NAG Toolbox for MATLAB

Chapter Introduction

F01 – Matrix Factorizations

Contents

1 Scope of the Chapter

This chapter provides facilities for three types of problem:

(i) Matrix Inversion

(ii) Matrix Factorizations

(iii) Matrix Arithmetic and Manipulation

These problems are discussed separately in Section 2.1, Section 2.2 and Section 2.3.

2 Background to the Problems

2.1 Matrix Inversion

(i) Non-singular square matrices of order n.

If A, a square matrix of order n, is nonsingular (has rank n), then its inverse X exists and satisfies the equations $AX = XA = I$ (the identity or unit matrix).

It is worth noting that if $AX - I = R$, so that R is the 'residual' matrix, then a bound on the relative error is given by $\|R\|$, i.e.,

$$\frac{\|X - A^{-1}\|}{\|A^{-1}\|} \leq \|R\|.$$

(ii) General real rectangular matrices.

A real matrix A has no inverse if it is square (n by n) and singular (has rank $< n$), or if it is of shape (m by n) with $m \neq n$, but there is a **Generalized** or **Pseudo Inverse** Z which satisfies the equations

$$AZA = A, \qquad ZAZ = Z, \qquad (AZ)^{\mathrm{T}} = AZ, \qquad (ZA)^{\mathrm{T}} = ZA$$

(which of course are also satisfied by the inverse X of A if A is square and nonsingular).

(a) if $m \geq n$ and $\mathrm{rank}(A) = n$ then A can be factorized using a ***QR*** **factorization**, given by

$$A = Q\begin{pmatrix} R \\ 0 \end{pmatrix},$$

where Q is an m by m orthogonal matrix and R is an n by n, nonsingular, upper triangular matrix. The pseudo-inverse of A is then given by

$$Z = R^{-1}\tilde{Q}^{\mathrm{T}},$$

where $\tilde{Q}$ consists of the first n columns of Q.

(b) if $m \leq n$ and $\mathrm{rank}(A) = m$ then A can be factorized using an ***RQ*** **factorization**, given by

$$A = (R \quad 0)P^{\mathrm{T}}$$

where P is an n by n orthogonal matrix and R is an m by m, nonsingular, upper triangular matrix. The pseudo-inverse of A is then given by

$$Z = \tilde{P}R^{-1},$$

where $\tilde{P}$ consists of the first m columns of P.

(c) if $m \geq n$ and $\mathrm{rank}(A) = r \leq n$ then A can be factorized using a QR factorization, with column interchanges, as

$$A = Q\begin{pmatrix} R \\ 0 \end{pmatrix}P^{\mathrm{T}},$$

where Q is an m by m orthogonal matrix, R is an r by n upper trapezoidal matrix and P is an n by n permutation matrix. The pseudo-inverse of A is then given by

$$Z = PR^{\mathrm{T}}\left(RR^{\mathrm{T}}\right)^{-1}\tilde{Q}^{\mathrm{T}},$$

where $\tilde{Q}$ consists of the first r columns of Q.

(d) if $\mathrm{rank}(A) = r \leq k = \min(m, n)$, then A can be factorized as the **singular value decomposition**

$$A = QDP^{\mathrm{T}},$$

where Q is an m by m orthogonal matrix, P is an n by n orthogonal matrix and D is an m by n diagonal matrix with nonnegative diagonal elements. The first k columns of Q and P are the **left-** and **right-hand singular vectors** of A respectively and the k diagonal elements of D are the **singular values** of A. D may be chosen so that

$$d_1 \geq d_2 \geq \cdots \geq d_k \geq 0$$

and in this case if $\mathrm{rank}(A) = r$ then

$$d_1 \geq d_2 \geq \cdots \geq d_r > 0, \qquad d_{r+1} = \cdots = d_k = 0.$$

If $\tilde{Q}$ and $\tilde{P}$ consist of the first r columns of Q and P respectively and Σ is an r by r diagonal matrix with diagonal elements $d_1, d_2, \ldots, d_r$ then A is given by

$$A = \tilde{Q}\Sigma\tilde{P}^{\mathrm{T}}$$

and the pseudo-inverse of A is given by

$$Z = \tilde{P}\Sigma^{-1}\tilde{Q}^{\mathrm{T}}.$$

Notice that

$$A^{\mathrm{T}}A = P\left(D^{\mathrm{T}}D\right)P^{\mathrm{T}}$$

which is the classical eigenvalue (spectral) factorization of $A^{\mathrm{T}}A$.

