Reliable Algorithm for Modal Decomposition

Robert A. Walker* and Arthur E. Bryson Jr.†
Stanford University, Stanford, California 94305

This paper describes a reliable, general algorithm for modal decomposition in real arithmetic and its use in analyzing and synthesizing control logic for linear dynamic systems. We describe the numerical difficulties associated with computing the Jordan canonical form when the system has repeated, or nearly repeated, eigenvalues. A new algorithm is described that satisfactorily solves these numerical difficulties. We discuss its relation and extension to related numerical analysis research to clarify the reliability of the techniques. And finally, its implementation as a practical modal decomposition method for efficiently computing the matrix exponential, transfer functions, and frequency response is also described.

I. Introduction

OUR purpose in this paper is to communicate in a convincing manner that a reliable, practical engineering tool for the general real eigenproblem is now available. Goals supporting this purpose are the following.

1) Give the engineering user an intuitive and practical familiarity with the numerical problems in computing the Jordan canonical form (JCF) of an unsymmetric matrix A with real entries (hereafter we will use the shorter term "Jordan form").

2) Report on the key research to date relating to a reliable solution to this problem.

3) Point out the innovations in this work that make the algorithm a practical automatic engineering tool rather than an experimental research tool.

4) Review applications for this algorithm, for which modal decomposition has not previously been considered a preferred method.

To achieve these goals, three contrasting emphases are made at various points in the paper, including 1) a tutorial discussion for engineering users vs clarification of the algorithm to numerical analysts, 2) a discussion of the significant work in Kagstrom and Ruhe's complex arithmetic Jordan normal form (JNF) program[1] vs our extensions to their algorithm in the principal vector algorithm (PVA), and 3) discussions of the method of modal decomposition in general vs the defense of the specific algorithm and implementation called PVA.

The real arithmetic implementation in PVA was quite a challenging task in its own right, which saves about half the storage and some of the computation, but PVA is not merely a real version of JNF. The PVA develops the clustering approach beyond that of JNF considerably, and thereby changes the underlying assumption that computing the Jordan form is a research problem to an assumption that we can compute a desirable Jordan form for most practical problems of interest.

Subsequent sections cover the following material: Section II describes how an engineering user will encounter numerically multiple roots, and the difficulty of computing a desirable eigensystem in this situation. Section III introduces the importance of proper root clustering and its effect in separating "artificial" vs "inherent" ill conditioning. Section IV outlines the algorithm with emphasis on the clustering strategy and the reliability that this brings. Section V reviews the advantages for the method of modal decomposition in three control-related applications of interest—the matrix exponential, transfer function matrices, and multiple-input/multiple-output (MIMO) frequency response. Section VI summarizes the emphases developed, focusing on the reliability of the algorithm, based on the soundness of the Kagstrom/Ruhe work and our innovations in the critical clustering steps that make a practical engineering analysis tool possible.

Received March 7, 1990; revision received March 27, 1990.

*Postdoctoral Research Affiliate, Department of Aeronautics and Astronautics; currently, Senior Member of the Technical Staff, FMC Corporate Technology Center, Santa Clara, CA.

†Paul Pigott Professor of Engineering, Department of Aeronautics and Astronautics. Fellow AIAA.

II. Background on the General Real Unsymmetric Eigenproblem

Modal decomposition of a linear time-invariant dynamic system has proven to be a useful design aid for synthesizing control logic, unifying linear-quadratic-Gaussian (LQG) and classical techniques, and yielding valuable physical insight. In general, an unsymmetric eigenproblem may include repeated eigenvalues with coupled modes, otherwise known as a system whose Jordan form is not diagonal.

Aitken completed the mathematical description of the classical algebraic eigenvalue problem in 1930.[2] This theory is available in books such as Gantmacher,[3] Lancaster,[4] Samuelson,[5] or Halmos.[6] The appendix in Strang[7] is particularly clear. The theory is briefly reviewed here before introducing the difficulties in numerically computing the Jordan form and their resolution.

Jordan Form—Principal Vectors and Chains

The modal similarity transformation

$$M^{-1}FM = J \tag{1}$$

can be written as

$$FM = MJ \tag{2}$$

Following Strang,[7] consider a fifth-order system, with two sets of repeated eigenvalues, whose Jordan form is not diagonal. Writing out this set of equations column by column, where m_i is the ith column of M, gives

$$F\left[m_1, m_2, \ldots, m_5\right] = \left[m_1, m_2, \ldots, m_5\right] \begin{bmatrix} \lambda_1 & 1 & & & \\ & \lambda_1 & & & \\ & & \lambda_2 & 1 & \\ & & & \lambda_2 & \\ & & & & \lambda_2 \end{bmatrix} \tag{3}$$

Elements of a column of J form linear combinations of the

columns of M, which we can equate to the left-hand side, therefore

$$Fm_1 = \lambda_1 m_1 \tag{4}$$

$$Fm_2 = \lambda_1 m_2 + m_1 \tag{5}$$

These two equations can be rewritten as

$$(F-\lambda_1 I)m_1 = 0 \tag{6}$$

$$(F-\lambda_1 I)m_2 = m_1 \tag{7}$$

If we multiply the last equation by $(F-\lambda_1 I)$,

$$(F-\lambda_1 I)^2 m_2 = (F-\lambda_1 I)m_1 = 0 \tag{8}$$

In general, a vector that satisfies the chain relation

$$(F-\lambda I)m_i = m_{i-1} \tag{9}$$

$$(F-\lambda I)^i m_i = 0 \tag{10}$$

is called a principal vector of grade i. Thus m_2 is a principal vector of grade 2, and m_1, an eigenvector, is also called a principal vector of grade 1. To further illustrate Jordan blocks and principal vector chains, if λ_2 equals λ_1 in this 5×5 example, then there would be more than one coupled chain associated with the same eigenvalue. Associated with the three chains in this case are the three eigenvectors m_1, m_3, and m_5. There would be two principal vectors of grade 2, m_2 and m_4.

Eigenvalue Clusters, Ill Conditioning, and Scaling

Numerical analysis of algorithms for general modal decomposition began appearing in 1970.[8-10] However, existing, widely used algorithms such as HQR2[11] still make no provision for repeated eigenvalues with coupled modes. The two critical problems are dealing with "eigenvalue clusters" and ill conditioning, i.e., unavoidable high-spectral-condition numbers. A new algorithm is described here that satisfactorily solves the clustering problem, and work on eigensystem refinement[12] can be employed to give improved accuracy where the system model is highly ill conditioned.

For linear time-invariant models, the scaling of the state and control variables so that one unit of each is of comparable engineering significance is valuable for practical analysis as well as computation.[13] With such scaling the relative magnitudes of the eigenvector components are significant. This scaling usually reduces the system ill conditioning. Assuming the engineering model has been meaningfully chosen, the physical significance of eigensystem ill conditioning may still be obscured by the computational technique as described next.

Failure of HQR Algorithm on Systems with Repeated Eigenvalues and Coupled Modes

The HQR2 algorithm[11] finds the real Schur (quasitriangular) form of a matrix and then backsolves for the eigenvectors. When applied to a matrix with repeated eigenvalues, but uncoupled modes, it yields linearly independent eigenvectors with no numerical difficulty. Consider the 2×2 matrix (already in Jordan form):

$$F = \begin{bmatrix} \lambda_1 & 0 \\ 0 & \lambda_1 \end{bmatrix} = J \tag{11}$$

To solve for the eigenvector of this triangular matrix, we backsolve the systems

$$\begin{bmatrix} \lambda_1-\lambda_1 & 0 \\ 0 & \lambda_1-\lambda_1 \end{bmatrix}\begin{bmatrix} v_1 \\ 1 \end{bmatrix} = 0 \tag{12}$$

$$\Rightarrow (\lambda_1-\lambda_1)v_1 + 0\cdot 1 = 0 \Rightarrow v_1 = 0/0 \tag{13}$$

The v_1 is thus arbitrary and can be replaced by zero. HQR2 incorporates a convenient way of realizing this in that whenever called upon to divide by zero, a small number ϵ is substituted, and thus

$$v_1 = 0/\epsilon = 0 \tag{14}$$

The back solution gives the other vector as $[1,0]^T$; hence the modal matrix is I_2, as it should be, since the matrix is already in its Jordan form. Thus equal eigenvalues, or eigenvalues that are very close to being equal, are treated properly by HQR2 provided the modes are totally uncoupled.

Now consider the 2×2 matrix

$$F = \begin{bmatrix} \lambda_1 & 1 \\ 0 & \lambda_1 \end{bmatrix} \equiv J \tag{15}$$

with repeated eigenvalues λ_1 whose modes are coupled; this plant is already in its Jordan form, hence the modal matrix M is a unit matrix. However, backsolving the second vector with the method of HQR2 gives the result shown below

$$\begin{bmatrix} \lambda_1-\lambda_1 & 1 \\ 0 & \lambda_1-\lambda_1 \end{bmatrix}\begin{bmatrix} v_1 \\ 1 \end{bmatrix} = 0 \tag{16}$$

$$\Rightarrow 0\cdot v_1 + 1 = 0 \Rightarrow v_1 = 1/0 \tag{17}$$

where a small number, $-\epsilon$, is substituted for zero. The computed vectors are then

$$\begin{bmatrix} 1 & 1/\epsilon \\ 0 & 1 \end{bmatrix} \text{ and rescaled} \Rightarrow \begin{bmatrix} 1 & 1 \\ 0 & \epsilon \end{bmatrix} \tag{18}$$

These vectors are almost linearly dependent and hence form a poor basis for modal decomposition. However, with only a coupled pair, we can sort them to the top of the quasitriangular form and subtract the two vectors, renormalize, and get the correct principal vectors. Unfortunately, longer chains do not accommodate such a straightforward resolution.

