Chapter f – Linear Algebra

1. Scope of the Chapter

This chapter is concerned with:

(i) Matrix factorizations and transformations
(ii) Solving matrix eigenvalue problems
(iii) Finding determinants
(iv) Solving systems of linear equations

Functions in this chapter can also be used in the solution of linear least-squares problems and for matrix inversion.

2. Background

2.1. Matrix Factorizations

Functions are provided to compute the

- LU factorization
- Cholesky factorization
- QR factorization
- Singular value decomposition (SVD)

of various types of matrix.

2.1.1. *LU* and Cholesky Factorizations

The LU factorization of an n by n general square matrix A is given by

$$A = PLU,$$

where L and U are n by n lower and upper triangular matrices respectively and P is an n by n permutation matrix. P is chosen to ensure the numerical stability of the factorization process.

nag_real_lu (f03afc) and nag_complex_lu (f03ahc) compute the LU factorization of real and complex non-singular matrices respectively.

When A is a real symmetric positive-definite matrix we can choose $P = I$ and $L = U^T$ giving

$$A = LL^T \quad \text{(or equivalently } A = U^TU\text{)},$$

which is the Cholesky factorization of A. In the case of a complex Hermitian positive-definite matrix, the Cholesky factorization is

$$A = LL^H \quad \text{(or equivalently } A = U^HU\text{)},$$

where L^H denotes the conjugate transpose of L.

nag_real_cholesky (f03aec) and nag_complex_cholesky (f01bnc) compute the Cholesky factorization of real symmetric and complex Hermitian positive-definite matrices respectively.

A variant of the Cholesky factorization, given by

$$A = LDL^T$$

where L is unit lower triangular and D is diagonal, is computed by nag_real_cholesky_skyline (f01mcc) in the case where A is a real variable band (skyline) matrix.

One of the main uses of LU and Cholesky factorizations is in the solution of systems of linear equations (see Section 2.4 below).

2.1.2. QR factorizations

The QR factorization of an m by n $(m \geq n)$ general matrix A is given by

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

where Q is an m by n orthogonal matrix (or unitary matrix in the complex case) and R is an n by n upper triangular matrix.

nag_real_qr (f01qcc) and nag_complex_qr (f01rcc) compute the QR factorization of real and complex matrices respectively.

Note: in the real case

$$A^TA = R^TR$$

and in the complex case

$$A^HA = R^HR,$$

so that R is the Cholesky factor of the matrix A^TA (or A^HA) which occurs in the normal equations.

The QR factorization can be used in the solution of linear least-squares problems (see Section 2.5 below).

2.1.3. The singular value decomposition

The singular value decomposition (SVD) of a real m by n matrix A is given by

$$A = QDP^T,$$

where

$$D = \begin{pmatrix} S \\ 0 \end{pmatrix},\ m > n; \quad D = S,\ m = n; \quad D = (S\ \ 0),\ m < n,$$

Q is an m by m orthogonal matrix, P is an n by n orthogonal matrix and S is a diagonal matrix of order $\min(m,n)$ with non-negative diagonal elements. S can be chosen so that

$$s_1 \geq s_2 \geq \ldots \geq s_{\min(m,n)} \geq 0,$$

where s_i is the ith diagonal element of S. In the complex case the SVD is given by

$$A = QDP^H$$

where Q and P are both unitary, but D is still real.

The first $\min(m,n)$ columns of Q and P are the left- and right-hand singular vectors of A respectively and the diagonal elements of S are the singular values of A.

nag_real_svd (f02wec) and nag_complex_svd (f02xec) compute the singular value decomposition of real and complex matrices respectively.

Note: in the real case

$$A^TA = P(D^TD)P^T$$

and in the complex case

$$A^HA = P(D^HD)P^H,$$

so that s_i^2 is an eigenvalue of the normal matrix A^TA (or A^HA) and the ith column of P is the corresponding eigenvector. Note also that

$$\|A\|_2 = s_1.$$

The singular value decomposition provides perhaps the most reliable means of estimating the rank of a matrix. If $\operatorname{rank}(A) = k < \min(m, n)$ then in exact arithmetic

$$s_1 \geq s_2 \geq \ldots \geq s_k > 0, \quad s_{k+1} = \ldots = s_{\min(m,n)} = 0.$$

Numerically, if $\operatorname{rank}(A) = k$, or if $\operatorname{rank}(A)$ is close to k, then $s_{k+1}, \ldots, s_{\min(m,n)}$ will be small relative to s_1. See Section 5.5.8 of Golub and Van Loan (1989) for further information.

