage for the resulting matrix blocks L_{ij}, U_{ij} .

In the case of a regular triangulation of the unit square and an $\mathcal{H}$ -matrix construction using a regular quadrosection in the cluster tree construction, there occur only eight different block formats, and the recursions for the triangular solve and the LU-factorization can be described explicitly for these formats [13].

3.2. The case of constant convection. In this section, we consider the model convection-diffusion equation with constant convection

$$(3.1) \quad -\epsilon \Delta u + \begin{pmatrix} b_1 \\ b_2 \end{pmatrix} \cdot \nabla u = f \quad \text{in } \Omega,$$

$$(3.2) \quad u = g \quad \text{on } \Gamma := \partial\Omega,$$

with $\epsilon \in (0, 1)$ and constant $b_1, b_2 \in \mathbb{R}$.

In the case of dominant convection (i.e., $\epsilon \ll 1$), the nonzero entries in the stiffness matrix obtained through some upwind discretization typically differ significantly in their magnitudes, and it is helpful to distinguish between large and small matrix entries. One may neglect the small matrix entries by replacing them with zeros and then order the unknowns in a *downwind* manner such that one obtains an upper triangular matrix which can be used as a preconditioner [2, 10, 9]. Such an ordering is not determined uniquely, and the following Algorithm 3.1 produces such an ordering using the standard $\mathcal{H}$ -matrix construction with geometric partitions. The key ingredient here is the ordering of the cluster sons according to the convection direction.

This, in turn, will ensure that the quality of the $\mathcal{H}$ - LU -decomposition will improve as $\epsilon \rightarrow 0$ since the stiffness matrix tends towards an upper triangular matrix.

ALGORITHM 3.1. Let $(b_1, b_2)^T$ denote the constant convection direction in (3.1). We assume an upwind discretization. Let T_I be a geometrically balanced cluster tree; i.e., a cluster r is subdivided into two successors s and t by “cutting” along either a vertical or horizontal line (see Figure 2.1). Let (x_i, y_i) be the point associated with an index i . If s, t result from a vertical cut such that $x_i < x_j$ for all indices $i \in s$, $j \in t$, then we will order the indices in s before those in t if $b_1 \geq 0$. If $b_1 < 0$, we will order indices in t before those in s . Correspondingly, if s, t result from a horizontal cut such that $y_i < y_j$ for all indices $i \in s$, $j \in t$, then we will order the indices in s before those in t if $b_2 \geq 0$. If $b_2 < 0$, we will order indices in t before those in s .

Remark 3.2. The $\mathcal{H}$ -matrix that results from Algorithm 3.1 will have all “large” entries in the lower triangular part, i.e., $|a_{ij}| > |a_{ji}|$ if $j < i$. As $\epsilon \rightarrow 0$ in (3.1), the stiffness matrix tends to be a lower triangular matrix.

We point out that the above Remark 3.2 establishes a significant difference between the computation of $\mathcal{H}$ - LU -factors and an approximate $\mathcal{H}$ -matrix inverse: In [11], it is shown that the computation of a convergent approximate $\mathcal{H}$ -matrix inverse in the case of dominant constant convection requires the adjustment of the clustering as well as the admissibility condition to the convection direction and its magnitude (in particular, Algorithm 3.1 cannot be used successfully for the $\mathcal{H}$ -matrix inversion). Remark 3.2 above states that such adjustments are not required for the computation of $\mathcal{H}$ - LU -factors. This difference results from the fact that an $\mathcal{H}$ -inverse approximates the inverse A^{-1} , whereas an $\mathcal{H}$ - LU factorization approximates the LU -factors of an LU -factorization $A = LU$.

3.3. The case of arbitrary convection. In this section, we consider the model convection-diffusion equation with arbitrary convection

$$\begin{aligned} (3.3) \quad & -\epsilon \Delta u + \mathbf{b} \cdot \nabla u = f \quad \text{in } \Omega, \\ (3.4) \quad & u = g \quad \text{on } \Gamma := \partial\Omega, \end{aligned}$$

with $\epsilon \in (0, 1)$ and arbitrary (possibly cyclic) $\mathbf{b} = (b_1(x), b_2(x))^T$. If the convection is acyclic, one may still imitate downwind ordering strategies to obtain a mostly triangular matrix using the standard clustering algorithm. In the cyclic case, however, such an ordering does not exist, not even without the restriction of a standard clustering. Even though some convection-adapted clustering may seem reasonable, we found in our numerical results that the standard clustering still yields the best and very satisfactory results. One explanation may be the numerical viscosity of the underlying discretization.

4. Numerical results for $\mathcal{H}$ -matrix preconditioners. In this section we will demonstrate some numerical results using the $\mathcal{H}$ - LU -decomposition as a preconditioner. All tests were performed on a Dell Workstation 530 with a Xeon 2.4GHz processor using the standard $\mathcal{H}$ -matrix library HLIB (cf. <http://www.hlib.org>). Throughout all tests, we set the parameter $\eta = 4.0$ in the admissibility condition (2.2). However, we make a slight change in the representation of small, nonadmissible matrix blocks in order to reduce the storage of the resulting matrices considerably: Instead of an exact representation of a nonadmissible block of size smaller than $n_{min} = 32$ (i.e., a full matrix representation), we approximate these small blocks in Rk-representation with possibly larger rank than for the admissible blocks. This change is based on the observation that these blocks, even though they do not satisfy the admissibility

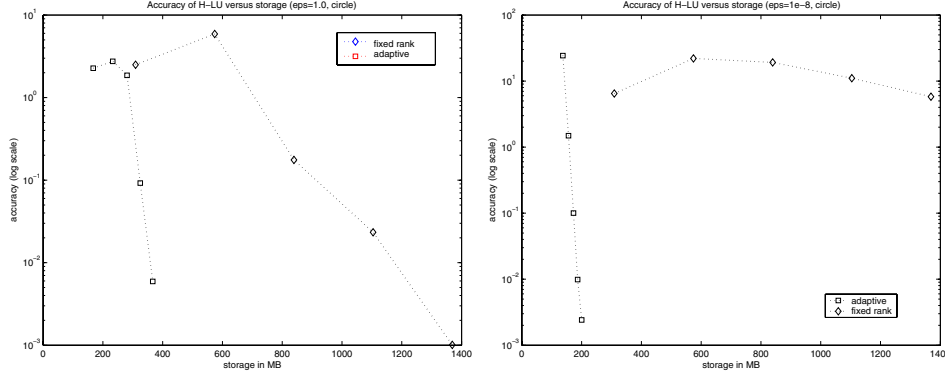


FIG. 4.1. Approximation accuracy versus required storage of LU -factors for fixed rank (diamonds) or adaptive rank (squares); left for $\epsilon = 1.0$ and right for convection-dominant case with $\epsilon = 1e - 8$.

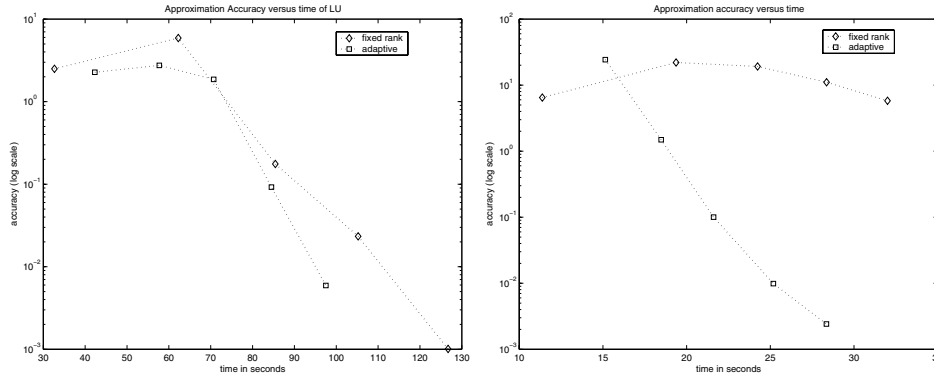


FIG. 4.2. Approximation accuracy versus required time to compute $\mathcal{H}$ - LU -factors for fixed rank (diamonds) or adaptive rank (squares); left for $\epsilon = 1.0$ and right for convection-dominant case with $\epsilon = 1e - 8$.

condition, typically are not of full rank. Whereas this approach leads to a considerable reduction in storage, it has the following disadvantage for small ranks (or low adaptive accuracies): The original stiffness matrix may no longer be expressed exactly as an $\mathcal{H}$ -matrix; i.e., the $\mathcal{H}$ - LU factorization will be computed for an $\mathcal{H}$ -matrix approximation of the stiffness matrix. This disadvantage, however, disappears when the rank (or adaptive accuracy) is increased.