(e) if A is complex then the above relationships are still true if we use 'unitary' in place of 'orthogonal' and conjugate transpose in place of transpose. For example, the singular value decomposition of A is

$$A = QDP^{\mathrm{H}},$$

where Q and P are unitary, P^{H} the conjugate transpose of P and D is as in (d) above.

2.2 Matrix Factorizations

The functions in this section perform matrix factorizations which are required for the solution of systems of linear equations with various special structures. A few functions which perform associated computations are also included.

Other functions for matrix factorizations are to be found in Chapters F03, F07, F08 and F11.

This section also contains a few functions associated with eigenvalue problems (see Chapter F02). (Historical note: this section used to contain many more such functions, but they have now been superseded by functions in Chapter F08.)

2.3 Matrix Arithmetic and Manipulation

The intention of functions in this section (sub-chapters F01C and F01Z) is to cater for some of the commonly occurring operations in matrix manipulation, e.g., transposing a matrix or adding part of one matrix to another, and for conversion between different storage formats, e.g., conversion between rectangular band matrix storage and packed band matrix storage. For vector or matrix-vector or matrix-matrix operations refer to Chapter F06.

3 Recommendations on Choice and Use of Available Functions

3.1 Matrix Inversion

Note: before using any function for matrix inversion, consider carefully whether it is really needed.

Although the solution of a set of linear equations $Ax = b$ can be written as $x = A^{-1}b$, the solution should **never** be computed by first inverting A and then computing $A^{-1}b$; the functions in Chapters F04 or F07 should **always** be used to solve such sets of equations directly; they are faster in execution, and numerically more stable and accurate. Similar remarks apply to the solution of least-squares problems which again should be solved by using the functions in Chapter F04 rather than by computing a pseudo-inverse.

(a) Non-singular square matrices of order n

This chapter describes techniques for inverting a general real matrix A and matrices which are positive-definite (have all eigenvalues positive) and are either real and symmetric or complex and Hermitian. It is wasteful and uneconomical not to use the appropriate function when a matrix is known to have one of these special forms. A general function must be used when the matrix is not known to be positive-definite. In most functions the inverse is computed by solving the linear equations $Ax_i = e_i$, for $i = 1, 2, \ldots, n$, where e_i is the ith column of the identity matrix.

Functions are given for calculating the approximate inverse, that is solving the linear equations just once, and also for obtaining the accurate inverse by successive iterative corrections of this first approximation. The latter, of course, are more costly in terms of time and storage, since each correction involves the solution of n sets of linear equations and since the original A and its LU decomposition must be stored together with the first and successively corrected approximations to the inverse. In practice the storage requirements for the 'corrected' inverse functions are about double those of the 'approximate' inverse functions, though the extra computer time is not prohibitive since the same matrix and the same LU decomposition is used in every linear equation solution.

Despite the extra work of the 'corrected' inverse functions they are superior to the 'approximate' inverse functions. A correction provides a means of estimating the number of accurate figures in the inverse or the number of 'meaningful' figures relating to the degree of uncertainty in the coefficients of the matrix.

The residual matrix $R = AX - I$, where X is a computed inverse of A, conveys useful information. Firstly $\|R\|$ is a bound on the relative error in X and secondly $\|R\| < \frac{1}{2}$ guarantees the convergence of the iterative process in the 'corrected' inverse functions.

The decision trees for inversion show which functions in Chapter F04 and Chapter F07 should be used for the inversion of other special types of matrices not treated in the chapter.

(b) General real rectangular matrices

For real matrices f08ae and f01qj return QR and RQ factorizations of A respectively and f08be returns the QR factorization with column interchanges. The corresponding complex functions are f08as, f01rj and f08bs respectively. Functions are also provided to form the orthogonal matrices and transform by the orthogonal matrices following the use of the above functions. f01qg and f01rg form the RQ factorization of an upper trapezoidal matrix for the real and complex cases respectively.

f01bl uses the QR factorization as described in Section 2.1(ii)(a) and is the only function that explicitly returns a pseudo-inverse. If $m \geq n$, then the function will calculate the pseudo-inverse Z of the matrix A. If $m < n$, then the n by m matrix A^{T} should be used. The function will calculate the pseudo-inverse Z of A^{T} and the required pseudo-inverse will be Z^{T}. The function also attempts to calculate the rank, r, of the matrix given a tolerance to decide when elements can be regarded as zero. However, should this function fail due to an incorrect determination of the rank, the singular value decomposition method (described below) should be used.

f08kb and f08kp compute the singular value decomposition as described in Section 2 for real and complex matrices respectively. If A has rank $r \leq k = \min(m, n)$ then the $k - r$ smallest singular values will be negligible and the pseudo-inverse of A can be obtained as $Z = P\Sigma^{-1}Q^{\mathrm{T}}$ as described in Section 2. If the rank of A is not known in advance it can be estimated from the singular values (see Section 2.2 in the F04 Chapter Introduction). In the real case with $m \geq n$, f02wd provides details of the *QR* factorization or the singular value decomposition depending on whether or not *A* is of full rank and for some problems provides an attractive alternative to f08kb. For large sparse matrices, leading terms in the singular-value decomposition can be computed using functions from Chapter F12.

