NAG Fortran Library Routine Document

F08FCF (SSYEVD/DSYEVD)

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.

Warning. The specification of the parameters LWORK and LIWORK changed at Mark 20 in the case where JOB = 'V' and N > 1: the minimum dimension of the array WORK has been reduced whereas the minimum dimension of the array IWORK has been increased.

1 Purpose

F08FCF (SSYEVD/DSYEVD) computes all the eigenvalues, and optionally all the eigenvectors, of a real symmetric matrix. If the eigenvectors are requested, then it uses a divide and conquer algorithm to compute eigenvalues and eigenvectors. However, if only eigenvalues are required, then it uses the Pal–Walker–Kahan variant of the QL or QR algorithm.

2 Specification

```
 SUBROUTINE F08FCF(JOB, UPLO, N, A, LDA, W, WORK, LWORK, IWORK, LIWORK,
1                  INFO)
 ENTRY      ssyevd (JOB, UPLO, N, A, LDA, W, WORK, LWORK, IWORK, LIWORK,
1                  INFO)
 INTEGER            N, LDA, LWORK, IWORK(*), LIWORK, INFO
 real               A(LDA,*), W(*), WORK(*)
 CHARACTER*1        JOB, UPLO
```

The ENTRY statement enables the routine to be called by its LAPACK name.

3 Description

This routine computes all the eigenvalues, and optionally all the eigenvectors, of a real symmetric matrix A. In other words, it can compute the spectral factorization of A as

$$A = Z\Lambda Z^T,$$

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

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

4 References

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

5 Parameters

1: JOB – CHARACTER*1 *Input*

On entry: indicates whether eigenvectors are computed as follows:

if JOB = 'N', only eigenvalues are computed;

if 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 as follows:

if UPLO = 'U', the upper triangular part of A is stored;

if 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,*) – ***real*** 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 triangular part of A must be stored and the elements of the array below the diagonal are not referenced; if UPLO = 'L', the lower triangular part of A must be stored and the elements of the array above the diagonal are not referenced.

On exit: if JOB = 'V', this is overwritten by the orthogonal matrix Z which contains the eigenvectors of A.

5: LDA – INTEGER *Input*

On entry: the first dimension of the array A as declared in the (sub)program from which F08FCF (SSYEVD/DSYEVD) is called.

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

6: W(*) – ***real*** array *Output*

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

On exit: the eigenvalues of the matrix A in ascending order.

7: WORK(*) – ***real*** array *Workspace*

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

On exit: if INFO = 0, WORK(1) contains the required minimal size of LWORK.

8: LWORK – INTEGER *Input*

On entry: the dimension of the array WORK as declared in the (sub)program from which F08FCF (SSYEVD/DSYEVD) is called, unless LWORK = −1, in which case a workspace query is assumed and the routine only calculates the minimum dimension of WORK.

Constraints:

if $\mathrm{N} \leq 1$, $\mathrm{LWORK} \geq 1$ or LWORK = −1;
if JOB = 'N' and $\mathrm{N} > 1$, $\mathrm{LWORK} \geq 2 \times \mathrm{N} + 1$ or LWORK = −1;
if JOB = 'V' and $\mathrm{N} > 1$, $\mathrm{LWORK} \geq 2 \times \mathrm{N}^2 + 6 \times \mathrm{N} + 1$ or LWORK = −1.

9: IWORK(*) – INTEGER array *Workspace*

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

On exit: if INFO = 0, IWORK(1) contains the required minimal size of LIWORK.

10: LIWORK – INTEGER *Input*

On entry: the dimension of the array IWORK as declared in the (sub)program from which F08FCF (SSYEVD/DSYEVD) is called, unless LIWORK $= -1$, in which case a workspace query is assumed and the routine only calculates the minimum dimension of IWORK.

Constraints:

if N ≤ 1, LIWORK ≥ 1 or LIWORK $= -1$,
if JOB = 'N' and N > 1, LIWORK ≥ 1 or LIWORK $= -1$,
if JOB = 'V' and N > 1, LIWORK $\geq 5 \times$ N $+ 3$ or LIWORK $= -1$.

