NAG 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,      &
                   Z, LDZ, ISUPPZ, WORK, LWORK, IWORK, LIWORK, INFO)
INTEGER              N, IL, IU, M, LDZ, ISUPPZ(*), LWORK,                     &
                     IWORK(max(1,LIWORK)), LIWORK, INFO
REAL (KIND=nag_wp) D(*), E(*), VL, VU, ABSTOL, W(*), Z(LDZ,*),               &
                     WORK(max(1,LWORK))
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 various 'good' LDL^{T} representations (also known as Relatively Robust Representations). Gram–Schmidt orthogonalization 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 $T - \sigma_i I = L_i D_i L_i^{\mathrm{T}}$, such that $L_i D_i L_i^{\mathrm{T}}$ is a relatively robust representation,

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

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

(d) given the approximate eigenvalue λ_j of $L_i D_i L_i^{\mathrm{T}}$, compute the corresponding eigenvector by forming a rank-revealing twisted factorization.

The desired accuracy of the output can be specified by the argument 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 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 Arguments

1: JOBZ – CHARACTER(1) *Input*

On entry: indicates whether eigenvectors are computed.

JOBZ = 'N'
Only eigenvalues are computed.

JOBZ = 'V'
Eigenvalues and eigenvectors are computed.

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.

Constraint: RANGE = 'A', 'V' or 'I'.

3: N – INTEGER *Input*

On entry: n, the order of the matrix.

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

4: D($*$) – REAL (KIND=nag_wp) 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 T.

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

5: E($*$) – REAL (KIND=nag_wp) array *Input/Output*

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

On entry: the $(n - 1)$ subdiagonal 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 – REAL (KIND=nag_wp) *Input*
7: VU – REAL (KIND=nag_wp) *Input*

On entry: if RANGE = 'V', the lower and upper bounds of the interval to be searched for eigenvalues.

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

Constraint: if RANGE = 'V', $\mathrm{VL} < \mathrm{VU}$.

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 RANGE = 'I' and N = 0, IL = 1 and IU = 0;
if RANGE = 'I' and N > 0, $1 \leq \text{IL} \leq \text{IU} \leq \text{N}$.

10: ABSTOL – REAL (KIND=nag_wp) *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

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

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. $0 \leq \text{M} \leq \text{N}$.

If RANGE = 'A', M = N.

If RANGE = 'I', M = IU − IL + 1.

12: W($*$) – REAL (KIND=nag_wp) array *Output*

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

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

13: Z(LDZ, $*$) – REAL (KIND=nag_wp) array *Output*

Note: the second dimension of the array Z must be at least $\max(1, \text{M})$ if JOBZ = 'V', and at least 1 otherwise.

On exit: if JOBZ = 'V', 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 W(i).

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

Note: you must ensure that at least $\max(1, \text{M})$ columns are supplied in the array Z; if RANGE = 'V', the exact value of M is not known in advance and an upper bound of at least N 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', $\text{LDZ} \geq \max(1, \text{N})$;
otherwise $\text{LDZ} \geq 1$.

15: ISUPPZ($*$) – INTEGER array *Output*

Note: the dimension of the array ISUPPZ must be at least $\max(1, 2 \times \mathrm{M})$.

On exit: the support of the eigenvectors in Z, i.e., the indices indicating the nonzero elements in Z. The ith 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.

16: WORK$(\max(1, \mathrm{LWORK}))$ – REAL (KIND=nag_wp) array *Workspace*

On exit: if INFO $= 0$, WORK(1) contains the minimum value of LWORK required for optimal performance.

17: LWORK – INTEGER *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 minimum 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 $\geq \max(1, 20 \times \mathrm{N})$.

18: IWORK$(\max(1, \mathrm{LIWORK}))$ – INTEGER array *Workspace*

On exit: if INFO $= 0$, IWORK(1) returns the minimum LIWORK.

19: LIWORK – INTEGER *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 minimum 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 $\geq \max(1, 120 \times \mathrm{N})$.

20: INFO – INTEGER *Output*

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

6 Error Indicators and Warnings

INFO < 0

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

INFO > 0

An internal error has occurred in this routine. Please refer to INFO in F08JJF (DSTEBZ).

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 Parallelism and Performance

F08JDF (DSTEVR) is threaded by NAG for parallel execution in multithreaded implementations of the NAG Library.

F08JDF (DSTEVR) makes calls to BLAS and/or LAPACK routines, which may be threaded within the vendor library used by this implementation. Consult the documentation for the vendor library for further information.

Please consult the X06 Chapter Introduction for information on how to control and interrogate the OpenMP environment used within this routine. Please also consult the Users' Note for your implementation for any additional implementation-specific information.

9 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.

10 Example

This example finds the eigenvalues with indices in the range $[2, 3]$, and the corresponding eigenvectors, of the symmetric tridiagonal matrix

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

10.1 Program Text

```
    Program f08jdfe

!     F08JDF Example Program Text

!     Mark 26 Release. NAG Copyright 2016.

!     .. Use Statements ..
      Use nag_library, Only: dstevr, nag_wp, x04caf
!     .. Implicit None Statement ..
      Implicit None
!     .. Parameters ..
      Real (Kind=nag_wp), Parameter    :: zero = 0.0_nag_wp
      Integer, Parameter               :: nin = 5, nout = 6
!     .. Local Scalars ..
      Real (Kind=nag_wp)               :: abstol, vl, vu
      Integer                          :: ifail, il, info, iu, ldz, liwork,    &
                                          lwork, m, n
!     .. Local Arrays ..
      Real (Kind=nag_wp), Allocatable  :: d(:), e(:), w(:), work(:), z(:,:)
      Real (Kind=nag_wp)               :: rdum(1)
      Integer                          :: idum(1)
      Integer, Allocatable             :: isuppz(:), iwork(:)
!     .. Intrinsic Procedures ..
      Intrinsic                        :: max, nint
!     .. 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
      ldz = n
      m = n
      Allocate (d(n),e(n-1),w(n),z(ldz,m),isuppz(2*n))
```

```
!     Use routine workspace query to get optimal workspace.
      lwork = -1
      liwork = -1
!     The NAG name equivalent of dstevr is f08jdf
      Call dstevr('Vectors','Indices',n,d,e,vl,vu,il,iu,abstol,m,w,z,ldz,      &
        isuppz,rdum,lwork,idum,liwork,info)

!     Make sure that there is enough workspace for block size nb.
      lwork = max(20*n,nint(rdum(1)))
      liwork = max(10*n,idum(1))
      Allocate (work(lwork),iwork(liwork))

!     Read the diagonal and off-diagonal elements of the matrix A
!     from data file

      Read (nin,*) d(1:n)
      Read (nin,*) e(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
!     The NAG name equivalent of dstevr is f08jdf
      Call dstevr('Vectors','Indices',n,d,e,vl,vu,il,iu,abstol,m,w,z,ldz,      &
        isuppz,work,lwork,iwork,liwork,info)

      If (info==0) Then

!       Print solution

        Write (nout,*) 'Selected eigenvalues'
        Write (nout,99999) w(1:m)
        Flush (nout)

!       ifail: behaviour on error exit
!              =0 for hard exit, =1 for quiet-soft, =-1 for noisy-soft
        ifail = 0
        Call x04caf('General',' ',n,m,z,ldz,'Selected eigenvectors',ifail)

      Else
        Write (nout,99998) 'Failure in DSTEVR. INFO =', info
      End If

99999 Format (3X,(8F8.4))
99998 Format (1X,A,I5)
    End Program f08jdfe
```

10.2 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
```

10.3 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
```