3.2 Matrix Factorizations

Each of these functions serves a special purpose required for the solution of sets of simultaneous linear equations or the eigenvalue problem. For further details you should consult Sections 3 or 4 in the F02 Chapter Introduction or Sections 3 or 4 in the F04 Chapter Introduction.

f01br and f01bs are provided for factorizing general real sparse matrices. A more recent algorithm for the same problem is available through f11me. For factorizing real symmetric positive-definite sparse matrices, see f11ja. These functions should be used only when *A* is **not** banded and when the total number of nonzero elements is less than 10% of the total number of elements. In all other cases either the band functions or the general functions should be used.

3.3 Matrix Arithmetic and Manipulation

The functions in the F01C section are designed for the general handling of *m* by *n* matrices. Emphasis has been placed on flexibility in the parameter specifications and on avoiding, where possible, the use of internally declared arrays. They are therefore suited for use with large matrices of variable row and column dimensions. functions are included for the addition and subtraction of sub-matrices of larger matrices, as well as the standard manipulations of full matrices. Those functions involving matrix multiplication may use additional-precision arithmetic for the accumulation of inner products. See also Chapter F06.

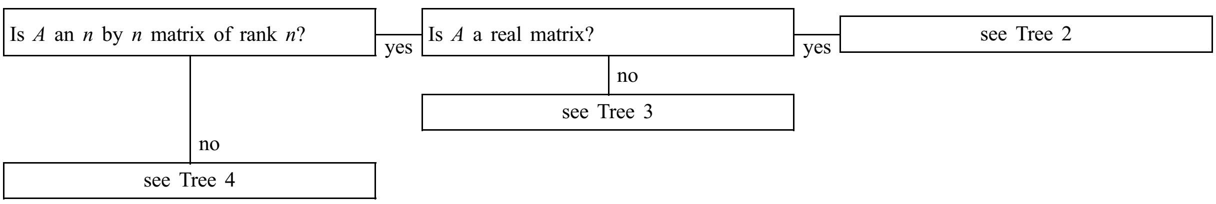
The functions in the F01Z section are designed to allow conversion between square storage and the packed storage schemes required by some of the functions in Chapters F02, F04, F06, F07 and F08.

