NAG Library Routine Document

F08JHF (DSTEDC)

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

F08JHF (DSTEDC) computes all the eigenvalues and, optionally, all the eigenvectors of a real n by n symmetric tridiagonal matrix, or of a real full or banded symmetric matrix which has been reduced to tridiagonal form.

2 Specification

```
SUBROUTINE F08JHF (COMPZ, N, D, E, Z, LDZ, WORK, LWORK, IWORK, LIWORK,          &
                   INFO)
INTEGER             N, LDZ, LWORK, IWORK(max(1,LIWORK)), LIWORK, INFO
REAL (KIND=nag_wp) D(*), E(*), Z(LDZ,*), WORK(max(1,LWORK))
CHARACTER(1)        COMPZ
```

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

3 Description

F08JHF (DSTEDC) computes all the eigenvalues and, optionally, the eigenvectors of a real symmetric tridiagonal matrix T. That is, the routine computes the spectral factorization of T given by

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

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

$$Tz_i = \lambda_i z_i, \qquad i = 1, 2, \ldots, n.$$

The routine may also be used to compute all the eigenvalues and vectors of a real full, or banded, symmetric matrix A which has been reduced to tridiagonal form T as

$$A = QTQ^{\mathrm{T}},$$

where Q is orthogonal. The spectral factorization of A is then given by

$$A = (QZ)\Lambda(QZ)^{\mathrm{T}}.$$

In this case Q must be formed explicitly and passed to F08JHF (DSTEDC) in the array Z, and the routine called with COMPZ = 'V'. Routines which may be called to form T and Q are

full matrix	F08FEF (DSYTRD) and F08FFF (DORGTR)
full matrix, packed storage	F08GEF (DSPTRD) and F08GFF (DOPGTR)
band matrix	F08HEF (DSBTRD), with VECT = 'V'

When only eigenvalues are required then this routine calls F08JFF (DSTERF) to compute the eigenvalues of the tridiagonal matrix T, but when eigenvectors of T are also required and the matrix is not too small, then a divide and conquer method is used, which can be much faster than F08JEF (DSTEQR), although more storage is required.

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

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

5 Parameters

1: COMPZ – CHARACTER(1) *Input*

On entry: indicates whether the eigenvectors are to be computed.

COMPZ = 'N'
Only the eigenvalues are computed (and the array Z is not referenced).

COMPZ = 'I'
The eigenvalues and eigenvectors of T are computed (and the array Z is initialized by the routine).

COMPZ = 'V'
The eigenvalues and eigenvectors of A are computed (and the array Z must contain the matrix Q on entry).

Constraint: COMPZ = 'N', 'V' or 'I'.

2: N – INTEGER *Input*

On entry: n, the order of the symmetric tridiagonal matrix T.

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

3: 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 diagonal elements of the tridiagonal matrix.

On exit: if INFO = 0, the eigenvalues in ascending order.

4: 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 subdiagonal elements of the tridiagonal matrix.

On exit: E is overwritten.

5: Z(LDZ,$*$) – REAL (KIND=nag_wp) array *Input/Output*

Note: the second dimension of the array Z must be at least $\max(1, \mathrm{N})$ if COMPZ = 'V' or 'I', and at least 1 otherwise.

On entry: if COMPZ = 'V', Z must contain the orthogonal matrix Q used in the reduction to tridiagonal form.

On exit: if COMPZ = 'V', Z contains the orthonormal eigenvectors of the original symmetric matrix A, and if COMPZ = 'I', Z contains the orthonormal eigenvectors of the symmetric tridiagonal matrix T.

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

6: LDZ – INTEGER *Input*

On entry: the first dimension of the array Z as declared in the (sub)program from which F08JHF (DSTEDC) is called.

Constraints:

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

7: WORK(max(1, LWORK)) – REAL (KIND=nag_wp) array *Workspace*

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

8: LWORK – INTEGER *Input*

On entry: the dimension of the array WORK as declared in the (sub)program from which F08JHF (DSTEDC) 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.

Constraints: if LWORK $\neq -1$,

if COMPZ = 'N' or N ≤ 1, LWORK must be at least 1;
if COMPZ = 'V' and N > 1,
LWORK must be at least $\left(1 + 3 \times \mathrm{N} + 2 \times \mathrm{N} \times \mathrm{lg}(\mathrm{N}) + 4 \times \mathrm{N}^2\right)$, where $\mathrm{lg}(\mathrm{N}) =$ smallest integer k such that $2^k \geq \mathrm{N}$;
if COMPZ = 'I' and N > 1, LWORK must be at least $\left(1 + 4 \times \mathrm{N} + \mathrm{N}^2\right)$.

Note: that for COMPZ = 'I' or 'V' then if N is less than or equal to the minimum divide size, usually 25, then LWORK need only be $\max(1, 2 \times (\mathrm{N} - 1))$.

9: IWORK(max(1, LIWORK)) – INTEGER array *Workspace*

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

10: LIWORK – INTEGER *Input*

On entry: the dimension of the array IWORK as declared in the (sub)program from which F08JHF (DSTEDC) 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.

Constraints: if LIWORK $\neq -1$,

if COMPZ = 'N' or N ≤ 1, LIWORK must be at least 1;
if COMPZ = 'V' and N > 1, LIWORK must be at least $(6 + 6 \times \mathrm{N} + 5 \times \mathrm{N} \times \mathrm{lg}(\mathrm{N}))$;
if COMPZ = 'I' and N > 1, LIWORK must be at least $(3 + 5 \times \mathrm{N})$.

Note: that for COMPZ = 'I' or 'V', then if N is less than or equal to the minimum divide size, usually 25, then LIWORK need only be 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 $= -i$, argument i had an illegal value. An explanatory message is output, and execution of the program is terminated.