Our test problem is the convection-diffusion equation (3.3), discretized on a regular triangulation of the square $\Omega = [-1, 1] \times [-1, 1]$ using Tabata's upwind triangle scheme [14, Chap. III, sec. 3.1.1]. We will provide numerical results for a constant convection $\mathbf{b}_{const}$ as well as a cyclic convection $\mathbf{b}_{cyclic}$ defined by

$$(4.1) \quad \mathbf{b}_{const}(x, y) = (1, 0)^T \quad \text{and} \quad \mathbf{b}_{cyclic}(x, y) = (0.5 - y, x - 0.5)^T.$$

In the first set of tests, the number of unknowns is $n = 160000$. We compare the required storage (Figure 4.1) and time (Figure 4.2) to obtain a certain approximation accuracy of the $\mathcal{H}$ - LU -factors when using fixed ranks (diamonds) in the Rk-blocks compared to adaptive ranks (squares). In the case of adaptive ranks, we choose the

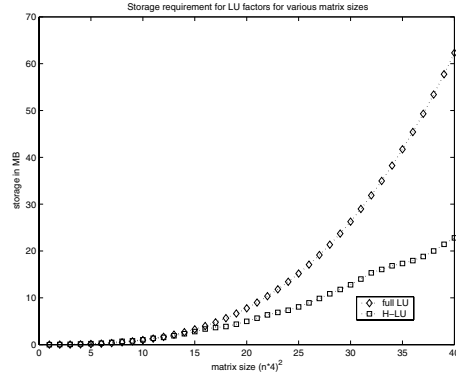


FIG. 4.3. Comparison of storage of exact and $\mathcal{H}$ -LU factors.

local ranks k such that $\sigma_k \leq \delta \sigma_1$, where σ_i denotes the i th largest singular value of the respective matrix block (see (2.4)), and $\delta = 10^{-j}$, $j = 1, \dots, 5$, for the results shown in Figures 4.1 and 4.2.

The results shown in Figure 4.1 were obtained for the convection-diffusion equation with cyclic convection $\mathbf{b}_{cyclic}$. The x -axis measures the required storage of the $\mathcal{H}$ -LU-factors in megabytes, whereas the y -axis shows the accuracy $\|(L^{\mathcal{H}}U^{\mathcal{H}})^{-1}A - I\|_2$ of the $\mathcal{H}$ -LU-factors on a logarithmic scale.

The left-hand plots in both Figures 4.1 and 4.2 show the results where $\epsilon = 1.0$ in the convection-diffusion equation (3.3), i.e., when the convection is only moderate. We notice that for both choices—adaptive or fixed ranks—the approximation accuracy improves exponentially as the used storage (Figure 4.1) and the required time to compute the $\mathcal{H}$ -LU-factors (Figure 4.2) increase (in view of the logarithmically scaled y -axis)—at least if we discard the very first entries in (most of) the curves. These initial spikes are explained by our modified representation of nonadmissible small blocks, as explained at the beginning of this section. It is apparent that a much faster convergence is obtained when adaptive ranks are used. The advantage of adaptive ranks over fixed ranks becomes even more apparent in the convection-dominant case: The plots on the right-hand sides of Figures 4.1 and 4.2 show the results where $\epsilon = 1e - 8$ in the convection-diffusion equation (3.3). The advantage of adaptive ranks can be explained by the nonsymmetric nature of the stiffness matrix with greatly varying magnitudes of matrix entries: Whereas some Rk-blocks require a large rank to obtain a certain accuracy, other blocks are close to zero blocks, and imposing a fixed rank onto these blocks will yield hardly any improvement in accuracy but still be costly with respect to storage and time.

In the case of *constant convection* $\mathbf{b}_{const}$, the results for $\epsilon = 1.0$ (diffusion-dominated) are nearly identical with those for cyclic convection. In the convection-dominated case $\epsilon = 10^{-8}$, both fixed and adaptive ranks lead to nearly exact solvers; both methods converge in only one iteration step for $rank = 1$ or $\delta = 10^{-1}$, respectively, with storage requirement for the LU-factors of 310 MB for the fixed rank and 110 MB for the adaptive rank.

In Figure 4.3 we compare the storage requirements of an exact LU-factorization (in diamonds) and the $\mathcal{H}$ -LU factors (in squares) for the Laplace matrix $A \in \mathbb{R}^{n,n}$ up to size $n = (4 \cdot 40)^2 = 25600$. The graphs confirm the theoretical estimates of $\mathcal{O}(n^{1.5})$ and $\mathcal{O}(n \log n)$ for the exact and $\mathcal{H}$ -LU factorization, respectively. In comparison, the

TABLE 4.1
 $\epsilon = 1.0$, *cyclic convection, adaptive rank.*

$n \backslash \text{accuracy}$	Time LU -decompose (in sec.)				Time solver (in sec.)/#steps			
	0.1	1e-2	1e-3	1e-4	0.1	1e-2	1e-3	1e-4
40000	8.0	10.9	13.3	15.5	1.9/14	0.7/4	0.4/2	0.4/2
80089	17.4	24.2	29.3	34.9	6.5/22	1.8/5	1.2/3	0.9/2
160000	42.2	57.9	70.4	83.6	19.3/31	4.4/6	2.5/3	1.9/2
320356	93.2	127.3	158.7	189.9	52.6/42	9.4/6	7.2/4	4.1/2
640000	218.8	313.5	387.4	431.9	157/59	33/10	19.1/5	8.5/2
1281424	472.9	705.9			422/80	79/12		

TABLE 4.2
 $\epsilon = 1e - 8$, *cyclic convection, adaptive rank.*

$n \backslash \text{accuracy}$	Time LU -decompose (in sec.)				Time solver (in sec.)/#steps			
	0.1	1e-2	1e-3	1e-4	0.1	1e-2	1e-3	1e-4
40000	3.1	3.6	4.2	4.6	0.5/4	0.3/2	0.3/2	0.2/1
80089	6.3	7.6	8.9	10.2	1.0/4	0.6/2	0.6/2	0.6/2
160000	14.8	18.2	21.5	24.7	2.6/5	1.7/3	1.2/2	0.7/1
320356	31.2	39.7	47.6	57.1	5.3/5	3.5/3	2.5/2	2.7/2
640000	72.1	95.2	117.5	138.2	15.5/7	7.4/3	5.4/2	5.7/2
1281424	158.4	214.3	270.1	332.6	36.4/8	15.6/3	11.5/2	12.2/2

sparse stiffness matrix for $n = 25600$ requires 2.13 MB for storage, and a full matrix of this size requires 904 MB.

Consequently, we will choose adaptive ranks in the following numerical tests, where we use the $\mathcal{H}$ - LU -factors as preconditioners in a bicgstab [15] iteration. As before, we choose a cyclic convection $\mathbf{b}_{cyclic}$ (4.1). Table 4.1 displays results for a moderate convection ($\epsilon = 1.0$ in (3.3)), and in Table 4.2 the same tests are repeated for a dominant convection ($\epsilon = 1e - 8$ in (3.3)). We record the times for the construction of the $\mathcal{H}$ - LU -factors for various problem sizes ($n = 40000, 80089, \dots, 1281424$) and accuracies δ responsible to determine the local ranks in the Rk-blocks. For a fixed accuracy, the time increases linearly as the problem size increases. (In the largest test problem we ran out of memory.) Next, we record the time it takes to reduce the initial residual $r_0 := \|Ax_0 - b\|_2$ by a factor of 10^{-6} and the required number of iteration steps. We notice that the construction of the $\mathcal{H}$ - LU -preconditioner dominates the solution time. This may change, however, if a more accurate solution is desired, if we try to solve a worse-conditioned test problem, or if we further increase the problem size n . For the lowest accuracy 0.1, the number of iteration steps increases as the problem size increases, whereas for higher accuracies the preconditioner becomes optimal since the number of required steps remains constant as we increase the problem size.

The set-up time of the preconditioner has less impact when it is used for solving systems of equations such as the (Navier-) Stokes equations. Here, the problem size becomes $d \cdot n + p$, where d denotes the spatial dimension, n the number of velocity unknowns (per spatial dimension), and p the number of pressure unknowns. Here, we would require only an $\mathcal{H}$ - LU -decomposition of the first diagonal block of the stiffness matrix which is of size $n \times n$ in order to construct a preconditioner. Therefore, the time for the construction of the $\mathcal{H}$ - LU -factors will decrease relative to the time per iteration step.

In Table 4.3, we provide set-up times (for reordering) and convergence results for the bicgstab iteration as well as a multigrid iteration to compare with the results for $\mathcal{H}$ - LU as a preconditioner. Whereas RCM (reverse Cuthill-McKee) may not be the

TABLE 4.3
 $\epsilon = 1e - 8$, *cyclic convection, bicgstab/multigrid with SGS(RCM).*

n	Time (in sec.)		Time solver (in sec.)/#steps		
	RCM	Reorder	bicgstab	MG(V,1/1)	MG(W,1/1)
16129	0.27	0.05	1.44/31	1.76/33	1.23/17
65025	1.07	0.22	14.36/76	12.47/57	5.77/19
261121	5.19	0.89	74.25/100	84.37/97	24.81/20
1046529	23.34	3.63	306.79/100	356.05.84/100	102.08/20

best possible reordering, it yields convergence, while all iterations diverge without such prior reordering. We choose the symmetric Gauß-Seidel method after an RCM ordering as a preconditioner or smoother, respectively. We present results for the multigrid V- and W-cycle with one pre- and one post-smoothing step. Only the multigrid W-cycle yields an optimal method, and we note that whereas the new $\mathcal{H}$ -LU preconditioner requires a greater set-up time, the actual iteration is considerably faster.

