NAG Fortran Library 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 non-singular (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

$$AZA = A, \quad ZAZ = Z, \quad (AZ)^T = AZ, \quad (ZA)^T = ZA$$

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

(a) if $m \geq n$ and $\text{rank}(A) = n$ 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, non-singular, upper triangular matrix. The pseudo-inverse of A is then given by

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

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

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

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

where P is an n by n orthogonal matrix and R is an m by m, non-singular, 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 $\text{rank}(A) = r \leq n$ then A can be factorized using a QR factorization, with column interchanges, as

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

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

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

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

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

$$A = QDP^T,$$

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 non-negative 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 $\text{rank}(A) = r$ then

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

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

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

and the pseudo inverse of A is given by

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

Notice that

$$A^TA = P\left(D^TD\right)P^T$$

which is the classical eigenvalue (spectral) factorization of A^TA.

(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

$$A = QDP^H,$$

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

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

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 (Historical note: this section used to contain many more such routines, but they have now been superseded by routines in Chapter F08.)

2.3 Matrix Arithmetic and Manipulation

The intention of routines 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 Routines

Note: please refer to the Users' Note for your implementation to check that a routine is available.

3.1 Matrix Inversion

Note: before using any routine 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 routines 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 routines 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 routine when a matrix is known to have one of these special forms. A general routine must be used when the matrix is not known to be positive-definite. In most routines 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.

Routines 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 routines are about double those of the 'approximate' inverse routines, 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 routines they are superior to the 'approximate' inverse routines. 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 routines.

The decision trees for inversion show which routines 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 F08AEF (DGEQRF) and F01QJF return QR and RQ factorizations of A respectively and F08BEF (DGEQPF) returns the QR factorization with column interchanges. The corresponding complex routines are F08ASF (ZGEQRF), F01RJF and F08BSF (ZGEQPF) respectively. Routines are also provided to form the orthogonal matrices and transform by the orthogonal matrices following the use of the above routines. F01QGF and F01RGF form the RQ factorization of an upper trapezoidal matrix for the real and complex cases respectively.

F01BLF uses the QR factorization as described in Section 2.1(ii)(a) and is the only routine that explicitly returns a pseudo inverse. If $m \geq n$, then the routine 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 routine will calculate the pseudo inverse Z of A^T and the required pseudo inverse will be Z^T. The routine 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 routine fail due to an incorrect determination of the rank, the singular value decomposition method (described below) should be used.

F08KBF (DGESVD) and F08KPF (ZGESVD) 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 $Z = P\Sigma^{-1}Q^T$ 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 $m \geq n$, F02WDF 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 F08KBF (DGESVD).

3.2 Matrix Factorizations

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

F01BRF and F01BSF are provided for factorizing general real sparse matrices. A more recent algorithm for the same problem is available through F11MEF. For factorizing real symmetric positive-definite sparse matrices, see F11JAF. These routines should be used only when A is **not** banded and when the total number of non-zero elements is less than 10% of the total number of elements. In all other cases either the band routines or the general routines should be used.

3.3 Matrix Arithmetic and Manipulation

The routines 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. routines are included for the addition and subtraction of sub-matrices of larger matrices, as well as the standard manipulations of full matrices. Those routines involving matrix multiplication may use additional-precision arithmetic for the accumulation of inner products. See also Chapter F06.

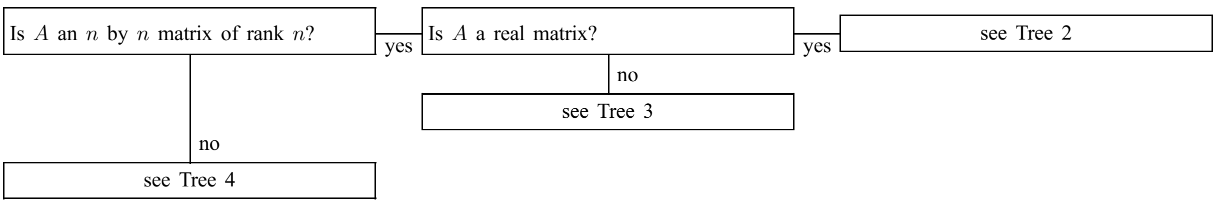
The routines in the F01Z section are designed to allow conversion between square storage and the packed storage schemes required by some of the routines in Chapters F02, F04, F06, F07 and F08.

