NAG Fortran Library Routine Document

F08JDF (DSTEVR)

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

F08JDF (DSTEVR) computes selected eigenvalues and, optionally, eigenvectors of a real n by n symmetric tridiagonal matrix T. 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 F08JDF (JOBZ, RANGE, N, D, E, VL, VU, IL, IU, ABSTOL, M, W,
1                   Z, LDZ, WORK, LWORK, IWORK, LIWORK, JFAIL, INFO)
 INTEGER            N, IL, IU, M, LDZ, LWORK(*), IWORK(*), LIWORK(*),
1                   JFAIL(*), INFO
 double precision   D(*), E(*), VL, VU, ABSTOL, W(*), Z(LDZ,*), WORK(*)
 CHARACTER*1        JOBZ, RANGE
```

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

3 Description

Whenever possible F08JDF (DSTEVR) computes the eigenspectrum using Relatively Robust Representations. F08JDF (DSTEVR) computes eigenvalues by the **dqds** algorithm, while orthogonal eigenvectors are computed from varous '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 ith unreduced block of T:

(a) 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.

(b) compute the eigenvalues, λ_j, of $L_i D_i L_i{}^T$ to high relative accuracy by the dqds algorithm.

(c) if there is a cluster of close eigenvalues, 'choose' σ_i close to the cluser, and go to a.

(d) 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 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 $(\mathrm{VL}, \mathrm{VU}]$ will be found.

If RANGE = 'I', the ILth to IUth eigenvalues will be found.

3: N – INTEGER *Input*

On entry: n, the order of the matrix.

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

4: D($*$) – ***double precision*** array *Input/Output*

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

On entry: the n diagonal elements of the tridiagonal matrix A.

On exit: may be multiplied by a constant factor chosen to avoid over/underflow in computing the eigenvalues.

5: E($*$) – ***double precision*** array *Input/Output*

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

On entry: the $(n - 1)$ sub-diagonal elements of the tridiagonal matrix T.

On exit: may be multiplied by a constant factor chosen to avoid over/underflow in computing the eigenvalues.

6: VL – ***double precision*** *Input*
7: 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.

8: IL – INTEGER *Input*
9: 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 $\mathrm{N} = 0$, $\mathrm{IL} = 1$ and $\mathrm{IU} = 0$;
if $\mathrm{N} > 0$, $1 \leq \mathrm{IL} \leq \mathrm{IU} \leq \mathrm{N}$.

10: 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

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

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. 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.

11: M – INTEGER *Output*

On exit: the total number of eigenvalues found.

If RANGE = 'A', $\mathrm{M} = \mathrm{N}$.

If RANGE = 'I', $\mathrm{M} = \mathrm{IU} - \mathrm{IL} + 1$.

Constraint: $0 \leq \mathrm{M} \leq \mathrm{N}$.

12: W(∗) – ***double precision*** array *Output*

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

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

13: Z(LDZ,∗) – ***double precision*** array *Output*

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

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

If an eigenvector fails to converge ($\mathrm{INFO} > 0$), then that column of Z contains the latest approximation to the eigenvector, and the index of the eigenvector is returned in JFAIL. If JOBZ = 'N', Z is not referenced.

Note: the user must ensure that at least $\max(1, \mathrm{M})$ 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.

14: LDZ – INTEGER *Input*

On entry: the first dimension of the array Z as declared in the (sub)program from which F08JDF (DSTEVR) is called.

Constraints:

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

15: WORK(∗) – ***double precision*** array *Workspace*

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

16: LWORK(∗) – INTEGER array *Input*

On entry: the dimension of the array WORK as declared in the (sub)program from which F08JDF (DSTEVR) is called.

If LWORK $= -1$, a workspace query is assumed; the routine only calculates the optimal size 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 WORK or IWORK is issued.

Constraint: LWORK $\geq \max(1, 20 \times \mathrm{N})$.

17: IWORK$(*)$ – INTEGER array *Workspace*

Note: the dimension of the array IWORK must be at least $\max(1, 5 \times \mathrm{N})$.

18: LIWORK$(*)$ – INTEGER array *Input*

On entry: the dimension of the array IWORK as declared in the (sub)program from which F08JDF (DSTEVR) is called.

If LIWORK $= -1$, a workspace query is assumed; the routine only calculates the optimal size 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 $\geq \max(1, 120 \times \mathrm{N})$.

19: JFAIL$(*)$ – INTEGER array *Output*

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

On exit: if JOBZ $=$ 'V', then if INFO $= 0$, the first M elements of JFAIL are zero. If INFO > 0, JFAIL contains the indices of the eigenvectors that failed to converge.

If JOBZ $=$ 'N', JFAIL is not referenced.

20: 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 argument had an illegal value.

INFO > 0

If INFO $= i$, then i eigenvectors failed to converge. Their indices are stored in array JFAIL. Please see ABSTOL.

7 Accuracy

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

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

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^2 if JOBZ $=$ 'N', and is proportional to n^3 if JOBZ $=$ 'V' and RANGE $=$ 'A', otherwise the number of floating point operations will depend upon the number of computed eigenvectors.

9 Example

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

$$T = \begin{pmatrix} 1 & 1 & 0 & 0 \\ 1 & 4 & 2 & 0 \\ 0 & 2 & 9 & 3 \\ 0 & 0 & 3 & 16 \end{pmatrix}.$$

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.

