

NAG Fortran Library Routine Document

F08FDF (DSYEVR)

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

F08FDF (DSYEVR) computes selected eigenvalues and, optionally, eigenvectors of a real n by n symmetric matrix A . Eigenvalues and eigenvectors can be selected by specifying either a range of values or a range of indices for the desired eigenvalues.

2 Specification

```

SUBROUTINE F08FDF (JOBZ, RANGE, UPLO, N, A, LDA, VL, VU, IL, IU, ABSTOL,
1                M, W, Z, LDZ, ISUPPZ, WORK, LWORK, IWORK, LIWORK,
2                INFO)
    INTEGER       N, LDA, IL, IU, M, LDZ, ISUPPZ(*), LWORK, IWORK(*),
1                LIWORK, INFO
    double precision A(LDA,*), VL, VU, ABSTOL, W(*), Z(LDZ,*), WORK(*)
    CHARACTER*1   JOBZ, RANGE, UPLO

```

The routine may be called by its LAPACK name *dsyevr*.

3 Description

The symmetric matrix is first reduced to a tridiagonal matrix T , using orthogonal similarity transformations. Then whenever possible, F08FDF (DSYEVR) computes the eigenspectrum using Relatively Robust Representations. F08FDF (DSYEVR) computes eigenvalues by the dqds algorithm, while orthogonal eigenvectors are computed from various 'good' LDL^T representations (also known as Relatively Robust Representations). Gram–Schmidt orthogonalisation is avoided as far as possible. More specifically, the various steps of the algorithm are as follows. For the i th unreduced block of T ,

- compute $T - \sigma_i I = L_i D_i L_i^T$, such that $L_i D_i L_i^T$ is a relatively robust representation,
- compute the eigenvalues, λ_j , of $L_i D_i L_i^T$ to high relative accuracy by the dqds algorithm,
- if there is a cluster of close eigenvalues, 'choose' σ_i close to the cluster, and go to step (a),
- given the approximate eigenvalue λ_j of $L_i D_i L_i^T$, compute the corresponding eigenvector by forming a rank-revealing twisted factorization.

The desired accuracy of the output can be specified by the input parameter ABSTOL. For more details, see Dhillon (1997) and Parlett and Dhillon (2000).

4 References

Anderson E, Bai Z, Bischof C, Blackford S, Demmel J, Dongarra J J, Du Croz J J, Greenbaum A, Hammarling S, McKenney A and Sorensen D (1999) *LAPACK Users' Guide* (3rd Edition) SIAM, Philadelphia URL: <http://www.netlib.org/lapack/lug>

Barlow J and Demmel J W (1990) Computing accurate eigensystems of scaled diagonally dominant matrices *SIAM J. Numer. Anal.* **27** 762–791

Demmel J W and Kahan W (1990) Accurate singular values of bidiagonal matrices *SIAM J. Sci. Statist. Comput.* **11** 873–912

Dhillon I (1997) A new $O(n^2)$ algorithm for the symmetric tridiagonal eigenvalue/eigenvector problem *Computer Science Division Technical Report No. UCB//CSD-97-971* UC Berkeley

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

Parlett B N and Dhillon I S (2000) Relatively robust representations of symmetric tridiagonals *Linear Algebra Appl.* **309** 121–151

5 Parameters

- 1: JOBZ – CHARACTER*1 *Input*
On entry: if JOBZ = 'N', compute eigenvalues only.
 If JOBZ = 'V', compute eigenvalues and eigenvectors.
Constraint: JOBZ = 'N' or 'V'.
- 2: RANGE – CHARACTER*1 *Input*
On entry: if RANGE = 'A', all eigenvalues will be found.
 If RANGE = 'V', all eigenvalues in the half-open interval (VL, VU] will be found.
 If RANGE = 'I', the ILth to IUth eigenvalues will be found.
 For RANGE = 'V' or 'I' and $IU - IL < N - 1$, F08JJF (DSTEBZ) and F08JKF (DSTEIN) are called.
- 3: UPLO – CHARACTER*1 *Input*
On entry: if UPLO = 'U', the upper triangle of A is stored.
 If UPLO = 'L', the lower triangle of A is stored.
- 4: N – INTEGER *Input*
On entry: n , the order of the matrix A .
Constraint: $N \geq 0$.
- 5: A(LDA,*) – **double precision** array *Input/Output*
Note: the second dimension of the array A must be at least $\max(1, N)$.
On entry: the n by n symmetric matrix A .
 If UPLO = 'U', the leading n by n upper triangular part of A contains the upper triangular part of the matrix A .
 If UPLO = 'L', the leading n by n lower triangular part of A contains the lower triangular part of the matrix A .
On exit: the lower triangle (if UPLO = 'L') or the upper triangle (if UPLO = 'U') of A , including the diagonal, is destroyed.
- 6: LDA – INTEGER *Input*
On entry: the first dimension of the array A as declared in the (sub)program from which F08FDF (DSYEVR) is called.
Constraint: $LDA \geq \max(1, N)$.
- 7: VL – **double precision** *Input*
 8: VU – **double precision** *Input*
On entry: if RANGE = 'V', the lower and upper bounds of the interval to be searched for eigenvalues.
Constraint: $VL < VU$.