4 Decision Trees

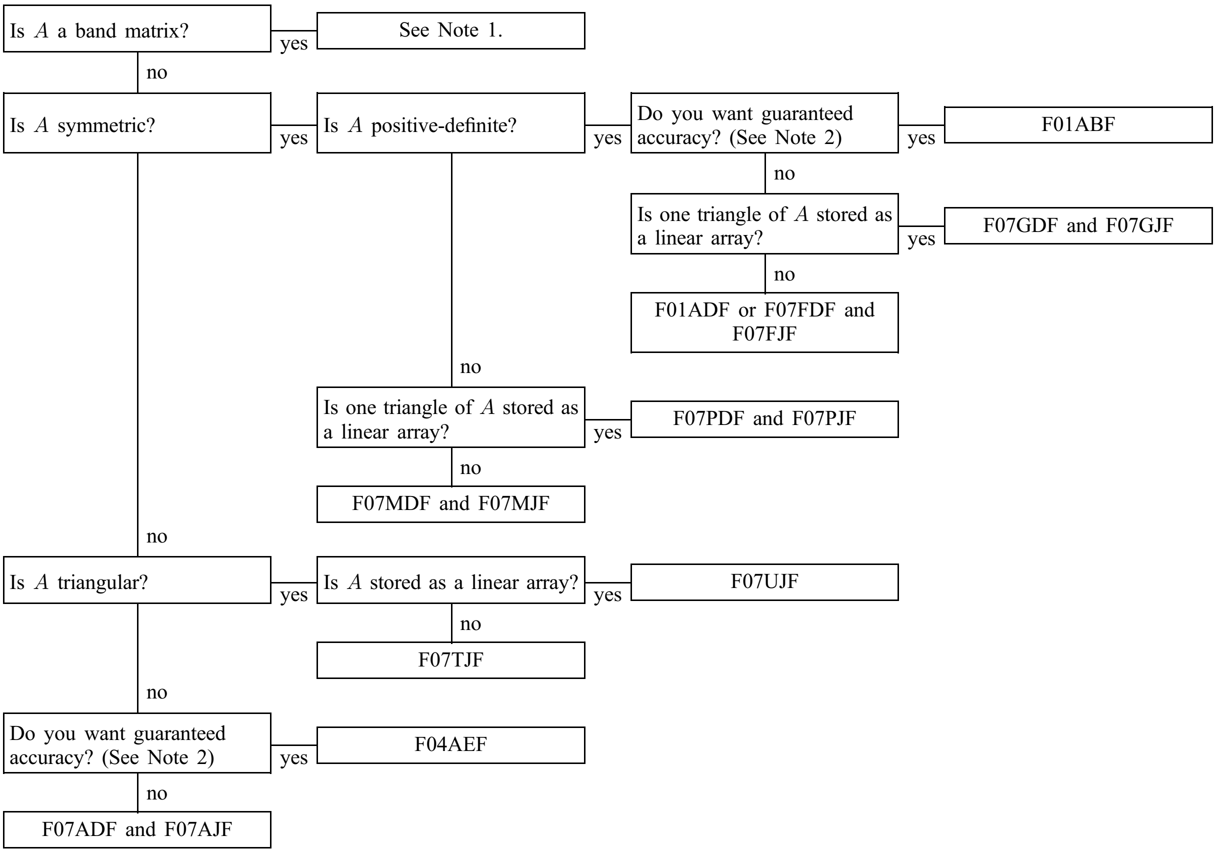
The decision trees show the routines in this chapter and in Chapter F04 that should be used for inverting matrices of various types. Routines marked with an asterisk ($*$) only perform part of the computation – see Section 3.1 for further advice.