REFERENCES

- [1] M. BEBENDORF AND W. HACKBUSCH, *Existence of $\mathcal{H}$ -matrix approximants to the inverse FE-matrix of elliptic operators with L^∞ -coefficients*, Numer. Math., 95 (2003), pp. 1–28.
- [2] J. BEY AND G. WITTUM, *Downwind numbering: Robust multigrid for convection diffusion problems*, Appl. Numer. Math., 23 (1997), pp. 177–192.
- [3] S. BÖRM, L. GRASEDYCK, AND W. HACKBUSCH, *Hierarchical matrices*, Lecture Notes 21, Max-Planck-Institute for Mathematics in the Sciences, Leipzig, Germany, 2003; also available online at www.mis.mpg.de/preprints/ln/index.html.
- [4] L. GRASEDYCK AND W. HACKBUSCH, *Construction and arithmetics of $\mathcal{H}$ -matrices*, Computing, 70 (2003), pp. 295–334.
- [5] L. GRASEDYCK, W. HACKBUSCH, AND S. LE BORNE, *Adaptive geometrically balanced clustering of $\mathcal{H}$ -matrices*, Computing, 73 (2003), pp. 1–23.
- [6] W. HACKBUSCH, *A sparse matrix arithmetic based on $\mathcal{H}$ -matrices. Part I: Introduction to $\mathcal{H}$ -matrices*, Computing, 62 (1999), pp. 89–108.
- [7] W. HACKBUSCH, L. GRASEDYCK, AND S. BÖRM, *An introduction to hierarchical matrices*, Math. Bohem., 127 (2002), pp. 229–241.
- [8] W. HACKBUSCH AND B. KHOROMSKIJ, *A sparse $\mathcal{H}$ -matrix arithmetic. Part II: Application to multi-dimensional problems*, Computing, 64 (2000), pp. 21–47.
- [9] W. HACKBUSCH AND T. PROBST, *Downwind Gauß-Seidel smoothing for convection dominated problems*, Numer. Linear Algebra Appl., 4 (1997), pp. 85–102.
- [10] S. LE BORNE, *Ordering techniques for two- and three-dimensional convection-dominated elliptic boundary value problems*, Computing, 64 (2000), pp. 123–155.
- [11] S. LE BORNE, *$\mathcal{H}$ -matrices for convection-diffusion problems with constant convection*, Computing, 70 (2003), pp. 261–274.
- [12] S. LE BORNE, *Hierarchical matrices for convection-dominated problems*, in Domain Decomposition Methods in Science and Engineering, Lecture Notes CSE 40, Springer, New York, 2004, pp. 631–638; also available online at ddm.org.
- [13] M. LINTNER, *The eigenvalue problem for the 2D Laplacian in H-matrix arithmetic and application to the heat and wave equation*, Computing, 72 (2004), pp. 293–323.
- [14] H. ROOS, M. STYNES, AND L. TOBISKA, *Numerical Methods for Singularly Perturbed Differential Equations: Convection Diffusion and Flow Problems*, Comp. Math. 24, Springer, Berlin, 1996.
- [15] H. A. VAN DER VORST, *BI-CGSTAB: A fast and smoothly converging variant of BI-CG for the solution of nonsymmetric linear systems*, SIAM J. Sci. Stat. Comp., 13 (1992), pp. 631–644.

COVARIANCE-PRECONDITIONED ITERATIVE METHODS FOR NONNEGATIVELY CONSTRAINED ASTRONOMICAL IMAGING*

JOHNATHAN M. BARDSLEY[†] AND JAMES G. NAGY[‡]

Abstract. We consider the problem of solving ill-conditioned linear systems $A\mathbf{x} = \mathbf{b}$ subject to the nonnegativity constraint $\mathbf{x} \geq \mathbf{0}$, and in which the vector $\mathbf{b}$ is a realization of a random vector $\hat{\mathbf{b}}$, i.e., $\mathbf{b}$ is noisy. We explore what the statistical literature tells us about solving noisy linear systems; we discuss the effect that a substantial black background in the astronomical object being viewed has on the underlying mathematical and statistical models; and, finally, we present several covariance-based preconditioned iterative methods that incorporate this information. Each of the methods presented can be viewed as an implementation of a preconditioned modified residual-norm steepest descent algorithm with a specific preconditioner, and we show that, in fact, the well-known and often used Richardson-Lucy algorithm is one such method. Ill-conditioning can inhibit the ability to take advantage of a priori statistical knowledge, in which case a more traditional preconditioning approach may be appropriate. We briefly discuss this traditional approach as well. Examples from astronomical imaging are used to illustrate concepts and to test and compare algorithms.

Key words. image restoration, linear models, preconditioning, statistical methods, weighted least squares

AMS subject classifications. 65F20, 65F30

DOI. 10.1137/040615043

1. Introduction. We seek solutions of the linear system with nonnegativity constraints

$$(1.1) \quad A\mathbf{x} = \mathbf{b}, \quad \mathbf{x} \geq \mathbf{0},$$

where by $\mathbf{x} \geq \mathbf{0}$ we mean $x_i \geq 0$ for all i . Such models arise naturally in image reconstruction, where $\mathbf{x}$ denotes the measured intensities at various pixels of an object being viewed by an imaging system. Such problems are typically solved via an iterative algorithm. The most common and straightforward approach is to simply ignore the nonnegativity constraint, in which case a large class of powerful methods can be utilized. Effective methods for solving (1.1) when A is an ill-conditioned matrix exist; see, e.g., [1, 3, 19]. In this paper we present preconditioned iterative methods that incorporate the nonnegativity constraint. Furthermore, we note that for many image deblurring problems, and in particular in astronomical imaging, the objects being viewed are often times composed, primarily, of a black background. This affects the mathematical model (1.1) in a nontrivial way. The algorithms that we present in this paper incorporate these changes as well.

An additional complication that is often ignored is that the data vector $\mathbf{b}$ in (1.1) is obtained via measurements that have associated random error, and is therefore a *single* realization of a random vector that we will call $\hat{\mathbf{b}}$. In this paper, we will focus

*Received by the editors September 14, 2004; accepted for publication (in revised form) by Y. Saad September 26, 2005; published electronically March 17, 2006.

<http://www.siam.org/journals/simax/27-4/61504.html>

[†]Department of Mathematical Sciences, University of Montana, Missoula, MT 59812 (bardsleyj@mso.umt.edu). Research supported in part by University of Montana research grant MRA682.

[‡]Department of Mathematics and Computer Science, Emory University, Atlanta, GA 30322 (nagy@mathcs.emory.edu). Research supported in part by the NSF under grant DMS-05-11454 and by an Emory University research committee grant.

on the case in which the statistical model for $\hat{\mathbf{b}}$ is given by

$$(1.2) \quad \hat{\mathbf{b}} = A\mathbf{x}_{\text{true}} + \boldsymbol{\eta},$$

where $\mathbf{x}_{\text{true}}$ is the discretized object, or true image, and is what we want to estimate; and the random vector $\boldsymbol{\eta}$ characterizes the measurement errors, or noise, in the data collection process, and is assumed to have a known distribution. We assume that the exact data $A\mathbf{x}_{\text{true}}$ and the noise $\boldsymbol{\eta}$ are statistically uncorrelated and independent. Each component of $\hat{\mathbf{b}}$ is then a random variable with distribution

$$(1.3) \quad \hat{b}_i = [A\mathbf{x}_{\text{true}}]_i + \eta_i, \quad i = 1, \dots, n,$$

where η_i is a random variable with a known distribution, and n is the number of pixels in the collected image $\mathbf{b}$. Our task is to estimate $\mathbf{x}_{\text{true}}$ from $\mathbf{b}$ given statistical model (1.2). It is tempting to simply ignore (1.2) and solve (1.1) directly, with or without the nonnegativity constraint. In fact, this is what is usually done. In many instances, ignoring (1.2) has no noticeable effect on the performance of the algorithm used, but as we will see, there are instances in which prior knowledge of noise statistics can be used to accelerate the convergence of the iterative method used.

As a test-bed of examples, we consider a class of ill-posed inverse problems that arise when reconstructing an image of an astronomical object taken by a ground-based telescope. In this case, light from an object, $\mathbf{x}_{\text{true}}$, in outer space travels through a medium with refractive index fluctuations (the earth's atmosphere), which has a blurring effect on the data. In addition, as the light passes through the telescope, diffractive blurring occurs. These blurring effects are assumed to be characterized by a (known) ill-conditioned blurring matrix A . That is, if $\mathbf{x}_{\text{true}}$ is the discretized object of interest, $A\mathbf{x}_{\text{true}}$ is what is seen after the light from $\mathbf{x}_{\text{true}}$ has traveled through the earth's atmosphere and through the telescope. In order to collect the blurred image data, a charge-coupled-device (CCD) camera is used. A CCD camera is, essentially, an array of sensors that creates a pixelated image by counting the number of photons that hit each sensor. We assume that the random noise that enters the data does so during the collection of the image by the CCD camera. Other information can be included in the formulation of the image reconstruction problem; namely, in our application, the object being viewed, $\mathbf{x}_{\text{true}}$, has nonnegative intensity. This information can be included as a nonnegativity constraint in the reconstruction algorithm.