Wilkinson[14] has defined eigenvector condition numbers

$$s_i = y_i^T x_i$$

where x_i is the right eigenvector and y_i the left eigenvector associated with λ_i, both normalized to length 1. Ruhe[9] and Golub and Wilkinson[10] have shown that when s_i is small, the given matrix is necessarily close to a matrix with multiple eigenvalues.

III. Recognizing Repeated Roots Numerically

Thus far, we have discussed matrices either having or not having multiple eigenvalues, as we would in classical mathematics. Numerically, however, there is no such distinction. The crux of the numerical difficulty in developing a useful general eigensolver is how to recognize a "cluster" of roots that should be treated as a multiple eigenvalue. Once recognized, treating them as confluent then eliminates what we will describe in this section as artificial ill conditioning.

Perturbations in Repeated Roots

In the preceding 2×2 example, a perturbation of one element of order ϵ causes a dispersion of the eigenvalues of order $\epsilon^{1/2}$:

$$\begin{bmatrix} \lambda_1 & 1 \\ \epsilon & \lambda_1 \end{bmatrix} \text{ is similar to } \begin{bmatrix} \lambda_1+\sqrt{\epsilon}, & 1 \\ 0 & \lambda_1-\sqrt{\epsilon} \end{bmatrix}$$

Similarly, a perturbation of 10^{-10} in the (10, 1) position of a 10×10 block of repeated eigenvalues coupled with unit values on the superdiagonal gives computed eigenvalues of

$$\lambda_i = \lambda_{ave} + \epsilon^{1/10}\epsilon^{2\pi\sqrt{-1}(i-1)/10}, \qquad \Rightarrow \max|\lambda_i - \lambda_j| \approx 0.2$$

Use of Nearly Linearly Dependent Eigenvectors

Why not just use the nearly linearly dependent eigenvectors? First, using a nearly singular eigenvector matrix amplifies errors in further computations on the order of $\kappa(M)\|A\|u$ where $\kappa(M)=\|M\|\|M^{-1}\|$ = condition number of the transformation matrix M, $\|A\|$ = norm of given matrix A, and u is the machine relative precision. Second, with IBM double precision and $\|A\|\approx 1$, a small number is $\epsilon\approx 10^{-16}$; thus, a 6×6 matrix with exactly repeated roots and coupled modes, with unit coupling or greater, treated as distinct, results in elements of the M matrix of $0(1/\epsilon^5)$, which overflows an IBM computer.

A common engineering fix has been to modify the equations somewhat, in effect supplying a larger ϵ than the algorithm does automatically, until the program no longer fails. This not only gives a less accurate solution but also requires multiple executions, which is often less than satisfactory.

Double pure-imaginary roots with coupled modes yield an unstable model with terms such as $t\cos\omega t$ in the general solution. If these roots are slightly perturbed and treated as distinct ($j\omega$ and $j\omega t+j\Delta\omega$), the solution can only approximate the unstable mode by producing "beats," i.e., terms such as $\sin(t\Delta\omega)\cos\omega t$.

Clustering Removes Artificial Ill Conditioning

Although individual eigenvalues of a perturbed coupled Jordan block are very sensitive to perturbations, their mean value is not sensitive. Kahan[15] has shown that the individual eigenvalue condition numbers of a multiple root are not meaningful and has defined a condition number for the "mean" or multiple root. This condition number is bounded by the norm of the spectral projector matrix for the group.[16] (The spectral projector matrix P of λ projects every vector onto the invariant subspace of λ. $P=X(Y^*X)^{-1}Y^*$, where X spans the space of the right principal vectors associated with λ, and Y spans the space of the corresponding left principal vectors.) Ruhe[9,17,18] has shown that once the clusters are properly determined, the subspaces of $(A-\lambda_{ave}I)$ can be determined within a small multiple of the size of the perturbation. Therefore it is not correct to say that the *problem* of computing principal vectors of a nonlinear divisor is inherently ill conditioned; rather there has not been a well-conditioned *algorithm* to find the clusters. We will discuss this more under Sec. IV on the algorithm. This type of ill conditioning, which results from treating perturbed multiple roots as distinct, can now be characterized as artificial ill conditioning. Conversely, within the finite precision we are working, theoretically distinct roots may be most accurately represented spectrally as a coalesced cluster (consider for example high-dimensional Frank matrices discussed in Ref. 10).

Inherent Ill Conditioning

However, relative to the finite precision we are using, the eigensystem of a set of equations may be inherently ill conditioned. In this case, it is not possible to replace the actual distinct but poorly conditioned eigenvalues by confluent eigenvalues without deleting much larger elements than $\|A\|u=0(\epsilon)$. What could be gained in lowering the overall spectral condition number by assuming confluent roots is lost by imposing an eigenstructure that is much more distant from the true eigensystem than the roundoff errors committed in finding the eigenstructure. Such ill conditioning must be accepted as a true description of the system model. Physical systems that exhibit this type of ill conditioning appear to be extremely unusual, having a high number of almost identical mode shapes for a sequence of modes of considerably different frequencies. (For more on this topic, see Ref. 14, pp. 91-92; Ref. 19, Appendix 2; and Ref. 20, Sec. 2.7.2.)

Simple Example

The following example of computing the matrix exponential illustrates first the importance of proper clustering to avoid artificial ill conditioning and second that an eigensystem with a high spectral condition number can be dealt with such that at the boundary of including another root in the cluster, either decomposition gives comparable accuracy. The example was given by Moler and Van Loan in their survey of "Nineteen dubious ways to compute the matrix exponential"[21] for the system:

$$A=\begin{bmatrix}1+\alpha & 1\\ 0 & 1-\alpha\end{bmatrix} \tag{19}$$

$$\Rightarrow M=\begin{bmatrix}1 & 1\\ 0 & -2\alpha\end{bmatrix},\qquad M^{-1}=\begin{bmatrix}1 & \tfrac{1}{2}\alpha\\ 0 & \tfrac{1}{2}\alpha\end{bmatrix}$$

$$D=\begin{bmatrix}1+\alpha & 0\\ 0 & 1-\alpha\end{bmatrix} \tag{20}$$

$$\exp_d(A)=M\exp(D)M^{-1} \tag{21}$$

and where $\exp(A)$ is exact to machine precision. With $\alpha=10^{-5}$ and using arithmetic rounded to seven significant digits to simulate a machine precision of $u=16^{-5}\approx 10^{-6}$ (IBM short precision), the matrix exponential error is

$$\left[\exp_d(A)-\exp(A)\right]=\left[M_d\exp(D)M_d^{-1}-\exp(A)\right]$$

$$=\begin{bmatrix}-2.E-6 & -4.17E-2\\ 0 & -1.E-6\end{bmatrix} \tag{22}$$

Note that one error is almost five orders of magnitude above the machine precision, approximately by the factor of $\kappa(M_d)=0(10^5)$. However, these eigenvalues are repeated from the standpoint of reasonable perturbations, since the maximum possible perturbation radius, obtained by treating the block as a single chain with each coupling element equal to the matrix norm, is

$$r_p=\left[n\cdot\|A\|_E u\right]^{1/2}=\left[2\cdot 3^{1/2}\cdot\frac{1}{2}10^{-6}\right]^{1/2}=1.32E-3$$

Assuming confluent roots here gives a double eigenvalue at 1. These eigenvalues may be obtained by applying a Jacobi transformation to A. Using a small angle approximation in this example gives

$$\begin{bmatrix}1 & -\alpha\\ \alpha & 1\end{bmatrix}\begin{bmatrix}1+\alpha & 1\\ 0 & 1-\alpha\end{bmatrix}\frac{\begin{bmatrix}1 & \alpha\\ -\alpha & 1\end{bmatrix}}{1+\alpha^2}=\begin{bmatrix}1 & 1\\ 0 & 1\end{bmatrix}+0(\alpha^2)=J \tag{23}$$

Again using arithmetic rounded to seven significant digits

$$\left[\exp_J(A)-\exp(A)\right]=\left[M_J\exp(J)M^{-1}-\exp(A)\right]$$

$$=\begin{bmatrix}2.718309 & 2.718282\\ -2.7E-10 & 2.718255\end{bmatrix}-\begin{bmatrix}2.718309 & 2.718282\\ 0 & 2.718255\end{bmatrix}$$

$$=\begin{bmatrix}0 & 0\\ -2.7E-10 & 0\end{bmatrix} \tag{24}$$

Thus with a proper clustering procedure, modal decomposition gives a very accurate answer for the matrix exponential on this example.