```
*     F08JDF Example Program Text
*     Mark 21.  NAG Copyright 2004.
*     .. Parameters ..
      INTEGER          NIN, NOUT
      PARAMETER        (NIN=5,NOUT=6)
      INTEGER          NMAX, MMAX
      PARAMETER        (NMAX=10,MMAX=5)
      INTEGER          LDZ, LIWORK, LWORK
      PARAMETER        (LDZ=NMAX,LIWORK=10*NMAX,LWORK=20*NMAX)
      DOUBLE PRECISION ZERO
      PARAMETER        (ZERO=0.0D+0)
*     .. Local Scalars ..
      DOUBLE PRECISION ABSTOL, VL, VU
      INTEGER          I, IFAIL, IL, INFO, IU, J, LIWOPT, LWOPT, M, N
*     .. Local Arrays ..
      DOUBLE PRECISION D(NMAX), E(NMAX-1), W(NMAX), WORK(LWORK),
     +                 Z(LDZ,MMAX)
      INTEGER          ISUPPZ(2*NMAX), IWORK(LIWORK)
*     .. External Subroutines ..
      EXTERNAL         DSTEVR, X04CAF
*     .. Executable Statements ..
      WRITE (NOUT,*) 'F08JDF Example Program Results'
      WRITE (NOUT,*)
*     Skip heading in data file and read N and the lower and upper
*     indices of the eigenvalues to be found
      READ (NIN,*)
      READ (NIN,*) N, IL, IU
      IF (N.LE.NMAX .AND. (IU-IL+1).LE.MMAX) THEN
*
*        Read the diagonal and off-diagonal elements of the matrix A
*        from data file
*
         READ (NIN,*) (D(I),I=1,N)
         READ (NIN,*) (E(I),I=1,N-1)
*
*        Set the absolute error tolerance for eigenvalues.  With ABSTOL
*        set to zero, the default value is used instead
*
         ABSTOL = ZERO
*
*        Solve the symmetric tridiagonal eigenvalue problem
*
         CALL DSTEVR('Vectors','Indices',N,D,E,VL,VU,IL,IU,ABSTOL,M,W,Z,
     +               LDZ,ISUPPZ,WORK,LWORK,IWORK,LIWORK,INFO)
         LWOPT = WORK(1)
         LIWOPT = IWORK(1)
*
         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)
         ELSE
            WRITE (NOUT,99998) 'Failure in DSTEVR. INFO =', INFO
         END IF
*
*        Print workspace information
*
         IF (LWORK.LT.LWOPT) THEN
            WRITE (NOUT,*)
            WRITE (NOUT,99997) 'Real workspace required = ', LWOPT,
     +        'Real workspace provided = ', LWORK
         END IF
         IF (LIWORK.LT.LIWOPT) THEN
            WRITE (NOUT,*)
            WRITE (NOUT,99997) 'Integer workspace required = ', LIWOPT,
     +        'Integer workspace provided = ', LIWORK
         END IF
       ELSE
          WRITE (NOUT,*) 'NMAX and/or MMAX too small'
       END IF
       STOP
*
99999 FORMAT (3X,(8F8.4))
99998 FORMAT (1X,A,I5)
99997 FORMAT (1X,A,I5,/1X,A,I5)
      END
```

9.2 Program Data

```
F08JDF Example Program Data

  4    2    3          :Values of N, IL and IU

  1.0  4.0  9.0  16.0  :End of diagonal elements
  1.0  2.0  3.0        :End of off-diagonal elements
```

9.3 Program Results

```
 F08JDF Example Program Results

 Selected eigenvalues
     3.5470  8.6578
 Selected eigenvectors
          1       2
 1   0.3388  0.0494
 2   0.8628  0.3781
 3  -0.3648  0.8558
 4   0.0879 -0.3497
```