Finally, we note that in the following we will use the notation $N(\mu, \sigma^2)$ to denote a Gaussian random variable with mean μ and variance σ^2 and $\text{Poiss}(\beta)$ to denote a Poisson random variable with mean and variance equal to β .

This paper is outlined as follows. In section 2, we present the statistical material that will be necessary in order to motivate the covariance-preconditioned iterative methods. The methods for solving (1.1), (1.2) are then presented in section 3. In section 4, we present numerical results to illustrate the effectiveness of the preconditioners, and some concluding remarks are given in section 5.

2. Statistical considerations. We consider the linear statistical model (1.2). We assume throughout this paper that $\boldsymbol{\eta}$ is a random n -vector with $E(\boldsymbol{\eta}) = \mathbf{0}$, where E denotes the componentwise expected value, or mean, of $\boldsymbol{\eta}$, i.e., $E(\boldsymbol{\eta}) = (E(\eta_1), \dots, E(\eta_n))$. We let $C_{\boldsymbol{\eta}}$ denote the *covariance matrix* of $\boldsymbol{\eta}$ defined by

$$[C_{\boldsymbol{\eta}}]_{ij} = E[\eta_i \eta_j],$$

and assume that it is a known and nonsingular matrix. Given (1.2), we seek the “best” estimate of the object $\mathbf{x}_{\text{true}}$ from our data vector $\mathbf{b}$, but first we must define what we mean by “best” in a statistical sense.

DEFINITION 2.1. *The Best Linear Unbiased Estimator (BLUE) for $\mathbf{x}_{\text{true}}$ from $\mathbf{b}$ in (1.2) is the random vector $\hat{\mathbf{x}}_{\text{BLUE}}$, which minimizes*

$$(2.1) \quad J(\hat{\mathbf{x}}) = E(\|\hat{\mathbf{x}} - \mathbf{x}_{\text{true}}\|_2^2)$$

subject to the constraints

$$(2.2) \quad \hat{\mathbf{x}} = B\mathbf{b}, \quad B \in \mathbb{R}^{n \times n},$$

$$(2.3) \quad E(\hat{\mathbf{x}}) = \mathbf{x}_{\text{true}}.$$

$\hat{\mathbf{x}}_{\text{BLUE}}$ is called linear because of (2.2) and unbiased because of (2.3).

Forgetting, for the moment, ill-conditioning, we assume that $\hat{\mathbf{x}}_{\text{BLUE}}$ is the reconstruction that we seek. The following theorem tells us how to find it.

THEOREM 2.2 (Gauss–Markov). *If A has full rank and $\mathbf{b}$ is a realization of the random vector $\hat{\mathbf{b}}$ in (1.2), then*

$$(2.4) \quad \hat{\mathbf{x}}_{\text{BLUE}} = (A^T C_{\boldsymbol{\eta}}^{-1} A)^{-1} A^T C_{\boldsymbol{\eta}}^{-1} \mathbf{b}.$$

Example. Suppose $\boldsymbol{\eta}$ is an independent and identically distributed (i.i.d.) Gaussian random vector, i.e., $\eta_i = N(0, \sigma^2)$ for all i , where $N(0, \sigma^2)$ denotes a Gaussian random variable with mean 0 and variance σ^2 . Then the covariance matrix is given by $C_{\boldsymbol{\eta}} = \sigma^2 I$, and hence, provided A has full rank, $\hat{\mathbf{x}}_{\text{BLUE}}$ is the solution of

$$(2.5) \quad \min_{\mathbf{x}} \|\mathbf{A}\mathbf{x} - \mathbf{b}\|_2^2.$$

In many of the applications in which the linear model (1.2) arises, the noise is, in fact, i.i.d. Gaussian. Nonetheless, in astronomical imaging the noise does not have this form. We now focus our attention on the noise statistics of CCD camera image formation.

2.1. CCD camera noise statistics. The following statistical model (see Refs. [16, 17]) applies to image data from a CCD detector array:

$$(2.6) \quad \hat{\mathbf{b}} = \text{Pois}(A\mathbf{x}_{\text{true}}) + \text{Pois}(\beta \cdot \mathbf{1}) + N(\mathbf{0}, \sigma^2 I),$$

where $\mathbf{1}$ is a vector of all ones. The relation described by (2.6) means that each element $\hat{b}_i$ of the vector $\hat{\mathbf{b}}$ is a random variable with distribution

$$(2.7) \quad \hat{b}_i = n_{\text{obj}}(i) + n_0(i) + g(i), \quad i = 1, \dots, n.$$

Equation (2.7) can be described as follows:

- $n_{\text{obj}}(i)$ is the number of object-dependent photoelectrons measured by the i th detector in the CCD array. It is a Poisson random variable with Poisson parameter $[A\mathbf{x}_{\text{true}}]_i$.
- $n_0(i)$ is the number of background photoelectrons, which arise from both natural and artificial sources, measured by the i th detector in the CCD array. It is a Poisson random variable with a fixed positive Poisson parameter β .
- $g(i)$ is the so-called readout noise, which is due to random errors caused by the CCD electronics and errors in the analog-to-digital conversion of measured voltages. It is a Gaussian random variable with mean 0 and fixed variance σ^2 .

The random variables $n_{\text{obj}}(i)$, $n_0(i)$, and $g(i)$ are assumed to be independent of one another and of $n_{\text{obj}}(j)$, $n_0(j)$, and $g(j)$ for $i \neq j$.

As in [16], we consider the case in which the readout noise variance σ^2 is large. Then, according to Feller [6, pp. 190 and 245], the following approximation is accurate:

$$(2.8) \quad N(\sigma^2, \sigma^2) \approx \text{Poiss}(\sigma^2).$$

Using the independence properties of the random variables in (2.7) we obtain the following approximation of (2.6):

$$(2.9) \quad \hat{\mathbf{b}} + \sigma^2 \cdot \mathbf{1} = \text{Poiss}(A\mathbf{x}_{\text{true}} + \beta \cdot \mathbf{1} + \sigma^2 \cdot \mathbf{1}).$$

This motivates computing a minimizer of the negative log-likelihood function for the statistical model (2.9), which is given by

$$(2.10) \quad \ell(A\mathbf{x}; \mathbf{b}) = \sum_{i=1}^N ([A\mathbf{x}]_i + \beta + \sigma^2) - \sum_{i=1}^N (b_i + \sigma^2) \log([A\mathbf{x}]_i + \beta + \sigma^2),$$

with respect to $\mathbf{x}$, and subject to the nonnegativity constraint $\mathbf{x} \geq \mathbf{0}$. A popular algorithm that computes such a minimizer is the Richardson–Lucy (RL) iteration [2], which has the form

$$(2.11) \quad \mathbf{x}_{k+1} = \mathbf{x}_k \odot A^T \left(\frac{\mathbf{b} + \sigma^2 \cdot \mathbf{1}}{A\mathbf{x}_k + \beta \cdot \mathbf{1} + \sigma^2 \cdot \mathbf{1}} \right),$$

where $\odot$ denotes componentwise multiplication, and the division operation is also done componentwise. The advantages of RL is that it is very simple to implement, it enforces a nonnegativity constraint, each iteration is relatively inexpensive, and it is statistically valid. A more sophisticated constrained optimization approach for minimizing (2.10) subject to $\mathbf{x} \geq \mathbf{0}$ can be employed (see, e.g., [1]), resulting in methods that are faster to converge, but are more costly per iteration. We do not consider such approaches here.

A different approach, and the one we focus on in this paper, is to invoke the approximation used in (2.8) yet again. First, we note that since (2.8) holds, and since $[A\mathbf{x}_{\text{true}}]_i + \beta + \sigma^2 \geq \sigma^2$ provided that the matrix A has nonnegative entries, which we will assume here, we obtain

$$(2.12) \quad \text{Poiss}([A\mathbf{x}_{\text{true}}]_i + \beta + \sigma^2) \approx N([A\mathbf{x}_{\text{true}}]_i + \beta + \sigma^2, [A\mathbf{x}_{\text{true}}]_i + \beta + \sigma^2),$$

again using Feller [6, pp. 190 and 245]. This allows us to express (2.9) as

$$(2.13) \quad \hat{\mathbf{b}} + \sigma^2 \cdot \mathbf{1} = A\mathbf{x}_{\text{true}} + \beta \cdot \mathbf{1} + \sigma^2 \cdot \mathbf{1} + N(\mathbf{0}, W^{-1}),$$

where

$$(2.14) \quad W^{-1} = \text{diag}(A\mathbf{x}_{\text{true}} + \beta \cdot \mathbf{1} + \sigma^2 \cdot \mathbf{1});$$

or, more succinctly,

$$(2.15) \quad \hat{\mathbf{b}} - \beta \cdot \mathbf{1} = A\mathbf{x}_{\text{true}} + N(\mathbf{0}, W^{-1}).$$

Assuming the linear statistical model (2.15), (2.14), the Gauss–Markov theorem tells us that given the data vector $\mathbf{b}$, $\hat{\mathbf{x}}_{\text{BLUE}}$ is obtained by solving

$$(2.16) \quad \min_{\mathbf{x}} \|A\mathbf{x} - (\hat{\mathbf{b}} - \beta \cdot \mathbf{1})\|_W^2.$$

3. The algorithms. We now derive algorithms that incorporate the statistical results from the previous section. In addition, we see that in the case in which the object being viewed has a substantial black background, i.e., is in large part zero, these algorithms implicitly incorporate this information. This is important because when $\mathbf{x}_{\text{true}}$ has a large number of zeros, the effect on the mathematical and statistical model can be significant.