The important place to compare the accuracy of the two decomposition choices, as the value of α is varied in this example, is $\alpha=r_p=1.32E-3$. At this point the diagonal similarity transformation matrix will have the worst possible conditioning we will accept, and the "assumed" confluent roots will admit the largest errors that we will accept by averaging. Treat-

ing λ_1 and λ_2 distinctly gives

$$\exp_d(A) = \begin{bmatrix} 1 & -1 \\ 0 & 2.64E-3 \end{bmatrix} \begin{bmatrix} 2.721872 & 0 \\ 0 & 2.714696 \end{bmatrix} \times \begin{bmatrix} 1 & 378.7879 \\ 0 & 378.7879 \end{bmatrix}$$

$$\exp_d(A) - \exp(A) = \begin{bmatrix} 2.721872 & 2.718286 \\ 0 & 2.714696 \end{bmatrix} - \begin{bmatrix} 2.721872 & 2.721872 \\ 0 & 2.714696 \end{bmatrix} = \begin{bmatrix} 0 & 4.E-6 \\ 0 & 0 \end{bmatrix} \quad (25)$$

The error is less than an order of magnitude above the machine precision.

Assuming confluent roots in this case gives for $\exp_J(A) = M^{-1} \exp(J) M$,

$$\exp_J(A) = \begin{bmatrix} 1 & 1.32E-3 \\ -1.32E-3 & 1 \end{bmatrix} \times \begin{bmatrix} 2.718282 & 2.718282 \\ 0 & 2.718282 \end{bmatrix} \times \begin{bmatrix} 0.9999983 & -1.319998E-3 \\ 1.319998E-3 & 0.9999983 \end{bmatrix}$$

$$\exp_J(A) - \exp(A) = \begin{bmatrix} 2.721870 & 2.718277 \\ -4.7E-6 & 2.714694 \end{bmatrix} - \begin{bmatrix} 2.721872 & 2.718282 \\ 0 & 2.714696 \end{bmatrix} = -\begin{bmatrix} 2.E-6 & 5.E-6 \\ 5.E-6 & 2.E-6 \end{bmatrix} \quad (26)$$

Again, the error is less than an order of magnitude above the machine precision.

The computations were performed on a hand calculator with 10 significant digits, rounding after every operation to seven significant digits. This simple example suggests a higher-order machine solution with similar high accuracy using eigensystem refinement, which is described in Ref. 20. There the matrix exponential function and candidate algorithms are analyzed in much more detail. Our purpose here is to illustrate that modal decomposition is an attractive candidate algorithm for computing the matrix exponential, particularly for perturbed confluent roots. Apart from Kagstrom's work (see Refs. 22 and 23), this seems not to have been recognized. The matrix exponential in this example can be computed to within an order of magnitude of the simulated machine precision regardless of the degree of ill conditioning of A!

IV. Reliable Algorithm

Two Categories of Jordan Form Algorithms

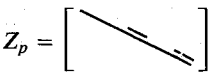
There are two categories of algorithms to compute the Jordan form: those that work from the eigenvectors up the chains to the highest grade principal vectors and those that find the nullspace of powers of the matrix working from the highest grade vectors down the chains to the eigenvectors. Inverse iteration[24] and orthogonal deflation[10] are of the first type, whereas powering the matrix[25] and transformation methods[18,25,26] are of the second type. Of the second type, powering a matrix is not as attractive numerically as transformation methods, which rely on finding a unitary basis of successive nullspaces.

Outline of the Principal Vector Algorithm

Our algorithm uses transformation methods and follows the work of Kagstrom and Ruhe, but uses a more discriminatory deflation procedure to separate roots included by initial overconservative clustering, and implements the algorithm in real arithmetic, for a reduction in storage and a more useful engineering design tool.

Conserving storage is one attraction of transformation methods. Orthogonal or elimination transformations (such as are used in Gaussian elimination) operate on the input matrix sequentially to eliminate all but the Jordan form. Each of these transformations is accumulated by postmultiplication into a storage matrix, which becomes the modal matrix. These elimination matrices merely continue the reduction begun in the initial triangular decomposition step. The process is shown below:

$$Q^t A Q = T = \quad (27)$$



$$Z_p^{-1} \ldots Z_2^{-1} Z_1^{-1} T Z_1 Z_2 \ldots Z_p =$$

$$= J = M^{-1} A M \quad (28)$$

$$M = Q Z_1 Z_2 \ldots Z_p \quad (29)$$

The steps of the algorithm are listed below.

1) Find the real triangular decomposition (Schur canonical form) of the system matrix with the eigenvalues ordered by magnitude.

2) Group potentially perturbed eigenvalues into adjacent locations.

3) Eliminate around the groups to a block triangular form, and orthogonalize the basis of this invariant subspace.

4) Prepare complex blocks for complex arithmetic (a real representation is used consistent with Fortran 77).

5) Determine the Jordan structure of the blocks.

6) Develop principal vector chains by stabilized eliminations.

7) Reassemble complex blocks into real form.

Reliable Exchange Algorithm

Step 1 uses a modification of Stewart's algorithm, HQR3, documented in the Communications of the ACM,[27] an extension of HQR2 and corrected in Ref. 28, which orders the eigenvalues by magnitude. As each eigenvalue is exposed in the lower corner of the upper Hessenberg block, it is exchanged with any previous eigenvalues if it is out of order. Eigenvalues to be exchanged are either a 1×1 block for a real root or a 2×2 block for a complex pair. To exchange adjacent blocks, HQR3 computes a shift from the upper block, doing one arbitrary QR interaction to scramble the two blocks, and then reapplies the QR algorithm with the previously computed shift to draw out the upper block in the lower position. The difficulty is that with nearly repeated eigenvalues, the QR algorithm may not converge on such an attempted exchange. We have found no cases that will not converge initially in 50 iterations of the implicit double-shifted QR algorithm. However, it is not difficult to find examples that will not converge when exchanging repeated eigenvalues with coupled modes.

If the eigenvalues are nearly repeated, why exchange them? To initially find potential clusters of roots, we sort not only by magnitude but also by sign and type. The logic to decide when not to exchange nearly repeated roots would be cumbersome. Grouping the eigenvalues in these ways allows the determination of the clusters with a reasonable amount of computation. Therefore, we have modified HQR3 to make direct (HQR3D) rather than iterative exchanges. Thus, in nearly repeated cases we are only redistributing the errors, but most important, our exchanging algorithm has no convergence problems, as experienced with HQR3.

There are four different exchanges when the matrix is represented in real form. In exchanging two 1×1 blocks, we can easily solve for the sine-cosine plane rotation that exchanges the roots. For an explicit expression of the rotation in the complex case, see Ruhe.[18] However, a constructive way of performing this rotation can be shown,[20] which makes it easier to understand the other three exchanges. All of the exchanges

are based on the observation that in the Schur decomposition, the first orthogonal vector associated with the top eigenvalue is an eigenvector of the triangular matrix.[29] By finding this eigenvector, the rest of the orthogonal matrix (except for sign or order) is specified.

Proper Preliminary Clustering

The accuracy of finding principal vectors depends on properly clustering the eigenvalues. What do we mean by proper clustering? Simply that there is a nearby Jordan form the distance to which is on the order of the size of the backward error in finding the Schur form. A cluster of eigenvalues is properly grouped when they could be "perturbed back" to an exact multiple eigenvalue with a perturbation whose norm is comparable to the norm of the rounding errors introduced by our computations in finding the eigenvalues with a Schur decomposition. The preliminary clustering procedure of PVA extends in ways discussed at the end of this section the approach used by Kagstrom and Ruhe[1,30] using Gerschgorin disks of the ordered Schur form to find potential clusters. The bound expressions for real and complex eigenvalues are shown below. The following symbols are used where K is a disk centered at z of radius r:

$$K(z,r) = \{\zeta : |\zeta - z| < r\} \tag{30}$$

$$A = T + E,\ T \quad \text{upper triangular} \tag{31}$$

$$D = \text{diag}(d_i), \qquad d_i \geq 1, \qquad i = 1,\ldots,n \tag{32}$$

The Gerschgorin circles of DAD^{-1} for each eigenvalue λ_k satisfy

$$G(DAD^{-1},\lambda_k) = K\left(\lambda_k + e_{dd}, d_k\left\{\sum_{1}^{k-1} |e_{ki}|/d_i + \sum_{k+1}^{n} |t_{ki} + e_{ki}|/d_j\right\}\right) \tag{33}$$

Since the $d_i \geq 1$, and by definition of the infinity norm, these circles can be bounded above, such that

$$G(DAD^{-1},\lambda_k) \subset K\left(\lambda_k, d_k\left\{\|E\|_\infty + \sum_{k+1}^{n} |t_{ki}|/d_i\right\}\right) \tag{34}$$

The procedure is then to construct a circle around the potential cluster reaching halfway to the nearest external neighbor. Circles are drawn tangent to this "cluster circle," and the radius $r_k (k = n-1,\ldots,1)$ is equated to

$$d_k\left\{\|E\| + \sum_{k+1}^{n} |t_{ki}|/d_i\right\} \tag{35}$$

to solve for d_k, where for $\|E\| = \epsilon$ we use $10n\|A\|u$ as an approximation to the backward error in the Schur decomposition. If all $d_k \geq 1$, then the cluster is complete. The procedure starts with a single root and adds members until the bound relationship yields $d_k \geq 1$ $(i = 1,\ldots,n)$.