The singular value decomposition can be used in the solution of linear least-squares problems, and is particularly appropriate when the matrix of coefficients (observations) is thought to be nearly rank deficient (see Section 2.5 below).

2.2. Eigenvalue Problems

Functions are provided in this chapter for solving the standard eigenvalue problem

$$Ax = \lambda x,$$

where A is an n by n matrix, λ is an eigenvalue and

$$x \neq 0$$

is an eigenvector, and the generalized eigenvalue problem

$$Ax = \lambda Bx,$$

where B is also an n by n matrix.

2.2.1. Standard symmetric or Hermitian eigenvalue problems

nag_real_symm_eigenvalues (f02aac) and nag_real_symm_eigensystem (f02abc) may be used to compute either all the eigenvalues, or all the eigenvalues and eigenvectors, respectively of a real symmetric matrix. The functions first reduce A to a symmetric tridiagonal matrix by Householder transformations and then apply the QL algorithm (a variant of the QR algorithm) to the tridiagonal matrix. The eigenvectors produced by nag_real_symm_eigensystem (f02abc) will be orthogonal to working accuracy. nag_hermitian_eigenvalues (f02awc) and nag_hermitian_eigensystem (f02axc) perform similar computations on complex Hermitian matrices.

2.2.2. Standard unsymmetric eigenvalue problems

Four functions are provided. nag_real_eigensystem_sel (f02ecc) and nag_complex_eigensystem_sel (f02gcc) can be used to compute some or all the eigenvalues and eigenvectors of either a real or a complex general unsymmetric matrix respectively. nag_real_eigenvalues (f02afc) and nag_real_eigensystem (f02agc) may be used to compute either all the eigenvalues, or all the eigenvalues and eigenvectors, respectively of a real general unsymmetric matrix. The eigenvalues and eigenvectors of an unsymmetric matrix A is computed by first reducing it to an upper Hessenberg matrix by Householder transformations and then applying the QR algorithm (with double shifts) to the Hessenberg matrix. Even though, in exact arithmetic, A may not have a full set of n linearly independent eigenvectors, nag_real_eigensystem_sel (f02ecc) nag_complex_eigensystem_sel (f02gcc) and nag_real_eigensystem (f02agc) will produce such a set, but they may be nearly linearly dependent.

2.2.3. Generalized symmetric-definite eigenvalue problems

nag_real_symm_general_eigenvalues (f02adc) and nag_real_symm_general_eigensystem (f02aec) may be used to solve the real generalized symmetric eigenvalue problem $Ax = \lambda Bx$, where A is symmetric and B is symmetric positive-definite; nag_real_symm_general_eigenvalues (f02adc) produces all the eigenvalues and nag_real_symm_general_eigensystem (f02aec) produces all the eigenvalues and eigenvectors. The functions first calculate the Cholesky factorization of B given by

$$B = LL^T,$$

where L is lower triangular, and then transform the generalized problem to the standard problem

$$Cy = \lambda y, \quad \text{where } C = L^{-1}AL^{-T} \text{ and } y = L^T x.$$

Since this method implicitly involves the inversion of B, it is strongly recommended that B be well-conditioned with respect to inversion; i.e., B should not have small eigenvalues.

The generalized symmetric eigenvalue problem

$$ABx = \lambda x$$

may be similarly solved via a Cholesky factorization, by solving the standard eigenvalue problem

$$Cy = \lambda y, \quad C = L^T AL, \quad x = L^{-T} y.$$

2.2.4. Generalized unsymmetric eigenvalue problems

nag_real_general_eigensystem (f02bjc) may be used to solve a real unsymmetric eigenproblem $Ax = \lambda Bx$, where both A and B are unsymmetric square matrices. As in unsymmetric standard eigenproblems, the eigenvalues and eigenvectors may be complex, in which case they occur in complex conjugate pairs. The function first simultaneously reduces A to an upper Hessenberg matrix and B to an upper triangular matrix, by orthogonal transformations, and then applies the QZ algorithm (with single or double shifts) to compute the eigenvalues. The eigenvectors (if wanted) are found by back-substitution and back-transformation.