If RANGE = 'A' or 'I', VL and VU are not referenced.

- 9: IL – INTEGER *Input*
 10: IU – INTEGER *Input*

On entry: if RANGE = 'I', the indices (in ascending order) of the smallest and largest eigenvalues to be returned.

If RANGE = 'A' or 'V', IL and IU are not referenced.

Constraints:

if $N = 0$, $IL = 1$ and $IU = 0$;
 if $N > 0$, $1 \leq IL \leq IU \leq N$.

- 11: ABSTOL – *double precision* *Input*

On entry: the absolute error tolerance for the eigenvalues. An approximate eigenvalue is accepted as converged when it is determined to lie in an interval $[a, b]$ of width less than or equal to

$$ABSTOL + \epsilon \max(|a|, |b|),$$

where ϵ is the *machine precision*. If ABSTOL is less than or equal to zero, then $\epsilon \|T\|_1$ will be used in its place, where T is the tridiagonal matrix obtained by reducing A to tridiagonal form. See Demmel and Kahan (1990).

If high relative accuracy is important, set ABSTOL to X02AMF(), although doing so does not currently guarantee that eigenvalues are computed to high relative accuracy. See Barlow and Demmel (1990) for a discussion of which matrices can define their eigenvalues to high relative accuracy.

- 12: M – INTEGER *Output*

On exit: the total number of eigenvalues found.

If RANGE = 'A', $M = N$.

If RANGE = 'I', $M = IU - IL + 1$.

Constraint: $0 \leq M \leq N$.

- 13: W(*) – *double precision* array *Output*

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

On exit: the first M elements contain the selected eigenvalues in ascending order.

- 14: Z(LDZ,*) – *double precision* array *Output*

Note: the second dimension of the array Z must be at least $\max(1, M)$.

On exit: if JOBZ = 'V', then if INFO = 0, the first m columns of Z contain the orthonormal eigenvectors of the matrix A corresponding to the selected eigenvalues, with the i th column of Z holding the eigenvector associated with $W(i)$.

If JOBZ = 'N', Z is not referenced.

Note: the user must ensure that at least $\max(1, M)$ columns are supplied in the array Z; if RANGE = 'V', the exact value of M is not known in advance and an upper bound must be used.

- 15: LDZ – INTEGER *Input*

On entry: the first dimension of the array Z as declared in the (sub)program from which F08FDF (DSYEVR) is called.

Constraints:

if JOBZ = 'V', $LDZ \geq \max(1, N)$;
 $LDZ \geq 1$ otherwise.

- 16: ISUPPZ(*) – INTEGER array *Output*
Note: the dimension of the array ISUPPZ must be at least $\max(1, 2 \times M)$.
On exit: the support of the eigenvectors in Z, i.e., the indices indicating the nonzero elements in Z. The i th eigenvector is nonzero only in elements ISUPPZ($2 \times i - 1$) through ISUPPZ($2 \times i$). Implemented only for RANGE = 'A' or 'I' and $IU - IL = N - 1$.
- 17: WORK(*) – *double precision* array *Workspace*
Note: the dimension of the array WORK must be at least $\max(1, LWORK)$.
On exit: if INFO = 0, WORK(1) returns the optimal LWORK.
- 18: LWORK – INTEGER *Input*
On entry: the dimension of the array WORK as declared in the (sub)program from which F08FDF (DSYEVR) is called.
 For optimal efficiency, $LWORK \geq (nb + 6) \times N$, where nb is the largest optimal block size for F08FEF (DSYTRD) and F08FGF (DORMTR).
 If $LWORK = -1$, a workspace query is assumed; the routine only calculates the optimal sizes of the WORK and IWORK arrays, returns these values as the first entries of the WORK and IWORK arrays, and no error message related to LWORK or LIWORK is issued.
Constraint: $LWORK \geq \max(1, 26 \times N)$.
- 19: IWORK(*) – INTEGER array *Workspace*
Note: the dimension of the array IWORK must be at least $\max(1, LIWORK)$.
On exit: if INFO = 0, IWORK(1) returns the optimal LIWORK.
- 20: LIWORK – INTEGER *Input*
On entry: the dimension of the array IWORK as declared in the (sub)program from which F08FDF (DSYEVR) is called.
 If $LIWORK = -1$, a workspace query is assumed; the routine only calculates the optimal sizes of the WORK and IWORK arrays, returns these values as the first entries of the WORK and IWORK arrays, and no error message related to LWORK or LIWORK is issued.
Constraint: $LIWORK \geq \max(1, 10 \times N)$.
- 21: 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 i th argument had an illegal value.