4 Decision Trees

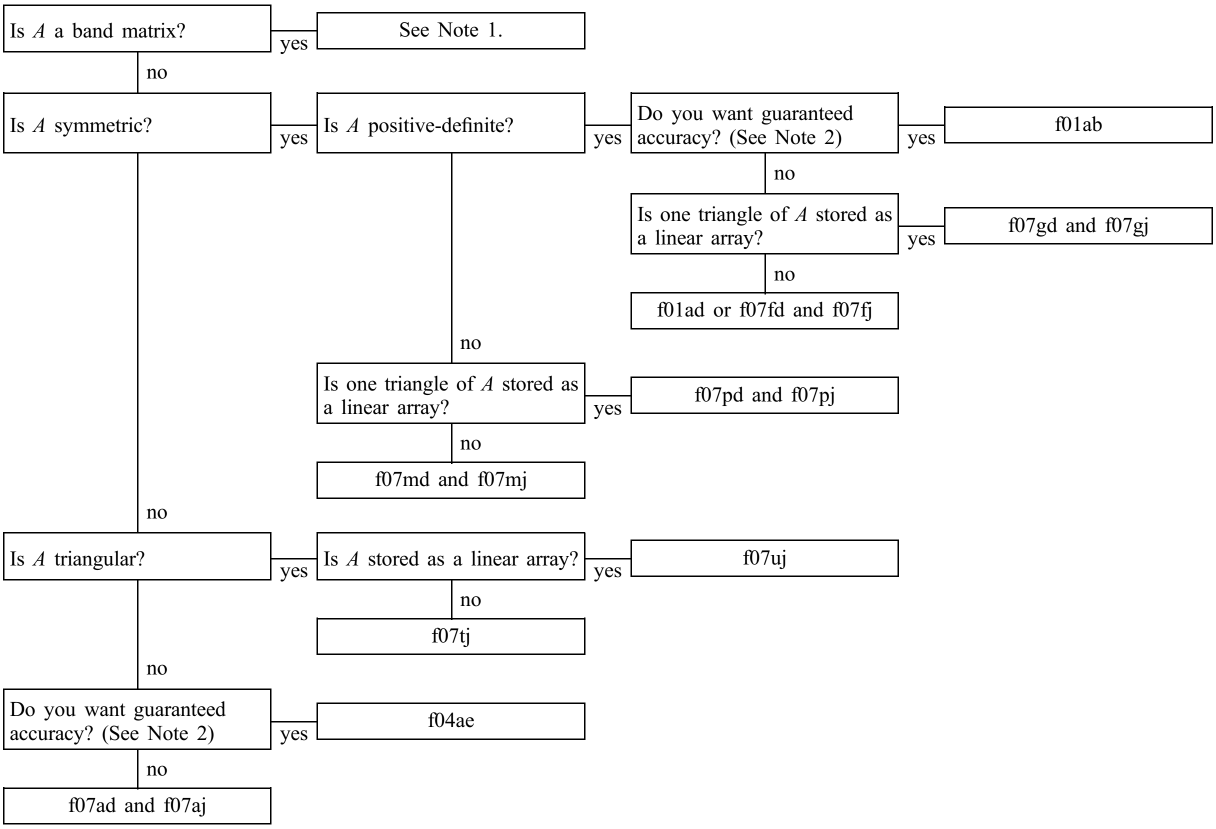
The decision trees show the functions in this chapter and in Chapter F04 that should be used for inverting matrices of various types. Functions marked with an asterisk ($*$) only perform part of the computation – see Section 3.1 for further advice.