11: INFO – INTEGER *Output*

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

6 Error Indicators and Warnings

Errors or warnings detected by the routine:

INFO < 0

If INFO $= -i$, the ith parameter had an illegal value. An explanatory message is output, and execution of the program is terminated.

INFO > 0

If INFO $= i$, the algorithm failed to converge; i indicates the number of elements of an intermediate tridiagonal form which did not converge to zero.

7 Accuracy

The computed eigenvalues and eigenvectors are exact for a nearby matrix $A + E$, where

$$\|E\|_2 = O(\epsilon)\|A\|_2,$$

and ϵ is the ***machine precision***.

8 Further Comments

The complex analogue of this routine is F08FQF (CHEEVD/ZHEEVD).

9 Example

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

$$A = \begin{pmatrix} 1.0 & 2.0 & 3.0 & 4.0 \\ 2.0 & 2.0 & 3.0 & 4.0 \\ 3.0 & 3.0 & 3.0 & 4.0 \\ 4.0 & 4.0 & 4.0 & 4.0 \end{pmatrix}.$$

9.1 Program Text

Note: the listing of the example program presented below uses ***bold italicised*** terms to denote precision-dependent details. Please read the Users' Note for your implementation to check the interpretation of these terms. As explained in the Essential Introduction to this manual, the results produced may not be identical for all implementations.

```
*     F08FCF Example Program Text.
*     Mark 20 Revised. NAG Copyright 2001.
*     .. Parameters ..
      INTEGER          NIN, NOUT
      PARAMETER        (NIN=5,NOUT=6)
      INTEGER          NMAX, LDA
      PARAMETER        (NMAX=8,LDA=NMAX)
      INTEGER          LWORK, LIWORK
```

```
      PARAMETER        (LWORK=2*NMAX*NMAX+6*NMAX+1,LIWORK=5*NMAX+3)
*     .. Local Scalars ..
      INTEGER          I, IFAIL, INFO, J, N
      CHARACTER        JOB, UPLO
*     .. Local Arrays ..
      real             A(LDA,NMAX), W(NMAX), WORK(LWORK)
      INTEGER          IWORK(LIWORK)
*     .. External Subroutines ..
      EXTERNAL         ssyevd, X04CAF
*     .. Executable Statements ..
      WRITE (NOUT,*) 'F08FCF 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
*
         READ (NIN,*) JOB
*
*        Calculate all the eigenvalues and eigenvectors of A
*
         CALL ssyevd(JOB,UPLO,N,A,LDA,W,WORK,LWORK,IWORK,LIWORK,INFO)
*
         WRITE (NOUT,*)
         IF (INFO.GT.0) THEN
            WRITE (NOUT,*) 'Failure to converge.'
         ELSE
*
*           Print eigenvalues and eigenvectors
*
            WRITE (NOUT,*) 'Eigenvalues'
            WRITE (NOUT,99999) (W(I),I=1,N)
            WRITE (NOUT,*)
            IFAIL = 0
*
            CALL X04CAF('General',' ',N,N,A,LDA,'Eigenvectors',IFAIL)
*
         END IF
      END IF
      STOP
*
99999 FORMAT (3X,(8F8.4))
      END
```

9.2 Program Data

```
F08FCF Example Program Data
  4                          :Value of N
  'L'                        :Value of UPLO
  1.0
  2.0   2.0
  3.0   3.0   3.0
  4.0   4.0   4.0   4.0      :End of matrix A
  'V'                        :Value of JOB
```

9.3 Program Results

```
 F08FCF Example Program Results

 Eigenvalues
    -2.0531 -0.5146 -0.2943 12.8621

 Eigenvectors
          1       2       3       4
 1   0.7003 -0.5144 -0.2767 -0.4103
 2   0.3592  0.4851  0.6634 -0.4422
 3  -0.1569  0.5420 -0.6504 -0.5085
 4  -0.5965 -0.4543  0.2457 -0.6144
```