INFO > 0

The algorithm failed to compute an eigenvalue while working on the submatrix lying in rows and columns INFO$/$(N $+ 1$) through INFO mod (N $+ 1$).

7 Accuracy

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

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

and ϵ is the ***machine precision***.

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

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

where $c(n)$ is a modestly increasing function of n.

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:

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

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

See Section 4.7 of Anderson *et al.* (1999) for further details. See also F08FLF (DDISNA).

8 Further Comments

If only eigenvalues are required, the total number of floating point operations is approximately proportional to n^2. When eigenvectors are required the number of operations is bounded above by approximately the same number of operations as F08JEF (DSTEQR), but for large matrices F08JHF (DSTEDC) is usually much faster.

The complex analogue of this routine is F08JVF (ZSTEDC).

9 Example

This example finds all the eigenvalues and eigenvectors of the symmetric band matrix

$$A = \begin{pmatrix} 4.99 & 0.04 & 0.22 & 0 \\ 0.04 & 1.05 & -0.79 & 1.04 \\ 0.22 & -0.79 & -2.31 & -1.30 \\ 0 & 1.04 & -1.30 & -0.43 \end{pmatrix}.$$

A is first reduced to tridiagonal form by a call to F08HEF (DSBTRD).

9.1 Program Text

```
    Program f08jhfe

!     F08JHF Example Program Text

!     Mark 24 Release. NAG Copyright 2012.

!     .. Use Statements ..
      Use nag_library, Only: dsbtrd, dstedc, nag_wp, x04caf
!     .. Implicit None Statement ..
      Implicit None
!     .. Parameters ..
      Integer, Parameter                :: nin = 5, nout = 6
      Character (1), Parameter          :: uplo = 'U'
```

```
!     .. Local Scalars ..
      Integer                          :: i, ifail, info, j, kd, ldab, ldz,    &
                                          lgn, liwork, lwork, n
!     .. Local Arrays ..
      Real (Kind=nag_wp), Allocatable  :: ab(:,:), d(:), e(:), work(:), z(:,:)
      Real (Kind=nag_wp)               :: rdum(1)
      Integer                          :: idum(1)
      Integer, Allocatable             :: iwork(:)
!     .. Intrinsic Procedures ..
      Intrinsic                        :: ceiling, log, max, min, nint, real
!     .. Executable Statements ..
      Write (nout,*) 'F08JHF Example Program Results'
      Write (nout,*)
!     Skip heading in data file
      Read (nin,*)
      Read (nin,*) n, kd
      ldab = kd + 1
      ldz = n
      lgn = ceiling(log(real(n,kind=nag_wp))/log(2.0E0_nag_wp))
      Allocate (ab(ldab,n),d(n),e(n-1),z(ldz,n))

!     Use routine workspace query to get optimal workspace.
      lwork = -1
      liwork = -1
!     The NAG name equivalent of dstedc is f08jhf
      Call dstedc('V',n,d,e,z,ldz,rdum,lwork,idum,liwork,info)

!     Make sure that there is enough workspace.
      lwork = max(1+3*n+2*n*lgn+4*n*n,nint(rdum(1)))
      liwork = max(6+6*n+5*n*lgn,idum(1))
      Allocate (work(lwork),iwork(liwork))

!     Read the upper or lower triangular part of the band matrix A
!     from data file

      If (uplo=='U') Then
        Do i = 1, n
          Read (nin,*)(ab(kd+1+i-j,j),j=i,min(n,i+kd))
        End Do
      Else If (uplo=='L') Then
        Do i = 1, n
          Read (nin,*)(ab(1+i-j,j),j=max(1,i-kd),i)
        End Do
      End If

!     Reduce A to tridiagonal form T = (Z**T)*A*Z, and form Z
!     The NAG name equivalent of dsbtrd is f08hef
      Call dsbtrd('V',uplo,n,kd,ab,ldab,d,e,z,ldz,work,info)

!     Calculate all the eigenvalues and eigenvectors of A,
!     from T and Z
!     The NAG name equivalent of dstedc is f08jhf
      Call dstedc('V',n,d,e,z,ldz,work,lwork,iwork,liwork,info)

      If (info==0) Then

!       Print eigenvalues and eigenvectors

        Write (nout,*) 'Eigenvalues'
        Write (nout,99999) d(1:n)

        Write (nout,*)
        Flush (nout)

!       Standardize the eigenvectors so that first elements are non-negative.
        Do i = 1, n
          If (z(1,i)<0.0_nag_wp) z(1:n,i) = -z(1:n,i)
        End Do

!       ifail: behaviour on error exit
!              =0 for hard exit, =1 for quiet-soft, =-1 for noisy-soft
```

```
        ifail = 0
        Call x04caf('General',' ',n,n,z,ldz,'Eigenvectors',ifail)

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

99999 Format ((3X,8F8.4))
99998 Format (1X,A,I10)
    End Program f08jhfe
```

9.2 Program Data

```
F08JHF Example Program Data

  4      2                   :Values of N and KD

  4.99   0.04   0.22
         1.05  -0.79   1.04
               -2.31  -1.30
                      -0.43  :End of matrix A
```

9.3 Program Results

```
 F08JHF Example Program Results

 Eigenvalues
    -2.9943 -0.7000  1.9974  4.9969

 Eigenvectors
          1       2       3       4
 1   0.0251  0.0162  0.0113  0.9995
 2  -0.0656 -0.5859  0.8077  0.0020
 3  -0.9002 -0.3135 -0.3006  0.0311
 4  -0.4298  0.7471  0.5070 -0.0071
```