3.1. The effects of zeros in the components of $\mathbf{x}_{\text{true}}$. We begin with a simple observation and then analyze its effects on the models (1.1) and (1.2): If $\mathbf{x}_{\text{true}}$ has a large number of zero components, then much of the matrix A is extraneous.

To see this, we first define the *active set* of $\mathbf{x}_{\text{true}}$ by

$$\mathcal{A}_{\text{true}} = \{i \mid [\mathbf{x}_{\text{true}}]_i = 0\},$$

and the diagonal matrix D_{true} by

$$[D_{\text{true}}]_{jj} = \begin{cases} 1, & j \notin \mathcal{A}_{\text{true}} \\ 0, & j \in \mathcal{A}_{\text{true}}. \end{cases}$$

We define A_{pos} to be the matrix that results after column i is removed from A for each $i \in \mathcal{A}_{\text{true}}$, and $\mathbf{x}_{\text{true}}^{\text{pos}}$ to be $\mathbf{x}_{\text{true}}$ restricted to the indices $i \notin \mathcal{A}_{\text{true}}$. The key observation, then, is

$$(3.1) \quad A\mathbf{x}_{\text{true}} = (AD_{\text{true}})\mathbf{x}_{\text{true}} = A_{\text{pos}}\mathbf{x}_{\text{true}}^{\text{pos}},$$

which yields the modified linear system

$$(3.2) \quad A_{\text{pos}}\mathbf{x}^{\text{pos}} = \mathbf{b},$$

and corresponding reduced linear statistical model

$$(3.3) \quad \hat{\mathbf{b}} = A_{\text{pos}}\mathbf{x}_{\text{true}}^{\text{pos}} + \boldsymbol{\eta}.$$

We note that for the indices $i \notin \mathcal{A}_{\text{true}}$, (3.2) and (3.3) are equivalent to (1.1) and (1.2), respectively.

By Theorem 2.2, if A has full column rank,

$$(3.4) \quad \hat{\mathbf{x}}_{\text{BLUE}}^{\text{pos}} = (A_{\text{pos}}^T C_{\boldsymbol{\eta}}^{-1} A_{\text{pos}})^{-1} A_{\text{pos}}^T C_{\boldsymbol{\eta}}^{-1} \mathbf{b}.$$

Combining (3.4) with the knowledge that the other indices are known to be zero yields the BLUE for $\mathbf{x}_{\text{true}}$ given $\mathbf{b}$; specifically,

$$(3.5) \quad \hat{\mathbf{x}}_{\text{BLUE}} = (D_{\text{true}} A^T C_{\boldsymbol{\eta}}^{-1} A D_{\text{true}})^{\dagger} D_{\text{true}} A^T C_{\boldsymbol{\eta}}^{-1} \mathbf{b},$$

where “ $\dagger$ ” denotes the *pseudoinverse*. This follows from the fact that (3.5) satisfies (2.1), (2.2), and (2.3), which is straightforward to check.

We conclude with the important observation that if $\mathbf{x}_{\text{true}}$ has a large number of zeros, solving the normal equations

$$(3.6) \quad (A^T C_{\boldsymbol{\eta}}^{-1} A) \mathbf{x} = A^T C_{\boldsymbol{\eta}}^{-1} \mathbf{b},$$

using, for example, the conjugate gradient algorithm (CG), may yield a solution that is far from the solution we desire, which is given by (3.5). Unfortunately, we cannot compute (3.5) since in practice we will never know D_{true} .

3.2. Covariance-preconditioned iterative methods. What we can do, though, is build an iterative method that does not require D_{true} . To do this, we first note that equation (3.5) implies that $\hat{\mathbf{x}}_{\text{BLUE}}$ is a solution of the equation

$$(3.7) \quad D_{\text{true}} A^T C_{\boldsymbol{\eta}}^{-1} (A D_{\text{true}} \hat{\mathbf{x}}_{\text{BLUE}} - \mathbf{b}) = \mathbf{0}.$$

But since $\hat{\mathbf{x}}_{\text{BLUE}}$ is zero whenever $\mathbf{x}_{\text{true}}$ is zero, we have $D_{\text{true}} \hat{\mathbf{x}}_{\text{BLUE}} = \hat{\mathbf{x}}_{\text{BLUE}}$ and $\hat{\mathbf{x}}_{\text{BLUE}} \odot D_{\text{true}} \mathbf{v} = \hat{\mathbf{x}}_{\text{BLUE}} \odot \mathbf{v}$, where $\mathbf{v}$ is a vector of length n . Here again we use $\odot$ to denote componentwise multiplication. It immediately follows, then, that $\hat{\mathbf{x}}_{\text{BLUE}}$ is a solution of the nonlinear equation

$$(3.8) \quad \mathbf{x} \odot A^T C_{\boldsymbol{\eta}}^{-1} (A \mathbf{x} - \mathbf{b}) = \mathbf{0}.$$

Equation (3.8) motivates the fixed point iteration

$$(3.9) \quad \mathbf{x}_{k+1} = \mathbf{x}_k - \tau_k \mathbf{x}_k \odot A^T C_{\boldsymbol{\eta}}^{-1} (A \mathbf{x}_k - \mathbf{b}),$$

to compute $\hat{\mathbf{x}}_{\text{BLUE}}$, which is nothing other than the Preconditioned Modified Residual Norm Steepest Descent Algorithm (PMRNSD) of [14], with covariance-preconditioner $C_{\boldsymbol{\eta}}^{-1/2}$. The basic MRNSD algorithm (i.e., without preconditioning) is a variant of an approach proposed by Kaufman [12], and can be viewed as a modification of the residual norm steepest descent method [15].

The line search parameter τ_k in (3.9) (see [14] for details) is given by

$$(3.10) \quad \tau_k = \min\{\tau_{\text{uc}}, \tau_{\text{bd}}\},$$

where, if $\mathbf{v}_k = \mathbf{x}_k \odot \nabla J(\mathbf{x}_k)$,

$$(3.11) \quad \tau_{\text{uc}} = -\frac{\langle \mathbf{v}_k, \nabla J(\mathbf{x}_k) \rangle}{\langle \mathbf{v}_k, A^T A \mathbf{v}_k \rangle},$$

and

$$(3.12) \quad \tau_{\text{bd}} = \min \{ -[\mathbf{x}_k]_i / [\mathbf{v}_k]_i \mid [\mathbf{v}_k]_i < 0 \}.$$

We note that (3.10) ensures that the PMRNSD iterates will satisfy the nonnegativity constraint $\mathbf{x}_k \geq \mathbf{0}$ for all k , which is what we wanted.

As was mentioned in the introduction, the nonnegativity constraint $\mathbf{x} \geq \mathbf{0}$ arises naturally as a physical constraint, but it serves another important purpose. Notice that (3.8) does not have a unique solution. Three that we can state immediately are $\mathbf{x} = \mathbf{0}$, $\mathbf{x} = \hat{\mathbf{x}}_{\text{BLUE}}$, and the minimum norm solution of $A \mathbf{x} = \mathbf{b}$ given by $\mathbf{x} = A^\dagger \mathbf{b}$. But the nonnegatively constrained problem

$$(3.13) \quad \min_{\mathbf{x} \geq \mathbf{0}} \|C_{\boldsymbol{\eta}}^{-1/2} A \mathbf{x} - C_{\boldsymbol{\eta}}^{-1/2} \mathbf{b}\|^2,$$

does have a unique solution (assuming A has full column rank and $C_{\boldsymbol{\eta}}$ is positive definite), which is given by

$$(3.14) \quad \mathbf{x}^* = (D^* A^T C_{\boldsymbol{\eta}}^{-1} A D^*)^\dagger D^* A^T \mathbf{b}.$$

Here D^* is a diagonal matrix defined by $[D^*]_{ii} = 1$ if $[\mathbf{x}^*]_i > 0$ and $[D^*]_{ii} = 0$ if $[\mathbf{x}^*]_i = 0$. More importantly, $\mathbf{x}^*$ is also a solution of (3.8). Furthermore, (3.9) can also be obtained by applying the MRNSD algorithm of [14] to the nonnegatively constrained problem (3.13). Thus (3.9) converges, at least in principle, to the unique solution of (3.8) given by (3.14).

