NAG Fortran Library Routine Document

F02FAF

Note: before using this routine, please read the Users' Note for your implementation to check the interpretation of ***bold italicised*** terms and other implementation-dependent details.

1 Purpose

F02FAF computes all the eigenvalues, and optionally all the eigenvectors, of a real symmetric matrix.

2 Specification

```
SUBROUTINE F02FAF (JOB, UPLO, N, A, LDA, W, WORK, LWORK, IFAIL)

INTEGER              N, LDA, LWORK, IFAIL
double precision     A(LDA,*), W(*), WORK(LWORK)
CHARACTER*1          JOB, UPLO
```

3 Description

F02FAF computes all the eigenvalues, and optionally all the eigenvectors, of a real symmetric matrix A:

$$Az_i = \lambda_i z_i, \quad i = 1, 2, \ldots, n.$$

In other words, it computes the spectral factorization of A:

$$A = Z\Lambda Z^{\mathrm{T}},$$

where Λ is a diagonal matrix whose diagonal elements are the eigenvalues λ_i, and Z is an orthogonal matrix, whose columns are the eigenvectors z_i.

4 References

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

Parlett B N (1998) *The Symmetric Eigenvalue Problem* SIAM, Philadelphia

5 Parameters

1: JOB – CHARACTER*1 *Input*

On entry: indicates whether eigenvectors are to be computed.

JOB = 'N'

Only eigenvalues are computed.

JOB = 'V'

Eigenvalues and eigenvectors are computed.

Constraint: JOB = 'N' or 'V'.

2: UPLO – CHARACTER*1 *Input*

On entry: indicates whether the upper or lower triangular part of A is stored.

UPLO = 'U'

The upper triangular part of A is stored.

UPLO = 'L'

The lower triangular part of A is stored.

Constraint: UPLO = 'U' or 'L'.

3: N – INTEGER *Input*

On entry: n, the order of the matrix A.

Constraint: $\mathrm{N} \geq 0$.

4: A(LDA,$*$) – ***double precision*** array *Input/Output*

Note: the second dimension of the array A must be at least $\max(1, \mathrm{N})$.

On entry: the n by n symmetric matrix A.

If UPLO = 'U', the upper triangle of A must be stored and the elements of the array below the diagonal need not be set.

If UPLO = 'L', the lower triangle of A must be stored and the elements of the array above the diagonal need not be set.

On exit: If JOB = 'V', A contains the orthogonal matrix Z of eigenvectors, with the ith column holding the eigenvector z_i associated with the eigenvalue λ_i (stored in W(i)).

If UPLO = 'U', the upper triangular part of A is overwritten.

If UPLO = 'L', the lower triangular part of A if overwritten.

5: LDA – INTEGER *Input*

On entry: the first dimension of the array A as declared in the (sub)program from which F02FAF is called.

Constraint: $\mathrm{LDA} \geq \max(1, \mathrm{N})$.

6: W($*$) – ***double precision*** array *Output*

Note: the dimension of the array W must be at least $\max(1, \mathrm{N})$.

On exit: the eigenvalues in ascending order.

7: WORK(LWORK) – ***double precision*** array *Workspace*

8: LWORK – INTEGER *Input*

On entry: the dimension of the array WORK as declared in the (sub)program from which F02FAF is called. On some high-performance computers, increasing the dimension of WORK will enable the routine to run faster; a value of $64 \times \mathrm{N}$ should allow near-optimal performance on almost all machines.

Constraint: $\mathrm{LWORK} \geq \max(1, 3 \times \mathrm{N})$.

9: IFAIL – INTEGER *Input/Output*

On entry: IFAIL must be set to 0, -1 or 1. If you are unfamiliar with this parameter you should refer to Chapter P01 for details.

On exit: IFAIL = 0 unless the routine detects an error (see Section 6).

For environments where it might be inappropriate to halt program execution when an error is detected, the value -1 or 1 is recommended. If the output of error messages is undesirable, then the value 1 is recommended. Otherwise, if you are not familiar with this parameter the recommended value is 0. **When the value -1 or 1 is used it is essential to test the value of IFAIL on exit.**

6 Error Indicators and Warnings

If on entry IFAIL $= 0$ or -1, explanatory error messages are output on the current error message unit (as defined by X04AAF).

Errors or warnings detected by the routine:

IFAIL $= 1$

On entry, JOB $\neq$ 'N' or 'V',
or UPLO $\neq$ 'U' or 'L',
or N < 0,
or LDA $< \max(1, \mathrm{N})$,
or LWORK $< \max(1, 3 \times \mathrm{N})$.

IFAIL $= 2$

The *QR* algorithm failed to compute all the eigenvalues.