INFO > 0

F08FDF (DSYEVR) failed to converge.

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*. See Section 4.7 of Anderson *et al.* (1999) for further details.

8 Further Comments

The total number of floating point operations is proportional to n^3 .

The complex analogue of this routine is F08FRF (ZHEEVR).

9 Example

To find the eigenvalues with indices in the range $[2,3]$, and the corresponding eigenvectors, of the symmetric matrix

$$A = \begin{pmatrix} 1 & 2 & 3 & 4 \\ 2 & 2 & 3 & 4 \\ 3 & 3 & 3 & 4 \\ 4 & 4 & 4 & 4 \end{pmatrix}.$$

Information on required and provided workspace is also output.

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.

```
*      F08FDF Example Program Text
*      Mark 21 Release. NAG Copyright 2004.
*      .. Parameters ..
      INTEGER          NIN, NOUT
      PARAMETER        (NIN=5,NOUT=6)
      INTEGER          NB, NMAX, MMAX
      PARAMETER        (NB=64,NMAX=10,MMAX=5)
      INTEGER          LDA, LDZ, LIWORK, LWORK
      PARAMETER        (LDA=NMAX,LDZ=NMAX,LIWORK=10*NMAX,LWORK=(NB+6)
+                      *NMAX)
      DOUBLE PRECISION ZERO
      PARAMETER        (ZERO=0.0D+0)
*      .. Local Scalars ..
      DOUBLE PRECISION ABSTOL, VL, VU
      INTEGER          I, IFAIL, IL, INFO, IU, J, LIWOPT, LWKOPT, M, N
*      .. Local Arrays ..
      DOUBLE PRECISION A(LDA,NMAX), W(NMAX), WORK(LWORK), Z(LDZ,MMAX)
      INTEGER          ISUPPZ(2*MMAX), IWORK(LIWORK)
*      .. External Subroutines ..
      EXTERNAL        DSYEVR, X04CAF
*      .. Executable Statements ..
      WRITE (NOUT,*) 'F08FDF Example Program Results'
      WRITE (NOUT,*)
*      Skip heading in data file and read N and the lower and upper
*      indices of the smallest and largest eigenvalues to be found
      READ (NIN,*)
      READ (NIN,*) N, IL, IU
      IF (N.LE.NMAX .AND. (IU-IL+1).LE.MMAX) THEN
*
*          Read the upper triangular part of the matrix A from data file
*
*          READ (NIN,*) ((A(I,J),J=I,N),I=1,N)
*
*          Set the absolute error tolerance for eigenvalues. With ABSTOL
*          set to zero, the default value is used instead
*

```

```

      ABSTOL = ZERO
*
*   Solve the symmetric eigenvalue problem
*
      CALL DSYEVR('Vectors','I','Upper',N,A,LDA,VL,VU,IL,IU,ABSTOL,M,
+           W,Z,LDZ,ISUPPZ,WORK,LWORK,IWORK,LIWORK,INFO)
*
      IF (INFO.EQ.0) THEN
*
*       Print solution
*
          WRITE (NOUT,*) 'Selected eigenvalues'
          WRITE (NOUT,99999) (W(J),J=1,M)
*
          IFAIL = 0
          CALL X04CAF('General',' ',N,M,Z,LDZ,'Selected eigenvectors',
+           IFAIL)
*
*       Print workspace information
*
          LWKOPT = WORK(1)
          LIWOPT = IWORK(1)
          IF (LWORK.LT.LWKOPT) THEN
              WRITE (NOUT,*)
              WRITE (NOUT,99998) 'Optimum real workspace required = ',
+           LWKOPT, 'Real workspace provided = ', LWORK
          END IF
          IF (LIWORK.LT.LIWOPT) THEN
              WRITE (NOUT,*)
              WRITE (NOUT,99998) 'Integer workspace required = ',
+           LIWOPT, 'Integer workspace provided = ', LIWORK
          END IF
          ELSE
              WRITE (NOUT,99997) 'Failure in DSYEVR. INFO =', INFO
          END IF
          ELSE
              WRITE (NOUT,*) 'NMAX and/or MMAX too small'
          END IF
          STOP
*
99999 FORMAT (3X,(8F8.4))
99998 FORMAT (1X,A,I5,/1X,A,I5)
99997 FORMAT (1X,A,I5)
      END

```

9.2 Program Data

F08FDF Example Program Data

```

4      2      3           :Values of N, IL and IU

1.0   2.0   3.0   4.0
      2.0   3.0   4.0
           3.0   4.0
           4.0 :End of matrix A

```

9.3 Program Results

F08FDF Example Program Results

```

Selected eigenvalues
-0.5146 -0.2943
Selected eigenvectors
      1      2
1  -0.5144  0.2767
2   0.4851 -0.6634
3   0.5420  0.6504
4  -0.4543 -0.2457

```