Feingold and Varga[31] have given a theorem that generalizes the Gerschgorin circle theorem. This is necessary for our real form matrix with complex eigenvalues. For a partitioned matrix A, each eigenvalue of A satisfies

$$\|(A_{jj} - \lambda_j I)^{-1}\|^{-1} \geq \sum_{k=1,\, k\neq j}^{n} \|A_{jk}\| \tag{36}$$

for at least one j, $1 \leq j \leq n$, where for our purposes the partition will be 1×1 for real roots and 2×2 blocks for the complex pairs. The relation

$$\left\|\left\{\begin{bmatrix} \sigma_j & \omega_j \\ -\omega_j & \sigma_j \end{bmatrix} - \lambda_j I_2\right\}^{-1}\right\|^{-1} \geq \sum_{k=1,\, k\neq j}^{n} \|A_{jk}\| \tag{37}$$

defines two disjoint disks in the complex plane, centered on the pairs $\sigma_j \pm j\omega_j$. These disks bound the Gerschgorin disks associated with a $0(\|E\|)$ perturbation as follows:

$$G(DAD^{-1},\sigma_k \pm j\omega_k) \subset K\left(\sigma_k \pm j\omega_k, d_k \times \left[2\|E\|_\infty + \sum_{j=k+1}^{n} \|T_{ki}\|_\infty/d_i\right]\right)$$

T_{ki} corresponds to 1×1, 1×2, 2×1, or 2×2 blocks of T [see Eqs. (36) and (37)]. This bound can now be used in exactly the same way as described for the real case to cluster those roots that potentially will coalesce with a $0(\epsilon)$ perturbation.

The Gerschgorin bound test is attractive because it explicitly takes into account the exact coupling of all superdiagonals and their effect on the computed (perturbed) root locations. However, the approach is not perfect because it assumes that the current cluster mean is the true cluster mean, which in general will not be true until all and only the roots of the desired cluster are included. In other words, the test can be, on a specific numerical example, either underconservative or overconservative in estimating the cluster size and members. Of particular concern is the mistake of underestimating the cluster size, since with a single pass algorithm there is no way to recover from this error. Therefore, the Gerschgorin bound test must be augmented with an always conservative check, which prevents it from being satisfied with too few roots, and thereby dissecting a cluster that could be otherwise accurately deflated later with the singular value decomposition (SVD). Discriminatory deflation at this later stage corrects overconservative clustering done in the initial cluster determination.

The overconservative test for completed cluster size is based on a bound on the maximum possible perturbation radius for the cluster indicated complete by the Gerschgorin test. Namely for the k roots in the potential cluster, we assume that all the superdiagonal elements were a single chain, each having a value of the norm of the present matrix, as follows:

$$r_p = \left[\|A\|^{k-1} 10n \|A\| u\right]^{1/k} = \|A\|(10nu)^{1/k}$$

This perturbation radius is then used with the Gerschgorin bound test in the following algorithm.

Pseudocode for Initial Clustering Algorithm

```
[Starting in a position (I,I) as the bottom corner of a potential cluster]
FOR J = 2, N - I
[Consider roots sequentially up to the top corner of a potential cluster]
Sort down the closest root from above adjacent to the previous test cluster
Evaluate Gerschgorin bound test
IF (Gerschgorin test is met)
  THEN
    Check radius of last root relative to perturbation radius bound
    IF (last root within perturbation radius)
      THEN
        Retain last root and go consider another one
      ELSE
        Drop last root and take previous group as completed
  ELSE
    Retain last root and go consider another one
```

As described previously with the Gerschgorin bound equations for both real (1×1) and complex (2×2) diagonal blocks, the PVA clustering algorithm is double of that previously described, with some interleaving of logic between the real and complex cases, which is omitted here for simplicity.

Efficient Storage

This initial clustering may be overconservative, but still yields small groups (say of j eigenvalues) relative to the potential matrix dimension of n. Further modification and reduction of the $j \times j$ blocks associated with the j eigenvalues including all complex arithmetic can be done in auxiliary $j \times j$ matrices, requiring only slightly more than $2n^2$ locations for larger n, since the input matrix and the final principal vectors are real matrices. In other words the storage requirement for PVA, without auxiliary vectors, is $2n^2+4.5j^2$ instead of the $4n^2$ in the complex case.

Orthogonal Reducing Subspaces

After eliminating to block triangular form, the vectors associated with block k satisfy the equation

$$AX_k = X_k T_{kk} \tag{38}$$

where the j columns of X_k span the invariant subspace associated with λ_k, and T_{kk} is the restriction of A to the $j \times j$ invariant subspace. By orthogonalizing the matrix X_k, we can find a well-conditioned basis for this invariant subspace as explained by Kagstrom and Ruhe[1]

$$X_k = Q_k R_k \tag{39a}$$

$$AQ_k = Q_k R_k T_{kk} R_k^{-1} \tag{39b}$$

Therefore

$$AQ_k = Q_k T'_{kk} \tag{40}$$

where

$$T'_{kk} = R_k T_{kk} R_k^{-1}$$

and T'_{kk} will be further reduced.

Discriminatory Rank Deflation Using the Singular Value Decomposition

Final reduction to the Jordan form is accomplished by rank deflation with the SVD (see Ref. 1). There are several situations, however, when a straightforward application of rank deflation will not give a reduction that is accurate to $0(\epsilon)$. These we have called pathological eigensystems; they exhibit a small "gap" in the singular values between some near zero and some also small but significant [e.g., $0(\epsilon^{1/2})$] singular values as well. The most common of these situations results from an overconservative clustering of roots by the Gerschgorin strategy, which cannot discriminate isolated roots that are close or within the region of the complex plane occupied by roots from one or several perturbed Jordan blocks and yet have nothing to do with these perturbed Jordan blocks. Namely, the dispersed roots are related to each other geometrically in that a $0(\epsilon)$ deletion and rotation of the subspace will make them coalesce, whereas the troublesome roots would require much larger deletions.

The difficulty that isolated roots cause in pulling off the true cluster mean or shift parameter λ used in

$$B_{kk} = T_{kk} - \lambda I \tag{41}$$

can be used to identify these roots. Namely, the isolated root location is not nearly as sensitive to perturbations as the individual roots of a true cluster are. Hence there will be a $0(\epsilon)$ singular value for each chain of the true cluster with the shift parameter anywhere in the entire region of the complex plane of the roots under consideration, whereas the isolated root will give a $0(\epsilon)$ singular value only if the shift parameter is within $0(\epsilon)$ of the specific root. In summary, if we shift on a computed eigenvalue and get a lower rank by one than resulted initially when shifting on the overall mean (one small but significant singular value disappears), then this eigenvalue estimate is an isolated (or unrelated) root with respect to the numerically true cluster. Note that an uncoupled root, a 1×1 small Jordan block whose value is at the cluster mean, will *not* get separated, because shifting on its value does not drop the number of small but significant singular values, since it is at the numerically true cluster mean. The separation gap between "zero" and "nonzero" has been set to 1000, as in Ref. 1, slightly more than $\epsilon^{-1/4}$ for most double precision floating point representations.

Shifting on isolated roots not only yields their location, but by using right singular vectors V in a similarity transformation, we can sort the root to the top of the triangular block in a manner that allows elimination between it and the rest of the cluster while maintaining a low condition number (see Sec. 2.6 of Ref. 20). This shifting only needs to be done when small but significant singular values exist, otherwise this extra work is bypassed.

The second type of pathological eigensystem, evidenced by small but significant singular values (i.e., a small gap in the singular values) during deflation, is caused by chains with coupling elements the size of these small but significant singular values. The rank does not drop when shifting on individual roots in this case, because there is not a diagonal element error but merely weak superdiagonal coupling. Straightforward rank deflation (as in Ref. 1) cannot help but introduce errors of an order

$$\frac{\max_i \tau_i}{\min_{i,j} |\sigma_j - \tau_i|}$$

where σ_j and τ_i are the retained and deleted singular values, respectively. The denominator is the condition number for singular subspace splitting of the SVD given by Stewart,[32] and the numerator is an approximation to the input perturbation. Thus the expression estimates the norm of the corrective rotation required in both U and V of the deflating SVD, $U\Sigma V^*$. Fortunately these errors in the singular subspaces with a small gap show up during the deflation process as small nonzero elements in the deleted portion of the deflated matrix, namely

$$B_{kk} = U\Sigma V^* \Rightarrow V^* B_{kk} V = V^* U\Sigma = (V^* U)\Sigma I$$

The last expression shows that $V^* U$ can be viewed as an SVD of the transformed block, and the upper left corner of this new "U" is a basis of the range of the last deflated portion and must therefore be zero. Thus, any nonzero elements can be zeroed with complex Givens rotations, which when applied like V, as a similarity transformation to B_{kk}, correct the perturbed V and thereby refine, in process, the rank deflation procedure. The contrasts of the next section give further insight into the importance of discriminatory deflation and the initial clustering.

Comparison to the Kagstrom/Ruhe Algorithm[1,30]

The work by Kagstrom and Ruhe on this problem can be called significant for a number of reasons. We highlight particularly their idea of using Wilkinson's Gerschgorin bound formulation of the eigenvalue sensitivity problem in a reverse fashion to decide when a cluster is complete,[14] their incorporation of Kahan's condition estimate for a cluster of roots,[15] their use of reorthogonalization of the bases of individual clusters, their implementation of the algorithm of Kublanovskaya for deflation of the clusters with the singular value decomposition,[25] their development of code for stabilized eliminations of the deflated blocks, and their use of transformation methods and well-organized code throughout, which requires very little extra storage and provides considerable problem conditioning information.