We now present algorithms of the form (3.9) that arise from three different choices of $C_{\boldsymbol{\eta}}^{-1}$.

Examples.

1. In the i.i.d. Gaussian noise case, $C_{\boldsymbol{\eta}}^{-1} = \sigma^2 I$, and (3.9) becomes

$$(3.15) \quad \mathbf{x}_{k+1} = \mathbf{x}_k - \tau_k \mathbf{x}_k \odot A^T(A\mathbf{x}_k - \mathbf{b}),$$

which is the MRNSD algorithm of [14] for approximately solving (2.5) subject to $\mathbf{x} \geq \mathbf{0}$.

2. When the noise model is given by (2.15), $C_{\boldsymbol{\eta}}^{-1} = W$, where W is defined in (2.14); then (3.9) becomes

$$(3.16) \quad \mathbf{x}_{k+1} = \mathbf{x}_k - \tau_k \mathbf{x}_k \odot A^T W(A\mathbf{x}_k - (\mathbf{b} - \beta \cdot \mathbf{1})).$$

We will call (3.16) weighted MRNSD (WMRNSD), and it approximately solves (2.16) subject to $\mathbf{x} \geq \mathbf{0}$. Unfortunately, the diagonal weight matrix W defined by (2.14) cannot be obtained exactly since $A\mathbf{x}_{\text{true}}$ is not known, but it can be approximated. For example, in the numerical experiments we will use the approximation

$$(3.17) \quad W^{-1} = \text{diag}(\mathbf{b} + \sigma^2),$$

which is motivated by (2.15). A more sophisticated approach for approximating the weight matrix W is given in [7].

3. Rather than approximate the weight matrix W as is suggested in example 2, one can instead replace it with an iteration dependent weight matrix

$$(3.18) \quad W(\mathbf{x}_k)^{-1} = \text{diag}(A\mathbf{x}_k + \beta \cdot \mathbf{1} + \sigma^2 \cdot \mathbf{1}).$$

Then (3.16) becomes

$$(3.19) \quad \begin{aligned} \mathbf{x}_{k+1} &= \mathbf{x}_k - \tau_k \mathbf{x}_k \odot A^T \left(\frac{A\mathbf{x}_k - (\mathbf{b} - \beta \cdot \mathbf{1})}{A\mathbf{x}_k + \beta \cdot \mathbf{1} + \sigma^2 \cdot \mathbf{1}} \right), \\ &= \mathbf{x}_k - \tau_k \mathbf{x}_k \odot A^T \left(\mathbf{1} - \frac{\mathbf{b} + \sigma^2 \cdot \mathbf{1}}{A\mathbf{x}_k + \beta \cdot \mathbf{1} + \sigma^2 \cdot \mathbf{1}} \right), \end{aligned}$$

where division is done componentwise. Assuming that $A^T \mathbf{1} = \mathbf{1}$ and setting $\tau_k = 1$ for all k , (3.19) takes the form of the RL algorithm (2.11)! Thus RL can be viewed as an application of algorithm (3.9) in which $C_{\boldsymbol{\eta}}^{-1} = W(\mathbf{x}_k)$. We note that RL iterates have been shown to satisfy $\mathbf{x}_k \geq \mathbf{0}$ for all k . Thus no line search is necessary.

3.3. The standard preconditioning approach and ill-conditioning. The standard approach to preconditioning amounts to constructing a matrix, M , such that $M \approx A$, and then applying the iterative method to the system $M^{-1}A\mathbf{x} = M^{-1}\mathbf{b}$. If M is a good approximation of A in the sense that the singular values of $M^{-1}A$ are clustered around a fixed point away from zero, then the iterative method will converge quickly. However, care must be taken for severely ill-conditioned problems that arise in image deblurring since linear systems of the form $M\mathbf{z} = \mathbf{w}$ must be solved at each iteration. If M is a good approximation of A , then M will be ill-conditioned, and therefore inverting M may cause large inaccuracies during the early iterations. The standard approach to preconditioning amounts to constructing a matrix, M , such that $M \approx A$, and then applying the iterative method to the system $M^{-1}A\mathbf{x} = M^{-1}\mathbf{b}$ [8].

One preconditioning approach for ill-posed problems, proposed in [10], uses an approximate truncated spectral decomposition preconditioner. Specifically, A is first approximated by a block circulant matrix with circulant blocks (BCCB) from which it is easy to compute the spectral decomposition

$$M = \mathcal{F}^* \Lambda \mathcal{F},$$

where $\mathcal{F}$ is the unitary two-dimensional discrete Fourier transform matrix, and Λ is a diagonal matrix containing the eigenvalues of M . This decomposition can be computed very efficiently using fast Fourier transforms (FFT). With this spectral decomposition, we define a preconditioner

$$(3.20) \quad M_\tau = \mathcal{F}^* \Lambda_\tau \mathcal{F},$$

where

$$(3.21) \quad [\Lambda_\tau]_i = \begin{cases} \lambda_i & \text{if } |\lambda_i| \geq \tau \\ 1 & \text{if } |\lambda_i| < \tau. \end{cases}$$

The *truncation parameter*, τ , can be chosen using a regularization parameter choice method, such as generalized cross validation [5, 11, 19]. To understand why this approach works, we need to understand a little about truncated iteration regularization. It can be shown (see [5, 11, 19]) that the early iterations filter out components of the solution corresponding to the small singular values of the matrix. That is, the early iterations tend to reconstruct lower frequency components of the data, while the higher frequency components of the data, where the noise is primarily contained, are filtered out. It is this part of the iteration that we want to accelerate. At some point, though, the high frequency components start to be reconstructed, and the iterates begin to be corrupted with noise; this part of the iteration we do not want to accelerate.

In the following, we denote the subspace comprised of the frequencies containing the noise the *noise subspace*. The subspace comprised of the remaining frequencies will be denoted as the *signal subspace*. For a preconditioned iterative method, it is the singular values of the preconditioned system $M^{-1}A$ that we must consider. By clustering all of the singular values around one, we no longer have the information to distinguish between the signal and noise subspaces. However, if we use the preconditioner M_τ with a good spectral approximation, then it can be shown that the large singular values (i.e., those corresponding to the signal subspace) are clustered at one, and are well separated from the small singular values (those corresponding to the noise subspace); see [9, 10] for further details. Iterative methods applied to such a system will converge quickly to the regularized solution.

The PMRNSD algorithm given in [14], has the form

$$(3.22) \quad \mathbf{x}_{k+1} = \mathbf{x}_k - \tau_k \mathbf{x}_k \odot A^T M^{-T} M^{-1} (A \mathbf{x} - \mathbf{b}),$$

where the preconditioning matrix M is chosen as described above. Although this approach is not statistically motivated, it is often quite effective. This is due to the fact that if A is an ill-conditioned matrix, which it will usually be in imaging applications, we seek a stable, or regularized, approximation of $\hat{\mathbf{x}}_{\text{BLUE}}$ rather than $\hat{\mathbf{x}}_{\text{BLUE}}$ itself, and the more ill-conditioned A is, the further away from $\hat{\mathbf{x}}_{\text{BLUE}}$ will be the regularized solution. Thus one can expect that in many instances, (3.22) with a good preconditioner will outperform the covariance-preconditioned algorithm (3.9).

Note that (3.22) uses a “left” preconditioning approach since the algorithm can be viewed as applying the (unpreconditioned) MRNSD algorithm to the linear system $M^{-1}A\mathbf{x} = M^{-1}\mathbf{b}$. We also mention here that right-preconditioning (2.5) or (2.16) does not work well for the algorithms presented above. This is due to the fact that, as was pointed out in [14], MRNSD is already a right-preconditioned algorithm with step-dependent right-preconditioner $X_k = \text{diag}(\mathbf{x}_k)$. Additional right-preconditioning modifies the algorithm in such a way that it can no longer be viewed as an iterative method for solving (3.8), which we have argued is the right problem to solve.

4. Numerical results. In this section we present numerical results using two image restoration test problems. The first set of data was developed at the U.S. Air Force Phillips Laboratory, Lasers and Imaging Directorate, Kirtland Air Force Base, New Mexico. The image is a computer simulation of a field experiment showing a satellite as taken from a ground-based telescope. The true and blurred images have 256×256 pixels, and are shown in Figure 4.1. We remark that the $65,536 \times 65,536$ blurring matrix A is not constructed explicitly, but is defined implicitly by a so-called *point spread function* (PSF). The data for this test problem, including the true image and PSF, is contained in the *RestoreTools* image restoration package [13]. In addition, this package contains an implementation of MRNSD, as well as functions to efficiently implement matrix-vector multiplications with A (using the PSF), and for construction BCCB preconditioners; see [13] for more details.

The second test problem, which is shown in Figure 4.2 is a computer simulation of a star cluster, which has been used by astronomers to test and compare image deblurring methods for the Hubble Space Telescope. The data can be obtained from the Space Telescope Science Institute. The blurring matrix A is defined by a PSF that was supplied by Dr. Brent Ellerbroek, Adaptive Optics Program Manager at the Gemini Observatory in Hilo, HI. This PSF is a simulation of the type of PSF that results from an adaptive optics telescope system. Such systems have PSFs that are nearly diffraction limited. This is due to the fact that the refractive blurring due to the atmosphere is, for the most part, canceled out by the deformable mirror of the telescope, and hence, the blur in the image is due almost entirely to the diffractive blurring effects of the finite aperture of the telescope.