(i) **Matrix Inversion:**

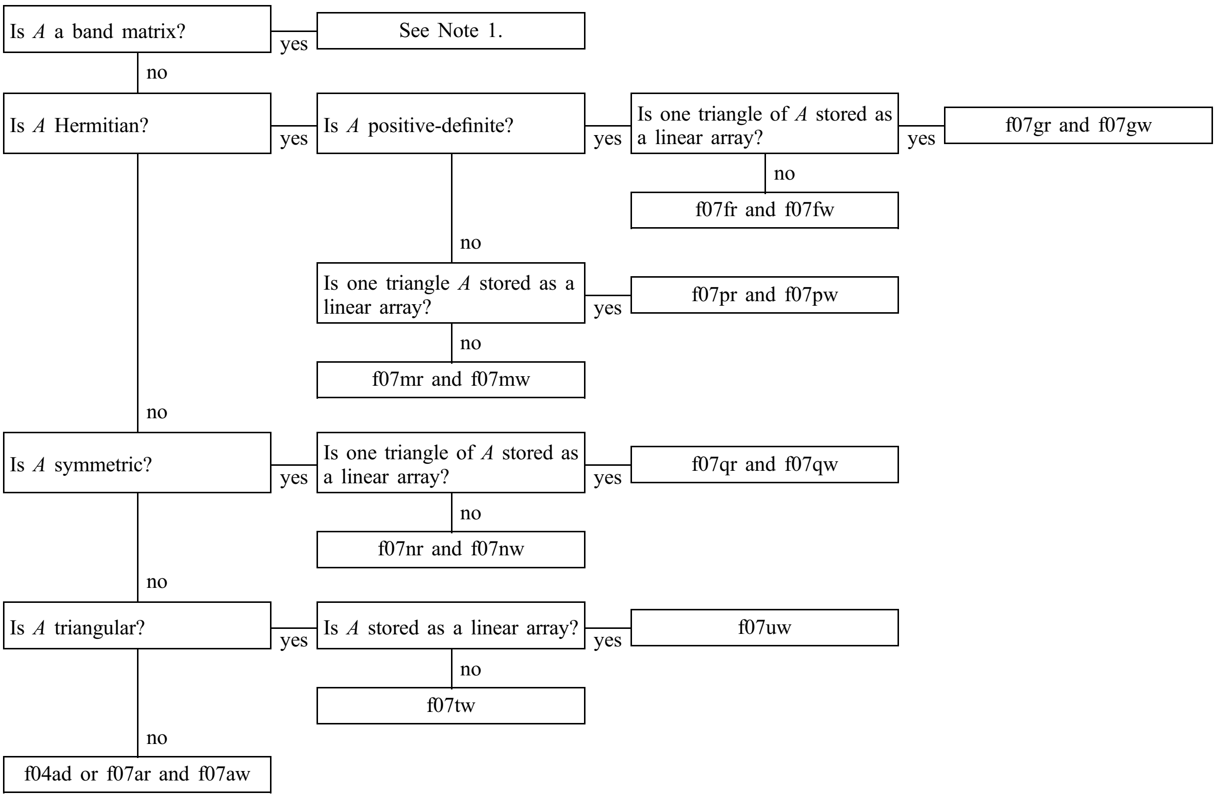
Tree 1



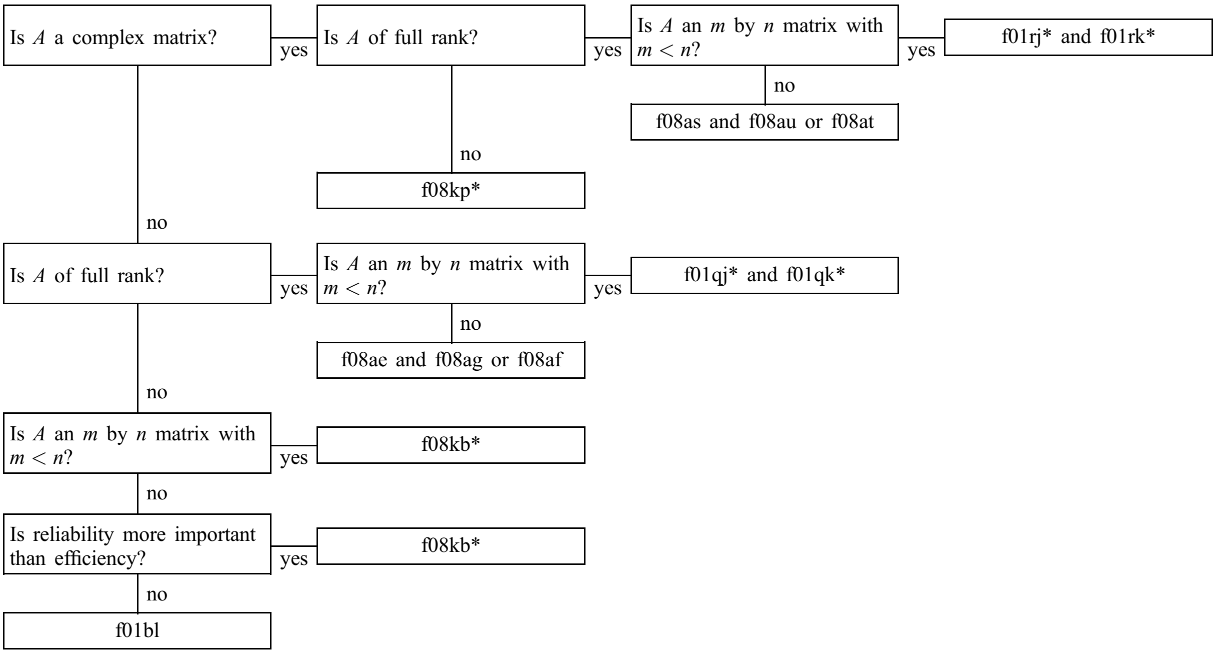
Tree 2: Inverse of a real *n* by *n* matrix of full rank



Tree 3: Inverse of a complex *n* by *n* matrix of full rank



Tree 4: Pseudo-inverses



Note 1: the inverse of a band matrix A does not in general have the same shape as A, and no functions are provided specifically for finding such an inverse. The matrix must either be treated as a full matrix, or the equations $AX = B$ must be solved, where B has been initialized to the identity matrix I. In the latter case, see the decision trees in Section 4 in the F04 Chapter Introduction.

Note 2: by 'guaranteed accuracy' we mean that the accuracy of the inverse is improved by use of the iterative refinement technique using additional precision.

(ii) **Matrix Factorizations:** See the decision trees in Section 4 in the F02 and F04 Chapter Introductions.

(iii) **Matrix Arithmetic and Manipulation:** Not appropriate.

5 Index

6 References

Golub G H and Van Loan C F 1996 *Matrix Computations* (3rd Edition) Johns Hopkins University Press, Baltimore

Wilkinson J H 1965 *The Algebraic Eigenvalue Problem* Oxford University Press, Oxford

Wilkinson J H 1977 Some recent advances in numerical linear algebra *The State of the Art in Numerical Analysis* (ed D A H Jacobs) Academic Press

Wilkinson J H and Reinsch C 1971 *Handbook for Automatic Computation II, Linear Algebra* Springer–Verlag