The development of PVA also followed a similarity transformation aproach (in real form) independently of the Kagstrom/Ruhe work at the outset, but it was the rank deflation

problem of confluent blocks that brought the Kagstrom/Ruhe algorithm and program JNF to our attention. There is really nothing about their approach that we would criticize, as far as it goes! It is just that to become a practical engineering tool, the algorithm must address the clustering problem fully, and that has been the focus of the work presented here, which required significant extensions to the Kagstrom/Ruhe algorithm.

Design of a preliminary clustering procedure that is conservative, but not overly conservative, was essential; otherwise there is no way to recover from underconservative errors at this stage. Kagstrom and Ruhe advise recomputing with a larger clustering tolerance. This, however, does not guarantee collecting the missing parts of the previously dissected cluster, and it definitely searches for a more distant Jordan form than the errors inherent in the computed triangular form.

Design of a discriminatory deflation procedure was also essential to handle eigensystems with pathological [e.g., $0(\epsilon^{1/2})$ —small but too significant to delete] singular values. They are dealt with either by splitting off isolated roots from the cluster during deflation or by refining the singular vectors during the deflation process in the case of small but significant coupling terms. Otherwise there will be deletions required on the order of the pathological singular values, and no experimentation with different clustering or deflation tolerances will fix it.

It should be pointed out that there are conceivable eigenstructures for which our algorithm will also have to make deletions much larger than an estimate of the errors committed thus far in the algorithm. Interlocking clusters and chains with two adjacent pathologically small chain elements are two problematic structures, discussed at some length in Sec. 2.6 of Ref. 20, but are so unlikely in physical dynamic systems as to not warrant the extra complexity required to handle them. In these cases we will accept a more distant but well-conditioned Jordan form automatically.

In the weakly coupled case, eliminations on the order of the norm of the block divided by the weak coupling term can give a very high condition number. The stabilized eliminations of Kagstrom and Ruhe are a type of partial pivoting, which we have redesigned to use a type of full pivoting for the eliminations required when the prospective condition of some of the single element eliminations is high. In the weakly coupled case, the elimination involved can be many orders of magnitude lower. This helps minimize errors accumulated into the principal vectors, regardless of what further rescaling of these vectors is done. Kagstrom[33] describes the nonuniqueness of the similarity transformation and points out that a final diagonal rescaling of the columns of M making them equal in the 2 norm gives minimal condition. This final rescaling should be added to future revisions of the current PVA codes.

Certainly all of the algorithmic and programming details necessary to implement PVA in real arithmetic were challenging and save about half the storage required by JNF, but the real motivation was that current control problems of interest start out in real form, and the answers are easier to interpret in real form. All of the deflations, whether for real or complex blocks, are done with the LINPACK complex SVD, but this is on relatively small blocks, and these intermediate results (the deflation singular values) are suppressed from the user unless diagnostics are requested.

Clearly the choice of tolerances at both the initial and final clustering decision points is crucial in the success of the overall algorithm. Therefore we next summarize the recommended design for computed tolerances in PVA and then discuss the underlying philosophy for their estimation. The spectrum of tolerances is as follows:

$$0, \ldots, \delta = \epsilon, \ldots, \beta_d, \ldots, \beta_{wc}, \ldots, \beta_{ir}, \ldots, \sigma_{\max}$$

where $\delta = \epsilon = \| E \|_\infty \approx 10n \| A \| u$ is the tolerance for both deflation and initial clustering, an estimate of the backward error in the Schur decomposition (which also carries forward because of the unitary transformation).

The $\beta_d = \max[10^3, \kappa(\lambda_{ave})]\delta$ is the tolerance gap for deflation. It is an estimated bound on the roundoff errors in the block to be deflated, based on the condition of the block as estimated by the spectral projector norm (which can be computed during the elimination to block triangular form).

The $\beta_{wc} = \sqrt{2\beta_d \sigma_{\max}}$ is the tolerance for correction of weakly coupled roots during the deflation process. Proportionality of the geometric mean between the deletion tolerance and the cluster norm is a good break point. Since, if we consider the gap between β_d and β_{wc}, splitting the subspaces at β_{wc} gives by Stewart's bound relationship[32] that the size of the corrective rotations should be less than twice the perturbation errors times the condition number for splitting the subspace:

$$\frac{\beta_{wc}}{\sigma_{\max}} \approx \frac{2\beta_d}{|\beta_{wc} - \delta|}$$

The $\beta_{ir} = \| A \| (10nu)^{1/k}$ is the tolerance for checking for isolated roots during deflation. The size should be below the perturbation radius cutoff in the initial clustering procedure of this $k \times k$ block.

Present versions of the PVA code have not currently implemented the complete set of these relationships, and it is a statement for the robustness of the clustering approach that we have had very reliable success with the gap between δ and β_d fixed as 10^3 and β_{wc} and β_{ir}, a few orders of magnitude larger than β_d based on empirical studies. It should be clear from the preceding discussion that the philosophy in the design of PVA was to estimate the errors involved at the two clustering stages, using a backward error estimate for the work done so far and the forward sensitivity estimate for the problem at hand. In other words, the tolerance selection should not be required of the user in repetitive program executions, but is part of the algorithm (the code does have a separate tolerance for data accuracy useful for clustering vs machine precision useful for convergence checks).

Since 1980–81, when JNF[30] and PVA[20] were developed, most of the subsequent research has focused on condition estimators for the separation of different clusters of eigenvalues. Wilkinson's work[34,35] is an exception to the condition bound approaches where Wilkinson works directly with rank one perturbation matrices, which make the two most sensitive eigenvalues coalesce. Demmel[36] shows how to compute a stable decomposition of an uncertain matrix T, which varies continuously and boundedly as T varies in a ball of radius ϵ. He defines a criteria, called dissociation (diss) as the smallest perturbation that makes eigenvalues from one block of a triangular matrix coalesce with one from another block. Using the separation (sep) function between operators (Stewart[32]), he develops upper and lower bounds for the diss function. Such a function is a more precise way to do the initial clustering of eigenvalues than the Gerschgorin and maximum perturbation bound approach we use. However, it should be pointed out that finding different combinations of eigenvalues and computing their estimated dissociation is quite computationally intensive in comparison to our overconservative but efficient method. Demmel and Kagstrom[37] extend the concepts to the generalized eigenvalue problem including additional perturbation analysis. The equivalent operator separation estimator for the $A - \lambda B$ problem is the dif^{-1} estimator. Kagstrom and Westin[38] have shown computed results on the generalized Sylvester equation, which are applied to the standard eigenvalue problem in Kagstrom and Poromaa.[39] Codes are being developed for this estimation,[40] which are in the spirit of the earlier Lyapunov equation estimator by Byers.[41] The incorporation of such estimators into an algorithm like JNF or PVA has not to our knowledge been considered, although refinement of invariant and deflating subspaces without the use of extended precision (as we do with our deflation of weakly coupled chains) is under development by Kagstrom and

Westin. In summary, excellent theoretical work on the clustering problem has been done, codes are just being developed, and they are quite computationally expensive.

Guidelines on Reliability and Accuracy Interpretation

Having outlined the PVA and having discussed the critical numerical issues, we can now comment on its reliability and accuracy indicators. A reliable algorithm is one that 1) describes clearly what it has done, including an accuracy estimate; 2) describes the nature of the underlying problem; and 3) does what you expect on the problem, given the problem type.

As pointed out in the previous subsection, we expect PVA to find an eigenstructure without deleting elements larger than the estimated roundoff errors committed up to the deflation points. Interlocking clusters are an exception that would require deletions on the order of the separation of the individual cluster means.

An extreme form of unreliability would be failure—the inability to complete the algorithm and hence the preceding three characteristics. The only cause of actual failure of PVA then would be the lack of initial convergence of the QR algorithm. Although convergence of the unsymmetric shifted QR algorithm has not been proved,[42] there are no known examples where it will not converge in a reasonable number (30–50) of iterations (with the ad hoc scrambling shifts). However, we have experienced a number of examples where using the QR algorithm to order coupled complex eigenvalues can easily cause a lack of convergence. Therefore the direct exchanges just mentioned add considerably to the overall reliability. The user would clearly be informed of a lack of initial convergence of the QR, although no such cases have been found in practice. The clustering procedure is reliable, the initial bound relationships ensuring that we will never underestimate the size of a cluster, and the SVD revealing the distance in the spectral norm, to the Jordan forms under consideration, during the discriminatory rank deflation step.

Upon completion, as in Ref. 30, the size of the deletions (Euclidean norm of all deleted singular values considered zero during all rank deflation steps of a block) together with the estimated condition of the clusters from their spectral norms, can give an indication of how well we found the best structure within rounding errors committed. Singular values involved in the discriminatory deflation can be printed out if there is any question of our choice of the best structure. These reveal both the nature of the underlying problem and chronicle the algorithm choices. The total deletion norm should be comparable to the residuals $\|AM-MJ\|$ or the more conclusive check of $\|M^{-1}AM-J\|$. As Kagstrom and Ruhe[1] point out, since the transformation-based algorithm returns both the cluster means and actual computed eigenvalues, their difference should not be larger than what we deleted for that cluster, and retained coupling elements should not be smaller than what we deleted. With the tolerances for gaps given previously, none of these postprocessing inconsistencies should occur except with highly unlikely interlocking clusters, for which we have not designed. Even in this case, we delete the intercluster distance errors to give a well-conditioned Jordan form that cannot be improved upon by further executions with different tolerances.