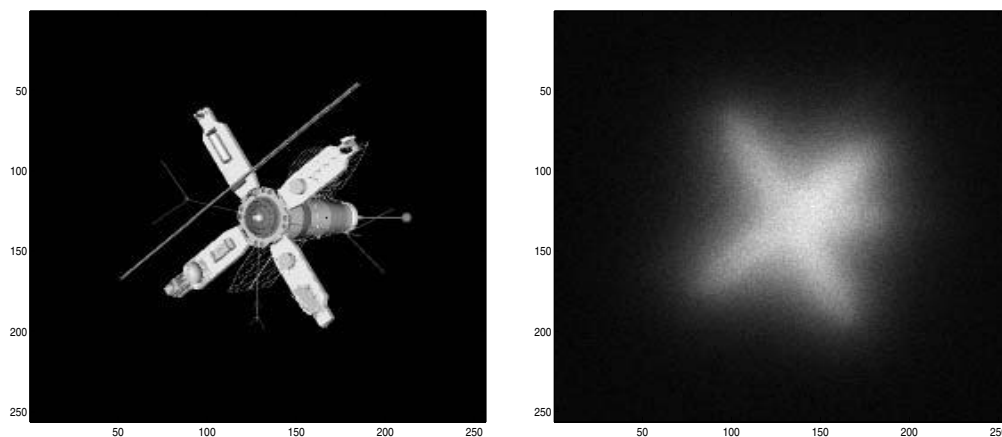


FIG. 4.1. *Satellite: True image and blurred, noisy data.*

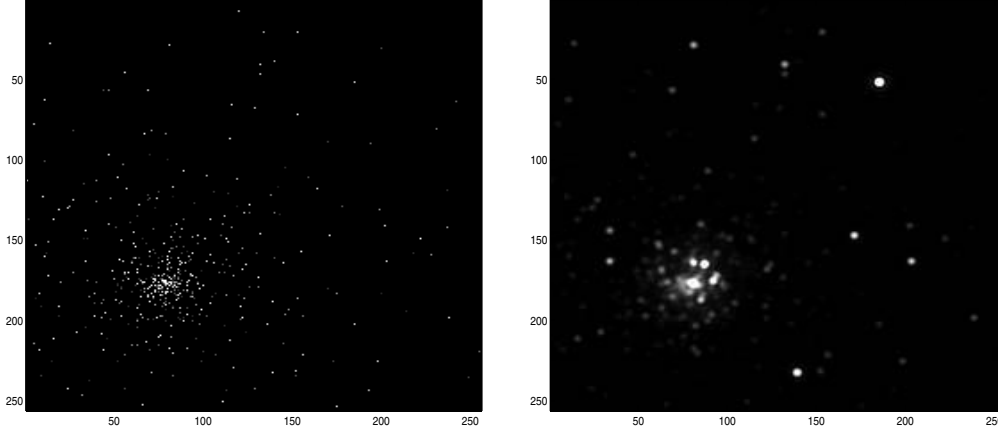
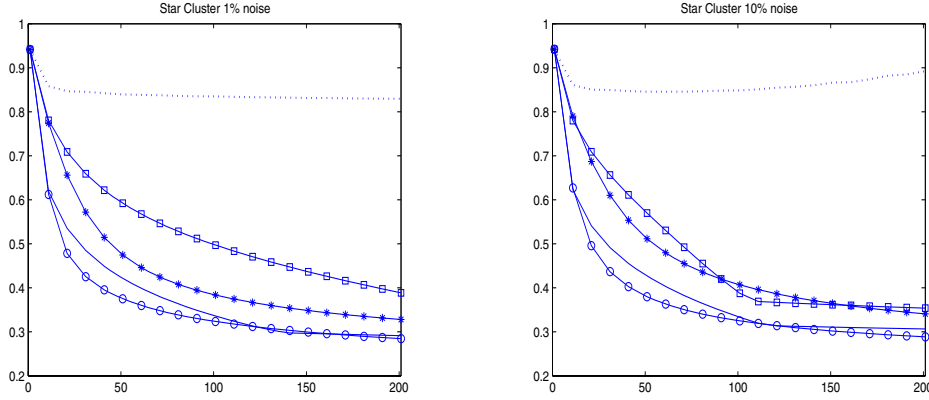
FIG. 4.2. *Star Field: True image and blurred, noisy data.*

FIG. 4.3. *Star field reconstructions: Relative error $\|\mathbf{x}_k - \mathbf{x}_{\text{true}}\|/\|\mathbf{x}_{\text{true}}\|$ versus iteration count. The left and right plots correspond to 1% and 10% error, respectively. The dotted line denotes PCG; the solid line with squares denotes MRNSD; the solid line with stars denotes RL; the solid line denotes PRMNSD; and the solid line with circles denotes WMRNSD.*

In Figures 4.3 and 4.4, we compare the performance of MRNSD (3.15), WMRNSD (3.16), RL (2.11), PMRNSD (3.22) and the preconditioned conjugate gradient algorithm (PCG). As a measure for the accuracy of the reconstructions, we use the *relative error* $\|\mathbf{x}_k - \mathbf{x}_{\text{true}}\|/\|\mathbf{x}_{\text{true}}\|$. We use statistical model (2.6) for data generation. We take the sky, background count parameter to be $\beta = 10$ and the standard deviation of the normal random variable to be $\sigma = 5$. These values are representative of CCD cameras used in astronomy. We compare reconstructions for both test problems at two different noise levels. In particular, we consider the cases in which the signal to noise ratios (SNR) are approximately 100 and 10, which corresponds to a noise power that is 1% and 10%, respectively, of the signal power. The SNR is defined by $\|A\mathbf{x}_{\text{true}}\|/\sqrt{E(\|\boldsymbol{\eta}\|^2)}$, where, for noise model (2.6), $E(\|\boldsymbol{\eta}\|^2) = n(\sigma^2 + \beta + \beta^2) + \sum_{i=1}^n \hat{b}_i$. For WMRNSD, we approximate W as in (3.17). Note that for moderate to large values of σ^2 , say $\sigma^2 \geq 3^2$, it is extremely unlikely for the Gaus-

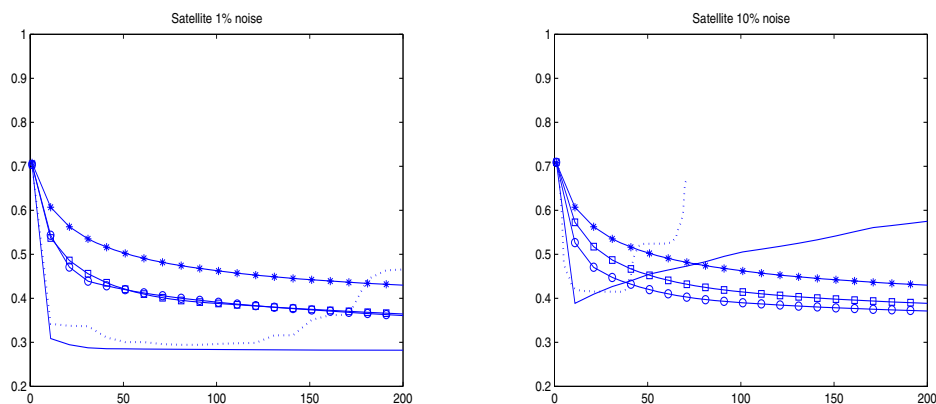


FIG. 4.4. *Satellite reconstructions: Relative error $\|\mathbf{x}_k - \mathbf{x}_{\text{true}}\|/\|\mathbf{x}_{\text{true}}\|$ versus iteration count. The left and right plots correspond to 1% and 10% error, respectively. The dotted line denotes PCG; the solid line with squares denotes MRNSD; the solid line with stars denotes RL; the solid line denotes PRMNSD; and the solid line with circles denotes WMRNSD.*

sian $N(\sigma^2, \sigma^2)$ to take on negative values. Then since Poisson random variables take on only nonnegative integer values, the random variable $b_i + \sigma^2$ will nearly always be positive. Thus our choice of W will nearly always be positive definite. This was the case in each of our experiments. For PMRNSD and PCG, the BCCB preconditioner (3.20), (3.21) computed by *RestoreTools* [13] was used.

In Figure 4.3, we see that for the star field data with both 1% and 10% noise, WMRNSD is the most effective algorithm. This is a rather surprising result if one views W as a preconditioner. Then, for this test problem, the statistically motivated, diagonal preconditioned algorithm (3.16) actually has better convergence properties than the structure motivated BCCB preconditioned algorithm (3.22).