(i) **Matrix Inversion:**

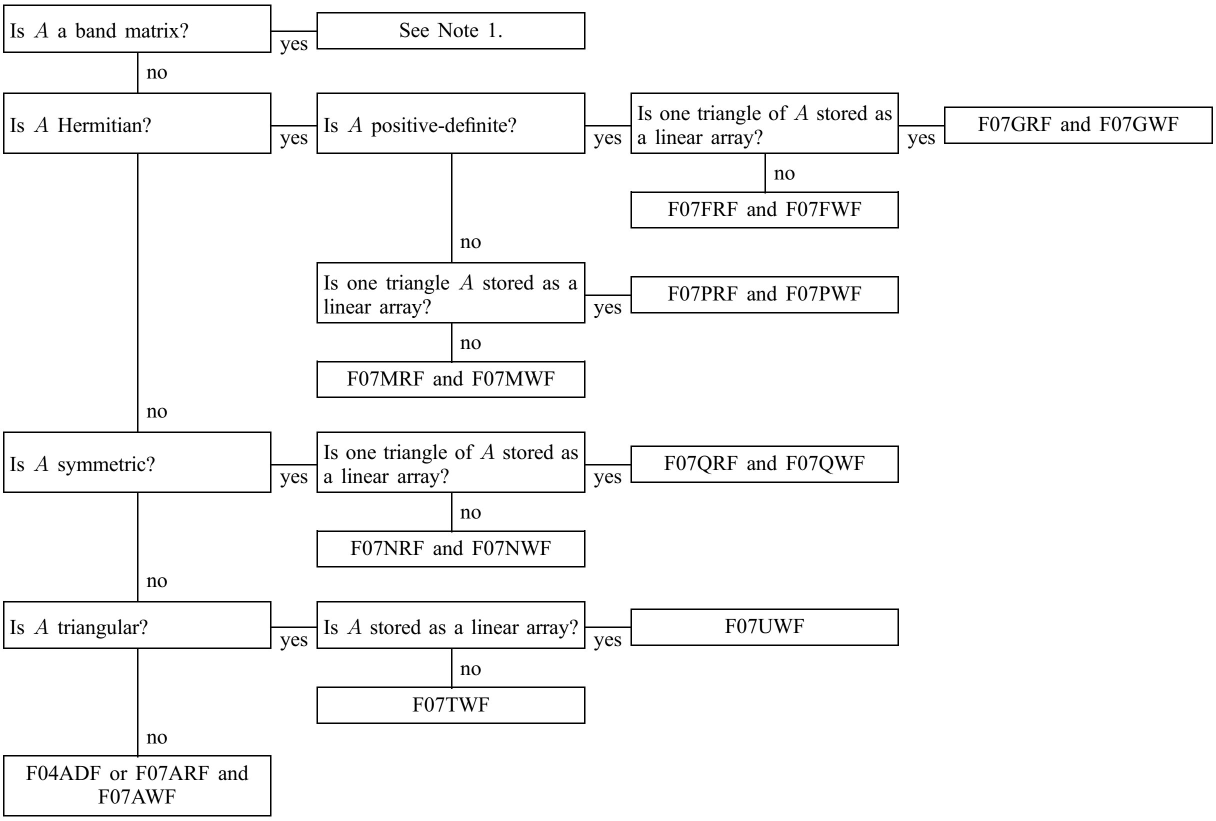
Tree 1



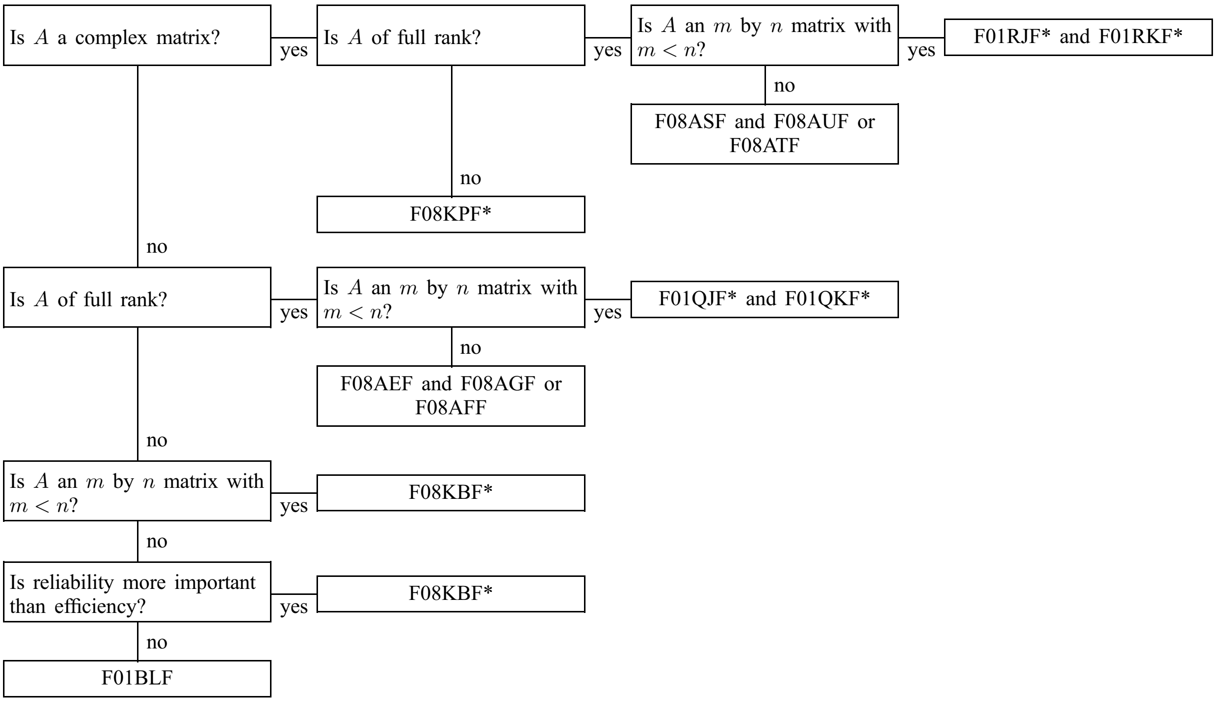
Tree 2: Inverse of a real *n* by *n* matrix of full rank