In summary, PVA outputs describe clearly what it has done, they give information on the underlying problem structure, and the algorithm does what we expect on each problem type. Therefore, PVA can be considered a highly reliable algorithm implementation for modal decomposition.

In terms of accuracy, one of the advantages of the use of modal decomposition in applications is that the accuracy of the application results can be estimated from the condition of the eigensystem. For some applications M^{-1} will be formed either explicitly, or $Mx=b$ solved, such that the overall condition of M can be estimated.[43] The $\kappa_2(M)$ can be found, although the SVD of M for large M is unnecessarily expensive, so that $\kappa_\infty(M)$ or $\kappa_E(M)$ are recommended, since $\|\ \|_\infty$ or $\|\ \|_E$ are much less expensive to compute. Where a matrix function can be evaluated via modal decomposition and has a condition number bound based on the spectral condition (for example, see the matrix exponential in the next section, as well as Refs. 20 and 22), then if the estimated accuracy with modal decomposition is acceptable (with or without eigensystem refinement), there is no need to check other matrix function condition estimates.

V. Applications to the Analysis of Linear Systems

At this point we would again differentiate between advocacy of the method of model decomposition and a specific algorithm and implementation called PVA to compute it. Given the reliability of PVA already described, we can now consider several important problems in linear systems analysis and discuss the merits of modal decomposition for solving them. Three applications are described here including computing the matrix exponential, computing transfer function representations via the residues, and computing MIMO frequency responses. We also briefly describe or reference other leading methods and compare them to modal decomposition in terms of reliability, accuracy, efficiency, and storage requirements.

Matrix Exponential

Computation of the transition matrix Φ and the zero-order-hold matrix Γ is an essential element in synthesizing digital controllers. If the continuous dynamic system is

$$\dot{x} = Fx + Gu \tag{42}$$

then the discrete-step (digital) system with zero-order hold is

$$x(k+1) = \Phi x(k) + \Gamma u(k) \tag{43}$$

where

$$\Phi = \exp(Ft_s) \tag{44}$$

$$\Gamma = \int_0^{t_s} \exp(Ft)G\ \mathrm{d}t \tag{45}$$

Modal decomposition with the principal vector algorithm performed with superior accuracy and reliability on a range of difficult types of problems compared with other leading algorithms (see below) for computing Φ and Γ when eigenvalue refinement is used (see Ref. 12 for eigensystem refinement and Ref. 20 for matrix exponential refinement). With a small amount of extended precision arithmetic, the accuracy of the computed matrix exponential can be doubled. If the eigensystem spectral condition is less than one over half the machine precision (10^8 for IBM double precision), $\exp(At)$ can be computed exactly to working precision! In the nomenclature of Eq. (1)

$$\Phi = M\exp(Jt_s)M^{-1} \tag{46}$$

$$\Gamma = M\int_0^{t_s} \exp(Jt)\,\mathrm{d}t M^{-1}G \tag{47}$$

where J = Jordan form of F and M = modal transformation matrix. Comparisons were made in Ref. 20 to Ward's Pade[44] and Parlett's[45] block triangular approaches. Bavely and Stewart's[46] single element condition limiting block triangularization could be used for the initial triangularization and ordering. It does not guarantee a well-conditioned overall transformation but can be improved with Kagstrom/Ruhe's reorthogonalization.[1] Alternatively one could use our block triangularization and then apply the triangular function evaluation by Parlett. This compromise between the Jordan (canonical) form and the Schur (canonical) form approaches to the $\exp(At)$ problem is

analyzed by Kagstrom.[22] A is reduced to block diagonal form where each block corresponds to a cluster of close eigenvalues. The Jordan structure of each cluster can be determined by keeping the orthogonal basis for the corresponding invariant subspace. Note that only codes for refinement of distinct eigenvalue/eigenvector pairs have been developed to date, but these codes could be used to handle cases that would have reasonable-sized confluent blocks, though embedded in a fairly large (say 50–100th order) matrix.

Modal decomposition is one of the more reliable algorithms for the matrix exponential with respect to estimating the accuracy of the computed result. The method also provides two bounds on the accuracy achievable with any other algorithm (JCF and Schur bounds for the matrix exponential condition number estimate). The minimum of these two bounds together with the lognorm bound should give a helpful estimate of the underlying matrix exponential problem sensitivity (see Kagstrom[23] and Van Loan[47]). Computational efficiency and storage are comparable to the Pade method.

Transfer Function Matrices

An essential aid in analyzing a linear dynamic system is the determination of transfer functions from the q inputs u_j (controls or disturbances) to the p outputs y_i. If Eq. (42) is the dynamic system and the vector of outputs is

$$y = Hx \tag{48}$$

then the transfer function matrix $T(s)$ is defined as

$$y(s) = T(s)u(s) \tag{49}$$

$$T(s) = H(sI-F)^{-1}G + L \equiv \frac{N(s)}{\Delta(s)} \tag{50}$$

where

$\Delta(s)$ = characteristic polynomial of F
$N(s)$ = matrix of numerator polynomials

There are three different ways to express an individual element of the transfer function matrix $T_{ij}(s)$:

$$\frac{b_0 s^n + \ldots + b_{n-1}s + b_n}{s^n + a_1 s^{n-1} + \ldots + a_n} \tag{51}$$

$$K_{ij}\frac{(s-z_1)(s-z_2)\ldots(s-z_m)}{(s-\lambda_1)(s-\lambda_2)\ldots(s-\lambda_n)}, \qquad m \le n \tag{52}$$

$$\frac{R_1}{(s-\lambda_1)} + \cdots + \frac{R_n}{(s-\lambda_n)} + L_{ij} \tag{53}$$

The first representation, a ratio of unfactored polynomials, does not yield much insight. The second representation, using poles and zeros, allows one to sketch the frequency response with a Bode plot or, by plotting the poles (λ_i) and zeros(z_i) in the complex s plane, to see which modes dominate the dynamic response. The third representation, with the residues R_i, also allows one to see which modes dominate the response and readily gives the time response to an impulsive input as a sum of modal responses. We shall discuss the methods, the efficiency, and the accuracy involved in obtaining these representations of the transfer function for MIMO systems.

Polynomial coefficient matrices can be obtained from the Leverrier/Fadeeva algorithm. The zeros of the individual transfer functions can be computed, without extraneous zeros, by the backward error stable algorithm developed by Emami-Naeini and Van Dooren.[48] The system matrices A and B, where g_j and h_i are a column and row vector from G and H, respectively,

$$A = \begin{bmatrix} F & g_j \\ h_i & L_{ij} \end{bmatrix}, \qquad B = \begin{bmatrix} I_{nxn} & 0 \\ 0 & 0 \end{bmatrix} \tag{54}$$

are formed and reduced with Householder transformations when possible, before applying the QZ algorithm on the reduced system. The residues can be computed (see Ref. 20) via PVA, with or without refinement of ill-conditioned eigenpairs, from the modal input and output matrices $H_m = HM$, $G_m = M^{-1}G$.

Efficiency is probably the most important criterion in computing transfer functions since their use in applications often does not require high accuracy. Of course, reliability is also important; the algorithm should definitely not fail catastrophically. We compare the efficiency of three algorithms for computing transfer function matrices; Table 1 summarizes the operations (multiplications/divisions) required for each algorithm. It is clear that the Leverrier/Fadeeva method is more costly in operations for matrices of a dimension greater than 18–25. Moreover, it has been one author's experience[49] that this algorithm deteriorates rapidly in accuracy for matrices greater in size than $n \cong 20$, even when running the recursion formulas[50] in both directions (Univac 1108 double precision). Thus this classical approach of computing transfer function polynomial coefficients is outdated today on the basis of both accuracy and reliability. We finish this subsection discussing the accuracy and efficiency of the other methods.

For modal decomposition the accuracy of both the eigenvalues and the residues (computed from the principal vectors) is of $0[\kappa(M)\|A\|u]$. In computing individual transfer function zeros, the QZ algorithm faces the same problems PVA does, since the generalized eigenvalue problem $A-\lambda B$ can have vector chains associated with both A and B,[51] although in this case B has a singular, but simple structure, as shown in Eq. (54). For an example of the increased sensitivity of roots from a generalized Jordan form see Emani-Naeini and Van Dooren[48] (example 8). Note that it is the backward error that is stable and bounded at a small value for this algorithm. The forward problem sensitivity of computing individual roots requires a clustering strategy similar to that of PVA to deflate the generalized subspaces without making deletions (in pathological cases) far larger than the bounded backward error.

When comparing the use of zeros vs residues for computing the matrix transfer function, we see that for one to three input/output transfer functions the methods may be comparable, but for more than three transfer functions, the residue approach requires far less operations.

Multi-Input/Multi-Output Frequency Response

Multi-input/multi-output frequency response is useful in evaluating design tradeoffs among good performance, disturbance rejection, insensitivity to sensor noise, and plant uncertainty.[53,54] From the frequency response of one output to disturbance and control inputs, the signal-to-noise ratio can be computed in particular frequency bands. The frequency response can also be used to approximate the system with a lower-order linear system in a particular frequency band.