7 Accuracy

If λ_i is an exact eigenvalue, and $\tilde{\lambda}_i$ is the corresponding computed value, then

$$\left|\tilde{\lambda}_i - \lambda_i\right| \leq c(n)\epsilon\|A\|_2,$$

where $c(n)$ is a modestly increasing function of n, and ϵ is the ***machine precision***.

If z_i is the corresponding exact eigenvector, and $\tilde{z}_i$ is the corresponding computed eigenvector, then the angle $\theta(\tilde{z}_i, z_i)$ between them is bounded as follows:

$$\theta(\tilde{z}_i, z_i) \leq \frac{c(n)\epsilon\|A\|_2}{\min\limits_{i \neq j}\left|\lambda_i - \lambda_j\right|}.$$

Thus the accuracy of a computed eigenvector depends on the gap between its eigenvalue and all the other eigenvalues.

8 Further Comments

F02FAF calls routines from LAPACK in Chapter F08. It first reduces A to tridiagonal form T, using an orthogonal similarity transformation: $A = QTQ^{\mathrm{T}}$. If only eigenvalues are required, the routine uses a root-free variant of the symmetric tridiagonal *QR* algorithm. If eigenvectors are required, the routine first forms the orthogonal matrix Q that was used in the reduction to tridiagonal form; it then uses the symmetric tridiagonal *QR* algorithm to reduce T to Λ, using a further orthogonal transformation: $T = S\Lambda S^{\mathrm{T}}$; and at the same time accumulates the matrix $Z = QS$.

Each eigenvector z is normalized so that $\|z\|_2 = 1$ and the element of largest absolute value is positive.

The time taken by the routine is approximately proportional to n^3.

9 Example

To compute all the eigenvalues and eigenvectors of the matrix A, where

$$A = \begin{pmatrix} 4.16 & -3.12 & 0.56 & -0.10 \\ -3.12 & 5.03 & -0.83 & 1.18 \\ 0.56 & -0.83 & 0.76 & 0.34 \\ -0.10 & 1.18 & 0.34 & 1.18 \end{pmatrix}.$$

9.1 Program Text

```
*     F02FAF Example Program Text
*     Mark 16 Release. NAG Copyright 1992.
*     .. Parameters ..
      INTEGER          NIN, NOUT
      PARAMETER        (NIN=5,NOUT=6)
      INTEGER          NMAX, LDA, LWORK
      PARAMETER        (NMAX=8,LDA=NMAX,LWORK=64*NMAX)
*     .. Local Scalars ..
      INTEGER          I, IFAIL, J, N
      CHARACTER        UPLO
*     .. Local Arrays ..
      DOUBLE PRECISION A(LDA,NMAX), W(NMAX), WORK(LWORK)
*     .. External Subroutines ..
      EXTERNAL         F02FAF, X04CAF
*     .. Executable Statements ..
      WRITE (NOUT,*) 'F02FAF Example Program Results'
*     Skip heading in data file
      READ (NIN,*)
      READ (NIN,*) N
      IF (N.LE.NMAX) THEN
*
*        Read A from data file
*
         READ (NIN,*) UPLO
         IF (UPLO.EQ.'U') THEN
            READ (NIN,*) ((A(I,J),J=I,N),I=1,N)
         ELSE IF (UPLO.EQ.'L') THEN
            READ (NIN,*) ((A(I,J),J=1,I),I=1,N)
         END IF
*
*        Compute eigenvalues and eigenvectors
*
         IFAIL = 0
*
         CALL F02FAF('Vectors',UPLO,N,A,LDA,W,WORK,LWORK,IFAIL)
*
         WRITE (NOUT,*)
         WRITE (NOUT,*) 'Eigenvalues'
         WRITE (NOUT,99999) (W(I),I=1,N)
         WRITE (NOUT,*)
*
         CALL X04CAF('General',' ',N,N,A,LDA,'Eigenvectors',IFAIL)
*
      END IF
      STOP
*
99999 FORMAT (3X,(8F8.4))
      END
```

9.2 Program Data

```
F02FAF Example Program Data
  4                            :Value of N
  'L'                          :Value of UPLO
  4.16
 -3.12   5.03
  0.56  -0.83   0.76
 -0.10   1.18   0.34   1.18   :End of matrix A
```

9.3 Program Results

```
F02FAF Example Program Results

Eigenvalues
    0.1239  1.0051  1.9963  8.0047

Eigenvectors
         1       2       3       4
1   0.1859 -0.4209  0.6230 -0.6325
2   0.3791 -0.3108  0.4405  0.7521
3   0.6621  0.7210  0.1588 -0.1288
4  -0.6192  0.4543  0.6266  0.1329
```