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 routines 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 Routines Withdrawn or Scheduled for Withdrawal

Withdrawn Routine	Mark of Withdrawal	Replacement Routine(s)
F01AAF	17	F07ADF (DGETRF) and F07AJF (DGETRI)
F01ACF	16	F01ABF
F01AEF	18	F07FDF (DPOTRF), F08SEF (DSYGST) and F06EGF (DSWAP)
F01AFF	18	F06YJF (DTRSM) and F06EGF (DSWAP)
F01AGF	18	F08FEF (DSYTRD)
F01AHF	18	F08FGF (DORMTR)
F01AJF	18	F08FEF (DSYTRD) and F08FFF (DORGTR)
F01AKF	18	F08NEF (DGEHRD)
F01ALF	18	F08NGF (DORMHR)
F01AMF	18	F08NSF (ZGEHRD)
F01ANF	18	F08NTF (ZUNGHR)
F01APF	18	F08NFF (DORGHR) and F06QFF
F01ATF	18	F08NHF (DGEBAL)
F01AUF	18	F08NJF (DGEBAK)
F01AVF	18	F08NVF (ZGEBAL)
F01AWF	18	F08NWF (ZGEBAK)
F01AXF	18	F08BEF (DGEQPF) and F06EFF (DCOPY)
F01AYF	18	F08GEF (DSPTRD)
F01AZF	18	F08GGF (DOPMTR)
F01BCF	18	F08FSF (ZHETRD) and F08FTF (ZUNGTR)
F01BDF	18	F07FDF (DPOTRF), F08SEF (DSYGST) and F06EGF (DSWAP)
F01BEF	18	F06YFF (DTRMM)
F01BFF	8	F07GDF (DPPTRF) or F07PDF (DSPTRF)
F01BHF	9	F08KBF (DGESVD)
F01BJF	9	F08HEF (DSBTRD)
F01BKF	9	F02WDF
F01BMF	9	F07BDF (DGBTRF)
F01BNF	17	F07FRF (ZPOTRF)
F01BPF	17	F07FRF (ZPOTRF) and F07FWF (ZPOTRI)

F01BQF	16	F07GDF (DPPTRF) and F07PDF (DSPTRF)
F01BTF	18	F07ADF (DGETRF)
F01BWF	18	F08HEF (DSBTRD)
F01BXF	17	F07FDF (DPOTRF)
F01CAF	14	F06QHF
F01CBF	14	F06QHF
F01CCF	7	F06QFF
F01CDF	15	F01CTF
F01CEF	15	F01CTF
F01CFF	14	F06QFF
F01CGF	15	F01CTF
F01CHF	15	F01CTF
F01CJF	8	F01CRF
F01CLF	16	F06YAF (DGEMM)
F01CMF	14	F06QFF
F01CNF	13	F06EFF (DCOPY)
F01CPF	13	F06EFF (DCOPY)
F01CQF	13	F06FBF
F01CSF	13	F06PEF (DSPMV)
F01DAF	13	F06EAF (DDOT)
F01DBF	13	X03AAF
F01DCF	13	F06GAF (ZDOTU)
F01DDF	13	X03ABF
F01DEF	14	F06EAF (DDOT)
F01LBF	18	F07BDF (DGBTRF)
F01LZF	15	F08KEF (DGEBRD), F08KFF (DORGBR) or F08KGF (DORMBR)
F01MAF	19	F11JAF
F01NAF	17	F07BRF (ZGBTRF)
F01QAF	15	F08AEF (DGEQRF)
F01QBF	15	F01QJF
F01QCF	18	F08AEF (DGEQRF)
F01QDF	18	F08AGF (DORMQR)
F01QEF	18	F08AFF (DORGQR)
F01QFF	18	F08BEF (DGEQPF)
F01RCF	18	F08ASF (ZGEQRF)
F01RDF	18	F08AUF (ZUNMQR)
F01REF	18	F08ATF (ZUNGQR)
F01RFF	18	F08BSF (ZGEQPF)

7 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