Frequency response can be computed in two ways: by substituting frequencies for the Laplace operator into the matrix differential equation and solving for the complex response, or by decomposing the system equations in some way to obtain an expression for each transfer function as a rational function of the Laplace operator s. A brute-force attack on the first method would be to find the LU decomposition of the $n \times n$ system, which requires $0(n^3)$ operations at each frequency point. Laub[55] has developed a Hessenberg reduction of the system matrix before doing the LU decomposition requiring approximately $\frac{1}{2}(q+1)n^2 + pqn$ complex operations per frequency point (with p outputs and q inputs).

Table 2 summarizes the operations (multiplications or divisions) required to determine MIMO frequency response using three reasonable methods currently available. One could also compute frequency responses from the poles and single-input/single-output (SISO) zeros, but the computational cost would be prohibitive at $0(pqn^3)$.

Table 1 Operations required to compute the transfer function matrix: order of system = n; number of outputs = p; number of inputs = q

Method	Leverrier/ Fadeeva	Zeros via QZ[a]	Modal decomposition via PVA[b]
Determination of the constants	$n^4+\max(p,q)n^3-2n^3$	$6n^3 \mid 25n^3$ per SISO transfer function	$18n^3 \mid 16n^3+n^4/6$

[a]The first count assumes each transfer function has only $n/2$ zeros so that the subspace can be deflated with Householder reflections.[48] The second count is for $n-1$ zeros.
[b]The first count is for distinct eigenvalues, the second, for one chain of height $n/2$. Note that the $0(n^4)$ term comes from the Jordan block of dimension $n/2$. Recently Beelen and Van Dooren[52] have presented a $0(n^3)$ algorithm for the worst case based on QR decompositions for row and column compressions. The algorithm is of course more efficient but not as reliable as the SVD approach.

Table 2 Operations required to compute MIMO frequency response: order of system = n; number of outputs = p; number of inputs = q

Methods	Leverrier/ Fadeeva	Hessenberg linear system solution (Laub)	Modal decomposition via PVA[a]
Evaluation of one transfer function at one frequency	$n-1$	n^2+n	$n \mid (3n/2-1)$
Evaluation of pq transfer function at k frequencies	$n^4+(pq+1)kn$	$[\frac{1}{2}(q+1)n^2+pqn]k$	$18n^3+pqnk \mid 25n^3+pqnk/2$ $(n<52)$[a]

[a]The first count is for distinct eigenvalues, the second, for one chain of height $n/2$. Note that for $n>52$, the $k>26$ large Jordan block would give a $0(n^4)$ term of meaningful size. In practice, with large matrices, the Jordan blocks would be much smaller than $n/2$.

Beyond the operation counts, a storage penalty (or input/output to disk) results from saving the n, $p \times q$ numerator matrices for later use when working with an individual transfer function. Thus, the amount of work, the susceptibility to roundoff errors, and the awkward implementation suggest that computing zeros, residues, and the frequency response from a ratio of polynomials [Eq. (51)] is not a desirable approach.

As a final point on accuracy, the refinement process described in Ref. 20 could also be used to form more accurate frequency responses. With several auxiliary extended precision vectors, all of the necessary products could be formed in extended precision before rounding the final residue to working precision. We acknowledge that using refinement for this extra accuracy would require some assembly level routines on some computers.

For many applications accurate values of the zeros, the residues, and the frequency response are not required. Therefore, for a control design package, a reasonable combination of algorithms to compute transfer functions would include computing the residues for all desired transfer functions via modal decomposition and using the *QZ* algorithm to compute zeros for selected primary SISO transfer functions. The *QZ* is also extremely useful in computing the system or MIMO zeros.[48] For efficiency and simplicity of implementation, the frequency response could be computed in all cases from the residues. Using the residues to compute the frequency response is particularly efficient when computing the power spectral density (PSD) of a controlled system,[20] since the PSD is an aggregate of a number of individual transfer function products with noise spectral densities.

Storage requirements for computing the frequency response with PVA are about half of that used in the Hessenberg approach since the initial decomposition is done in real rather than complex arithmetic.

VI. Discussion and Summary

A new algorithm for reliably decomposing a linear dynamic system into modal form, the principal vector algorithm (PVA), has been described. The PVA handles repeated, or nearly repeated eigenvalues that previous algorithms, such as HQR2, do not. The algorithm is based on recognizing when a cluster of nearly repeated eigenvalues may be treated as exactly repeated one-way-coupled modes and then calculating the well-conditioned principal vectors instead of the nearly linearly dependent eigenvectors of the distinct but nearly repeated eigenvalues in the cluster.

The guiding philosophy in the design of the numerical methods to deal with the structural ambiguities possible in computing the Jordan form has been 1) to not delete elements larger than the estimated roundoff errors at that point and 2) to minimize the condition number of the transformations applied throughout the reduction of the input matrix. We cannot guarantee that we find the minimal condition, but have taken considerable effort to strive for the condition in all of the types of ill conditioning an engineering user would see.

Beyond the developments of the Kagstrom/Ruhe algorithm, these efforts include initial conservative clustering, shifting on isolated roots during deflation to remove them when they appear as a small "gap" in the singular values, refining or purifying the invariant subspaces when they require it because of pathologically small coupling values, and pivoting between vertical and horizontal directions to maximize the pivot element while doing stabilized eliminations of the triangular blocks to the coupled chains.

With regard to reliability, PVA outputs describe clearly what it has done, they give information on the underlying problem structure, and the algorithm does what we expect on each problem type. Therefore, PVA can be considered a highly reliable algorithm implementation for modal decomposition.

The steps outlined for PVA in Sec. III include a number of stand-alone algorithms, apart from the clustering logic and condition improving innovations. The algorithms include the standard real unsymmetric *QR*, the quasitriangular matrix eigenvalue exchange algorithms, the modified Gram-Schmidt orthogonalization, and the singular value decomposition. The important point is that although the PVA code is considerably larger than HQR2 or HQR3, the computational overhead is minimal in the distinct case. In other words, the extra algorithms are only invoked when required by the input matrix eigenstructure.

For modal decomposition applications requiring high accuracy, eigensystem refinement, with a small amount of extended precision arithmetic, can be used to double the accu-

racy when strongly coupled modes are clearly not repeated, but still cause a highly skewed modal coordinate system.

Applications of modal decomposition to computing the matrix exponential, transfer function matrices, and multi-input/multi-output frequency responses were discussed as one of the more attractive methods for each. Modal decomposition is the most efficient method for computing the frequency response for moderate-to-large system sizes occurring in many practical problems. Applications to the interpretation of control logic synthesis in linear-quadratic-Gaussian problems and to the computation of time responses and power spectral densities are discussed in Ref. 20.

Program Availability

The PVA algorithm has been available since 1982 as a primative in the commercial package MATRIX$_x$, by Integrated Systems, Inc., of Santa Clara, California. The software for PVA is part of the Stanford TF (transfer function) package, available from the Office of Technology Licensing, Stanford University, Palo Alto, California.

Acknowledgments

This research was funded by NASA under Grant NGL-05-020-007. We gratefully acknowledge the detailed and helpful clarifying comments of the Institute of Electrical and Electronics Engineers (IEEE) reviewers in 1981 and Alan Laub, the then Associate Editor of the *IEEE Transactions on Automatic Control,* where the paper was approved but never revised and published. We similarly acknowledge those that contributed during the AIAA review process. Special thanks are expressed to Bo Kagstrom for his thorough and constructive comments on this manuscript, including helpful suggestions and comments on related recent research.

References

[1]Kagstrom, B., and Ruhe, A., "An Algorithm for Numerical Computation of the Jordan Normal Form of a Complex Matrix," *ACM Transactions on Mathematical Software,* Vol. 6, No. 3, 1980, pp. 398-419.

[2]Turnbull, H. W., and Aitken, A. C., *An Introduction to the Theory of Canonical Matrices,* Dover, New York, 1961.

[3]Gantmacher, F. R., *The Theory of Matrices,* (trans.), Chelsea, New York, 1959.

[4]Lancaster, P., *Theory of Matrices,* Academic, New York, 1969.

[5]Samuelson, H., *An Introduction of Linear Algebra,* Wiley, New York, 1974.

[6]Halmos, P. R., *Finite Dimensional Vector Spaces,* Springer-Verlag, New York, 1970.

[7]Strang, G., *Linear Algebra and Its Applications,* Academic, New York, 1976.

[8]Varah, J. M., "Computing Invariant Subspaces of a General Matrix when the Eigensystem is Poorly Conditioned," *Mathematical Computations,* Vol. 24, 1970, pp. 137-149.

[9]Ruhe, A., "Properties of a Matrix with a Very Ill-Conditioned Eigenproblem," *Numerical Mathematics,* Vol. 15, 1970, pp. 57-60.

[10]Golub, G. H., and Wilkinson, J. H., "Ill-Conditioned Eigensystems and the Computation of the Jordan Canonical Form," *SIAM Review,* Vol. 8, No. 4, 1976, pp. 578-619.

[11]Wilkinson, J. H., and Reinsch, C., (eds.), *Handbook for Automatic Computation: Linear Algebra,* Vol. II, Springer-Verlag, New York, 1971; also, Peters, G., and Wilkinson, J. H., "Eigenvectors of Real and Complex Matrices by LR and QR Triangularizations," *Numer. Math.,* Vol. 16, 1970, pp. 181-204.