2.3. Determinants

The determinant of an n by n matrix A is readily computed from its LU factorization (see Section 2.1.1) as

$$\det(A) = \mathrm{sign}(P)(l_{11}l_{22}\ldots l_{nn})(u_{11}u_{22}\ldots u_{nn}),$$

where sign(P) is the associated sign of the permutation matrix P. In the particular case of the Cholesky factorization this becomes

$$\det(A) = l_{11}^2 l_{22}^2 \ldots l_{nn}^2.$$

To avoid overflow and underflow in the computation of the determinant the functions in this chapter find $\det(A)$ in the form

$$\det(A) = d_1.2^{d_2},$$

where d_2 is an integer and

$$\tfrac{1}{16} \le |d_1| < 1.$$

nag_real_lu (f03afc) and nag_complex_lu (f03ahc) compute the determinant of real and complex general matrices respectively, and nag_real_cholesky (f03aec) computes the determinant of a real symmetric positive-definite matrix.

2.4. Simultaneous Linear Equations

Functions are provided to solve systems of linear equations of the form

$$Ax = b,$$

where A is an n by n non-singular matrix, b is an n element (single right-hand side) vector and x is the n element solution vector, as well as functions to solve systems with multiple right-hand sides

$$AX = B,$$

where B is an n by r matrix and X is the n by r solution matrix. In some cases the equations can be solved by calling a single function, and in other cases it is necessary first to call a function to factorize the matrix A (see Section 2.1.1) before calling the function to solve the linear equations.

nag_real_lin_eqn (f04arc) solves a system of equations with a real general matrix A and a single right-hand side, while the combination nag_real_lu (f03afc) and nag_real_lu_solve_mult_rhs (f04ajc) can be used for multiple right-hand sides. In the case of a complex general matrix with multiple right-hand sides either nag_complex_lin_eqn_mult_rhs (f04adc), or the combination of nag_complex_lu (f03ahc) and nag_complex_lu_solve_mult_rhs (f04akc), may be used.

The combination nag_real_cholesky (f03aec) and nag_real_cholesky_solve_mult_rhs (f04agc) may be used with a real symmetric positive-definite matrix A and multiple right-hand sides, and the combination nag_complex_cholesky (f01bnc) and nag_hermitian_lin_eqn_mult_rhs (f04awc) when A is complex Hermitian positive-definite.

Naturally the functions for multiple right-hand sides may be used with single right-hand sides by setting $r = 1$.

Note that to solve, for example, the equations

$$Ax = b \text{ and } Ay = c,$$

it is only necessary to factorize A once and then use two calls to the function to solve the equations, one for each of x and y.

2.5. Linear Least-squares Problems

The QR factorization discussed in Section 2.1.2 and the singular value decomposition discussed in Section 2.1.3 may readily be used to solve linear least-squares problems

$$\text{minimize } r^T r, \quad \text{where } r = b - Ax,$$

where A is a real m by n matrix, b is an m element vector and x is the n element solution vector. r, the vector whose Euclidean length is to be minimized, is the m element residual vector. In the complex case we replace $r^T r$ by $r^H r$.

The QR factorization can be used when A is of full rank with $m \geq n$ and the singular value decomposition can be used without restrictions on A.

2.5.1. Least-squares and the QR factorization

If A is a real matrix of full rank with $m \geq n$ then the QR factorization yields a non-singular upper triangular matrix R and we find that

$$r^T r = e^T e, \quad \text{where } e = Q^T b - \begin{pmatrix} R \\ 0 \end{pmatrix} x.$$

If we put

$$Q^T b = \begin{pmatrix} c_1 \\ c_2 \end{pmatrix}, \quad c_1 \text{ an } n \text{ element vector,}$$

then we see that $r^T r$ is minimized when

$$Rx = c_1,$$

in which case $r^T r = c_2^T c_2$. Thus x is the solution of the upper triangular system $Rx = c_1$.

nag_real_apply_q (f01qdc) can be used, following a call to nag_real_qr (f01qcc), to obtain $Q^T b$, and similarly for the complex case nag_complex_apply_q (f01rdc) can be used to obtain $Q^H b$ following a call to nag_complex_qr (f01rcc).