In Figure 4.4, we see that WMRNSD is not as effective when used on the satellite data, and that PMRNSD with the BCCB preconditioner is the fastest to converge for all of the methods and for both 1% and 10% noise levels. Thus, in this case, using the more accurate statistical model is not as critical, and, in fact, can hinder convergence. At first, this may seem disappointing, but in the recent paper [18], the authors make the observation that in the case of statistical model (2.6), reconstructions obtained using the standard (i.e., unweighted and without nonnegativity constraints) least squares approach is comparable, *in terms of relative error*, to reconstructions obtained using approaches that more accurately model noise statistics and that incorporate a nonnegativity constraint provided the object of interest is reasonably smooth with large regions of high intensity. The satellite is an object of this type. On the other hand, for stellar-like objects such as the star field example, it was shown in [18] that approaches that incorporate a nonnegativity constraint and that more accurately account for CCD noise statistics typically provide better reconstructions, again in terms of relative error. These observations are supported by the convergence graphs in Figures 4.3 and 4.4.

The convergence plot on the right in Figure 4.4 indicates that if the linear system is highly ill-conditioned and there is a large amount of noise in the data, the inversion of noise may happen after just a few iterations, in which case it is important to have reliable stopping criteria.

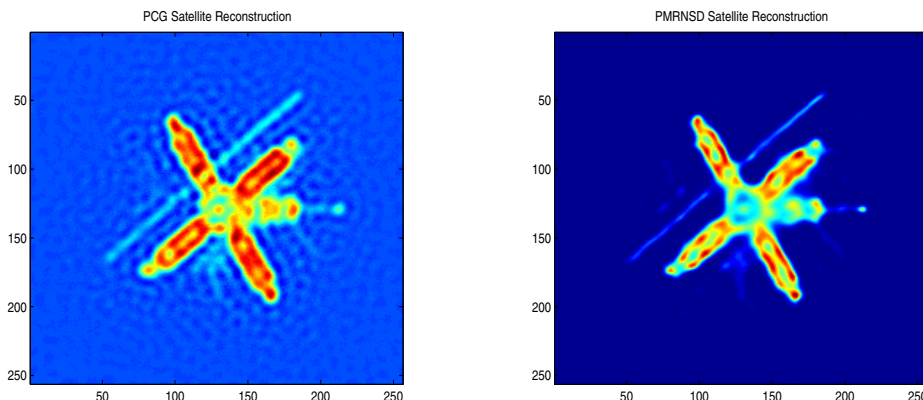


FIG. 4.5. *Satellite reconstructions for 1% noise. On the left-hand side is the PCG reconstruction. On the right-hand side is the PMRNSD reconstruction.*

It is also interesting to note that in Figure 4.3 the performance of PCG is extremely poor. This has nothing to do with the choice of preconditioner. The poor performance is due to the fact that PCG computes an approximate solution of the normal equations

$$(4.1) \quad A^T A \mathbf{x} = A^T \mathbf{b},$$

whereas the other algorithms in the comparison seek an approximation of the solution of nonnegatively constrained problems of the form (3.13), which has solution (3.14). It is clear that these two solutions can be quite different. Nonetheless, for extended objects such as the satellite, PCG performs quite well (see Figure 4.4). Thus, in some instances, incorporating nonnegativity constraints provides a significant advantage, but this is not always the case. The effects of incorporating nonnegativity constraints on image reconstruction problems is not well understood and, the authors believe, is something that needs further study.

An observation that we feel is of particular importance comes with a comparison of the WMRNSD and RL algorithms. We note that the statistical argument has always been the main motivation for using RL. As we have shown in this paper, WMRNSD is also statistically motivated, but, as the results in Figures 4.3 and 4.4 show, WMRNSD has superior convergence properties when compared to RL. This suggests that in place of RL, one might consider using WMRNSD. A hybrid of these two algorithms may also be effective.

Finally, we mention that a drawback of using relative error as a measure of reconstruction quality is that the accuracy with which higher frequencies in the object are reconstructed has little effect on its value. This is particularly important given the fact that the imposition of nonnegativity constraints is known to, in many cases, provide more accurate reconstructions of high frequency information. To see that this is the case, in Figure 4.5 we present the reconstructions given by the PCG and PMRNSD iterates that minimize relative error on the Satellite data with 1% noise. Notice that the nonnegatively constrained reconstruction has better resolution even though the corresponding relative errors are nearly equal.

5. Conclusions. In this paper we presented a class of algorithms that can be used to solve nonnegatively constrained least squares and weighted least squares prob-

lems. We provided motivation for using the weighted least squares formulation based on the noise statistics of CCD camera image formation. We saw that in some cases a diagonal preconditioner that incorporates statistical information can be more effective than structure-based preconditioners that attempt to approximate the coefficient matrix.

We showed that the RL algorithm, which is often used by application scientists, can be interpreted as a variable preconditioned version of the modified residual norm steepest descent algorithms considered in this paper. This provides an important link that should prove useful when deciding which algorithm should be used for a specific problem.

In our numerical experiments, we found that of the two statistically motivated algorithms, WMRNSD and RL, WMRNSD has superior convergence properties. This suggests that in place of RL, application scientists might consider using WMRNSD.

Another important attribute of the algorithms presented in this paper is that they enforce a nonnegativity constraint. The performance of the unconstrained PCG algorithm when compared to the MRNSD algorithms discussed in this paper suggest that the nonnegativity constraint can make a substantial difference in both the accuracy of the reconstructions and in the convergence properties of the algorithms.

Acknowledgments. The authors would like to thank the referees for their careful reading of the manuscript, and for the many suggestions they made to help improve the presentation of this work.

REFERENCES

- [1] J. M. BARDSLEY AND C. R. VOGEL, *A nonnegatively constrained convex programming method for image reconstruction*, SIAM. J. Sci. Comput., 25, pp. 1326–1343.
- [2] M. BERTERO AND P. BOCCACCI, *Introduction to Inverse Problems in Imaging*, IOP Publishing Ltd., London, 1998.
- [3] D. CALVETTI, B. LEWIS, L. REICHEL, AND F. SGALLARI, *Tikhonov regularization with nonnegativity constraint*, Electron. Trans. Numer. Anal., 18 (2004), pp. 153–173.
- [4] R. H. CHAN AND M. K. NG, *Conjugate gradient methods for Toeplitz systems*, SIAM Rev., 38 (1996), pp. 427–482.
- [5] H. W. ENGL, M. HANKE, AND A. NEUBAUER, *Regularization of Inverse Problems*, Kluwer Academic Publishers, Dordrecht, 2000.
- [6] W. FELLER, *An Introduction to Probability Theory and Its Applications*, Wiley, New York, 1971.
- [7] J. A. FESSLER, *Penalized weighted least-squares image reconstruction for positron emission tomography*, IEEE Trans. Med. Imag., 13 (1994), pp. 290–300.
- [8] G. H. GOLUB AND C. VAN LOAN, *Matrix Computations*, 3rd ed., Johns Hopkins Press, Baltimore, MD 1996.
- [9] M. HANKE AND J. G. NAGY, *Restoration of atmospherically blurred images by symmetric indefinite conjugate gradient techniques*, Inverse Problems, 12 (1996), pp. 157–173.
- [10] M. HANKE, J. G. NAGY, AND R. J. PLEMMONS, *Preconditioned iterative regularization for ill-posed problems*, in Numerical Linear Algebra, L. Reichel, A. Ruttan and R. S. Varga, eds., de Gruyter, Berlin, 1993, pp. 141–163.
- [11] P. C. HANSEN, *Rank-Deficient and Discrete Ill-Posed Problems*, SIAM, Philadelphia, 1997.
- [12] L. KAUFMAN, *Maximum likelihood, least squares, and penalized least squares for PET*, IEEE Trans. Med. Imag., 12 (1993), pp. 200–214.
- [13] J. G. NAGY, K. PALMER, AND L. PERRONE, *Iterative methods for image restoration: A Matlab object oriented approach*, Numer. Algorithms, 36 (2003), pp. 73–93.
- [14] J. NAGY AND Z. STRAKOŠ, *Enforcing nonnegativity in image reconstruction algorithms*, in Mathematical Modeling, Estimation, and Imaging, David C. Wilson et al., eds., SPIE, Bellingham, WA, 2000, pp. 182–190.
- [15] Y. SAAD, *Iterative Methods for Sparse Linear Systems*, PWS Publishing Company, Boston, 1996.

- [16] D. L. SNYDER, A. M. HAMMOUD, AND R. L. WHITE, *Image recovery from data acquired with a charge-coupled-device camera*, J. Opt. Soc. Amer. A, 10 (1993), pp. 1014–1023.
- [17] D. L. SNYDER, C. W. HELSTROM, A. D. LANTERMAN, M. FAISAL, AND R. L. WHITE, *Compensation for readout noise in CCD images*, J. Opt. Soc. Amer. A, 12 (1995), pp. 272–283.
- [18] R. VIO, J. BARDSLEY, AND W. WAMSTEKER, *Least-squares methods with Poissonian noise: An analysis and a comparison with the Richardson-Lucy algorithm*, Astronomy and Astrophysics, 437 (2005), pp. 741–755.
- [19] C. R. Vogel, *Computational Methods for Inverse Problems*, SIAM, Philadelphia, 2002.