[12]Dongarra, J. J., Moler, C., and Wilkinson, J., "Improving the Accuracy of Computed Matrix Eigenvalues," *SIAM Journal of Numerical Analysis,* Vol. 20, 1983, pp. 46-58.

[13]Smith, B. T., et al., "Matrix Eigensystem Routines—EISPACK Guide," *Lecture Notes in Computer Science,* Vol. 6, 2nd ed., Springer-Verlag, 1976.

[14]Wilkinson, J. H., *The Algebraic Eigenvalue Problem,* Clarendon, Oxford, UK, 1965.

[15]Kahan, N. M., "Conserving Confluence Curbs Ill-Condition," Computer Science Tech. Rept. 6, Univ. of California, Berkeley, CA, 1972.

[16]Kato, T., *Perturbation Theory for Linear Operators,* Springer-Verlag, Berlin, 1966.

[17]Ruhe, A., "Perturbation Bounds for Means of Eigenvalue and Invariant Subspaces," *BIT,* Vol. 10, 1970, pp. 343-354.

[18]Ruhe, A., "An Algorithm for Numerical Determination of the Structure of a General Matrix," *BIT,* Vol. 10, 1970, pp. 196-216.

[19]Poston, T., and Stewart, I., *Catastrophe Theory and its Applications,* Pitmann, San Francisco, 1978.

[20]Walker, R. A., "Computing the Jordan Form for Control of Dynamic Systems," Ph.D. Dissertation, Dept. of Aeronautics and Astronautics, Stanford Univ., Stanford, CA, SUDAAR 528, March 1981.

[21]Moler, C. B., and Van Loan, C., "Nineteen Dubious Ways to Compute the Exponential of a Matrix," *SIAM Review,* Vol. 20, No. 4, 1978, pp. 801-836.

[22]Kagstrom, B., "Numerical Computation of Matrix Functions," Dept. of Information Processing, Umea Univ., Umea, Sweden, Rept. UMINF-58.77, 1977.

[23]Kagstrom, B., "Bounds and Perturbation Bounds for the Matrix Exponential," *BIT,* Vol. 17, 1977, pp. 39-57.

[24]Varah, J. M., "The Calculation of the Eigenvectors of a General Complex Matrix by Inverse Iteration," *Mathematical Computations,* Vol. 22, 1968, pp. 785-791.

[25]Kublanovskaya, V. N., "On a Method of Solving the Complete Eigenvalue Problem for a Degenerate Matrix," *USSR Computational Mathematics and Mathematical Physics,* Vol. 6, No. 4, 1968, pp. 1-14.

[26]Hager, D. F., and Gordon, R. G., "The Reduction of a General Complex Matrix to a Condensed Form by Bounded Single Element Transformations," *Journal of Computational Physics,* Vol. 32, 1979, pp. 377-402.

[27]Stewart, G. W., "Algorithm 506, HQR3 and EXCHNG: Fortran Subroutines for Calculating and Ordering the Eigenvalues of a Real Upper Hessenberg Matrix [F2]," *ACM Transactions on Mathematical Software,* Vol. 2, No. 3, 1976, pp. 218-289.

[28]Flamm, D. S., and Walker, R. A., "Remark on Algorithm 506, HQR3 and EXCHNG: Fortran Subroutines for Calculating and Ordering the Eigenvalues of a Real Upper Hessenberg Matrix [F2]," *ACM Transactions on Mathematical Software,* Vol. 8, No. 2, 1982, pp. 219-220.

[29]Van Dooren, P., "A Generalized Eigenvalue Approach for Solving Riccati Equations," *SIAM Journal of Statistics and Scientific Computations,* Vol. 2, 1981, pp. 121-135.

[30]Kagstrom, B., and Ruhe, A., "ALGORITHM 650, JNF, An Algorithm for Numerical Computation of the Jordan Normal Form of a Complex Matrix [F2]," *ACM Transactions on Mathematical Software,* Vol. 6, No. 3, 1980, pp. 437-443.

[31]Feingold, D. G., and Varga, R. S., "Block Diagonally Dominant Matrices and Generalizations of the Gerschgorin Circle Theorem," *Pacific Journal of Mathematics,* Vol. 12, 1962, pp. 1241-1250.

[32]Stewart, G. W., "Error and Perturbation Bounds for Subspaces Associated with Certain Eigenvalue Problems," *SIAM Review,* Vol. 15, No. 4, 1973, pp. 727-764.

[33]Kagstrom, B., "How to Compute the Jordan Normal Form—The Choice Between Similarity Methods and Methods Using the Chain Relations," Inst. of Information Processing, Univ. of Umea, Umea, Sweden, Rept. UMINF 91.81, 1981.

[34]Wilkinson, J., "Sensitivity of Eigenvalues," *Utilitas Mathematica,* Vol. 25, 1984, pp. 5-76.

[35]Wilkinson, J., "Sensitivity of Eigenvalues II," *Utilitas Mathematica,* Vol. 30, 1986, pp. 243-286.

[36]Demmel, J., "Computing Stable Eigendecompositions of Matrices," *Linear Algebra Application,* Vol. 79, 1986, pp. 163-193.

[37]Demmel, J., and Kagstrom, B., "Stable Eigendecompositions of Matrix Pencils," *Linear Algebra Application,* Vol. 88/89, 1987, pp. 139-186.

[38]Kagstrom, B., and Westin, L., "Generalized Schur Methods with Condition Estimators for Solving the Generalized Sylvester Equation," *IEEE Transactions on Automatic Control,* Vol. AC-34, No. 10, 1989, pp. 745-751.

[39]Kagstrom, B., and Poromaa, P., "Distributed and Shared Memory Block Algorithms for the Triangular Sylvester Equation with Sep-inverse Estimators," Fourth SIAM Conf. on Parallel Processing for Scientific Computing, Chicago, Dec. 1989.

[40]Bai, Z., Demmel, J., and McKenney, A., "On the Conditioning of the Nonsymmetric Eigenproblem: Theory and Software," Computer Science Dept., Courant Inst., New York, LAPACK Working Note 13, CS-89-86, 1989.

[41]Byers, R., "A LINPACK-Style Condition Estimator for the Equation $AX - XB = C$," *IEEE Transactions on Automatic Control,* Vol. AC-29, No. 10, 1984, pp. 926-928.

[42]Moler, C. B., "Three Research Problems in Numerical Linear

Algebra," *Proceedings of Symposia in Applied Mathematics,* Vol. 22, 1978.

[43]Cline, A. K., Moler, C. B., Stewart, G. W., and Wilkinson, J. H., "An Estimate for the Condition Number of a Matrix," *SIAM Journal of Numerical Analysis,* Vol. 16, 1979, pp. 368–375.

[44]Ward, R. C., "Numerical Computation of the Matrix Exponential with Accuracy Estimate," *SIAM Journal of Numerical Analysis,* Vol. 14, 1977, pp. 600–610.

[45]Parlett, B. N., "A Recurrence Among the Elements of Functions of Triangular Matrices," *Linear Algebra and its Applications,* Vol. 14, 1976, pp. 117–121.

[46]Bavely, C. A., and Stewart, G. W., "An Algorithm for Computing Subspaces by Diagonalization," *SIAM Journal of Numerical Analysis,* Vol. 16, 1979, pp. 359–367.

[47]Van Loan, C., "The Sensitivity of the Matrix Exponential," *SIAM Journal of Numerical Analysis,* Vol. 14, 1977, pp. 971–981.

[48]Emani-Naeini, A., and Van Dooren, P., "Computation of Zeros of Linear Multivariable Systems," *Automatica,* Vol. 18, No. 4, 1982, pp. 415–430.

[49]Hall, W. E., Mohr, R. H., and Walker, R. A., "Analytical Evaluation of Tilting Proprotor Wind Tunnel Test Requirements," Systems Control, Inc. (VT.), Palo Alto, CA, Final Rept. under NASA Contract NAS2-8799, Jan. 1976.

[50]Faddeeva, V. N., *Computation Methods of Linear Algebra,* Dover, New York, 1959.

[51]Wilkinson, J. H., "The Differential System Bx and Ax and the Generalized Eigenvalue Problem $Au = \lambda Bu$," Div. of Numerical Analysis and Computation, National Physical Lab., Teddington, UK, NPL Rept. NAC 73, Jan. 1977.

[52]Beelen, T., and Van Dooren, P., "An Improved Algorithm for the Computation of Kronecker's Canonical Form of a Singular Pencil," *Linear Algebra Application,* Vol. 105, 1988, pp. 9–65.

[53]Doyle, J. C., and Stein, G., "Multivariable Feedback Design: Concepts for a Classical/Modern Synthesis," *IEEE Transactions on Automatic Control,* Vol. AC-26, Feb. 1981, pp. 4–16.

[54]Safanov, M. G., Laub, A. J., and Hartmann, G. L., "Feedback Properties of Multivariable Systems: The Role and Use of the Return Difference Matrix," *IEEE Transactions on Automatic Control,* Vol. AC-26, Feb. 1981, pp. 47–65.

[55]Laub, A. J., "Efficient Multivariable Frequency Response Computations," *IEEE Transactions on Automatic Control,* Vol. AC-26, No. 2, 1981, pp. 407–408.