2.5.2. Least-squares and the SVD

If A is a real matrix with $\text{rank}(A) = k$ then the singular value decomposition yields a diagonal matrix D of the form

$$D = \begin{pmatrix} \Sigma & 0 \\ 0 & 0 \end{pmatrix},$$

where Σ is a k by k non-singular diagonal matrix, with positive diagonal entries $s_1, s_2, \ldots, s_k$. If we put

$$Q^T b = \begin{pmatrix} c_1 \\ c_2 \end{pmatrix} \quad \text{and} \quad P^T x = y \equiv \begin{pmatrix} y_1 \\ y_2 \end{pmatrix},$$

where c_1 and y_1 are k element vectors, we find that

$$r^T r = e^T e, \quad \text{where } e = \begin{pmatrix} c_1 - \Sigma y_1 \\ c_2 \end{pmatrix},$$

so that $r^T r$ is minimized when

$$\Sigma y_1 = c_1,$$

in which case $r^T r = c_2^T c_2$. If $k = n$ then y_2 is not present ($y = y_1$) and $x = Py$ is the unique least-squares solution. But, if $k < n$ then the $(n - k)$ elements of y_2 are arbitrary. The so-called minimal length least-squares solution, for which $x^T x$ is a minimum, is obtained by choosing

$$y_2 = 0,$$

and once again x is obtained as

$$x = Py.$$

The SVD function nag_real_svd (f02wec) has options to allow the formation of $Q^T b$ and of P^T. In the complex case nag_complex_svd (f02xec) allows the formation of $Q^H b$ and of P^H.

The determination of the rank of A from its singular values is not always straightforward, since we have to decide which singular values are negligible. If *tol* is an estimate of the relative errors in the elements of A then it is often reasonable to choose rank$(A) = k$ so that

$$s_k = \min_i s_i \quad \text{for which } s_i > tol.s_1.$$

Certainly any singular values for which $s_i \le \varepsilon.s_1$, where ε is the ***machine precision***, should be regarded as negligible.

2.6. Matrix Inversion

The inverse of n by n non-singular matrix A may be found using one of the functions for solving systems of linear equations (see Section 2.4), by choosing $B = I$ and solving

$$AX = I,$$

to give $A^{-1} = X$.

3. Available Functions

3.1. Factorizations

LU factorization and determinant	
complex matrix	`f03ahc`
real matrix	`f03afc`
Cholesky factorization and determinant	
real symmetric, positive-definite matrix	`f03aec`
Cholesky factorization	
complex Hermitian positive-definite matrix	`f01bnc`
real symmetric positive-definite variable band matrix	`f01mcc`
QR factorization, m by n matrix	
complex matrix ($m \ge n$)	`f01rcc`
real matrix ($m \ge n$)	`f01qcc`
Operations with orthogonal matrices	
compute QB or $Q^T B$, after QR factorization by f01qcc	`f01qdc`
form columns of Q, after QR factorization by f01qcc	`f01qec`
Operations with unitary matrices	
compute QB or $Q^H B$, after QR factorization by f01rcc	`f01rdc`
form columns of Q, after QR factorization by f01rcc	`f01rec`

Singular value decomposition, m by n matrix	
complex matrix	`f02xec`
real matrix	`f02wec`

3.2. Eigenvalue problems

Real unsymmetric matrix	
all eigenvalues	`f02afc`
all eigenvalues and eigenvectors	`f02agc`
selected eigenvalues and eigenvectors	`f02ecc`
Complex unsymmetric matrix	
selected eigenvalues and eigenvectors	`f02gcc`
Real symmetric matrix	
all eigenvalues	`f02aac`
all eigenvalues and eigenvectors	`f02abc`
Complex Hermitian matrix	
all eigenvalues	`f02awc`
all eigenvalues and eigenvectors	`f02axc`
Generalized real eigenproblem $Ax = \lambda Bx$	
all eigenvalues and (optionally) eigenvectors	`f02bjc`
Generalized real symmetric definite eigenproblem $Ax = \lambda Bx$	
all eigenvalues	`f02adc`
all eigenvalues and eigenvectors	`f02aec`

3.3. Linear Equations

Solution of equations	
complex matrix, multiple right-hand sides	`f04adc`
real matrix, single right-hand side	`f04arc`
Solution of equations after factorizing the matrix	
complex matrix, multiple right-hand sides (factorization by f03ahc)	`f04akc`
complex Hermitian positive-definite matrix, multiple right-hand sides (factorization by f01bnc)	`f04awc`
real matrix, multiple right-hand sides (factorization by f03afc)	`f04ajc`
real symmetric positive-definite matrix, multiple right-hand sides (factorization by f03aec)	`f04agc`
real symmetric positive-definite variable band matrix, multiple right-hand sides (factorization by f01mcc)	`f04mcc`

4. Specification of Level 2 and Level 3 BLAS in C

The Level 2 and Level 3 BLAS have been made available in the NAG C Library. Please refer to Chapter f06 for further details.

5. References

Golub G H and Van Loan C F (1989) *Matrix Computations* (2nd Edn) Johns Hopkins University Press, Baltimore.