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,
     1                  INFO)
      INTEGER           N, LDZ, LWORK, IWORK(max(1,LIWORK)), LIWORK, INFO
      double precision  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

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

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

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

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, Philadelphiahttp://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($*$) – ***double precision*** 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 $\mathrm{INFO} = 0$, the eigenvalues in ascending order.

4: 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 subdiagonal elements of the tridiagonal matrix.

On exit: E is overwritten.

5: Z(LDZ,$*$) – ***double precision*** array *Input/Output*

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

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

On exit: if $\mathrm{INFO} = 0$, then if COMPZ = 'V', Z contains the orthonormal eigenvectors of the original symmetric matrix, and if COMPZ = 'I', Z contains the orthonormal eigenvectors of the symmetric tridiagonal matrix.

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', $LDZ \geq \max(1, N)$;
otherwise $LDZ \geq 1$.

7: WORK(max(1, LWORK)) – ***double precision*** 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 $N \leq 1$, LWORK must be at least 1;
if COMPZ = 'V' and $N > 1$,
LWORK must be at least $\left(1 + 3 \times N + 2 \times N \times \lg(N) + 3 \times N^2\right)$, where $\lg(N) =$ smallest integer k such that $2^k \geq N$;
if COMPZ = 'I' and $N > 1$, LWORK must be at least $\left(1 + 4 \times N + N^2\right)$.

.

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 $\max(1, 2 \times (N - 1))$.

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

.

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 $\mathrm{INFO}/(\mathrm{N}+1)$ through $\mathrm{mod}(\mathrm{INFO}, \mathrm{N}+1)$.

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

```
*     F08JHF Example Program Text
*     Mark 21 Release. NAG Copyright 2004.
*     .. Parameters ..
      INTEGER          NIN, NOUT
      PARAMETER        (NIN=5,NOUT=6)
      INTEGER          LGNMAX, NMAX, KDMAX
      PARAMETER        (LGNMAX=5,NMAX=2**LGNMAX,KDMAX=8)
      INTEGER          LDAB, LDZ, LIWORK, LWORK
      PARAMETER        (LDAB=KDMAX+1,LDZ=NMAX,
     +                 LIWORK=6+6*NMAX+5*NMAX*LGNMAX,
     +                 LWORK=1+3*NMAX+2*NMAX*LGNMAX+3*NMAX*NMAX)
      CHARACTER        UPLO
      PARAMETER        (UPLO='U')
*     .. Local Scalars ..
      INTEGER          I, IFAIL, INFO, J, KD, LIWOPT, LWOPT, N
*     .. Local Arrays ..
      DOUBLE PRECISION AB(LDAB,NMAX), D(NMAX), E(NMAX-1), WORK(LWORK),
     +                 Z(LDZ,NMAX)
      INTEGER          IWORK(LIWORK)
*     .. External Subroutines ..
      EXTERNAL         DSBTRD, DSTEDC, X04CAF
*     .. Intrinsic Functions ..
      INTRINSIC        INT, MAX, MIN
*     .. Executable Statements ..
      WRITE (NOUT,*) 'F08JHF Example Program Results'
      WRITE (NOUT,*)
*     Skip heading in data file
      READ (NIN,*)
      READ (NIN,*) N, KD
      IF (N.LE.NMAX .AND. KD.LE.KDMAX) THEN
*
*        Read the upper or lower triangular part of the band matrix A
*        from data file
*
         IF (UPLO.EQ.'U') THEN
            DO 20 I = 1, N
               READ (NIN,*) (AB(KD+1+I-J,J),J=I,MIN(N,I+KD))
   20       CONTINUE
         ELSE IF (UPLO.EQ.'L') THEN
            DO 40 I = 1, N
               READ (NIN,*) (AB(1+I-J,J),J=MAX(1,I-KD),I)
   40       CONTINUE
         END IF
*
*        Reduce A to tridiagonal form T = (Z**T)*A*Z, and form Z
*
         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
*
         CALL DSTEDC('V',N,D,E,Z,LDZ,WORK,LWORK,IWORK,LIWORK,INFO)
         LWOPT = INT(WORK(1))
         LIWOPT = IWORK(1)
*
         IF (INFO.EQ.0) THEN
*
*           Print eigenvalues and eigenvectors
*
            WRITE (NOUT,*) 'Eigenvalues'
            WRITE (NOUT,99999) (D(I),I=1,N)
*
            WRITE (NOUT,*)
            IFAIL = 0
            CALL X04CAF('General',' ',N,N,Z,LDZ,'Eigenvectors',IFAIL)
*
         ELSE
            WRITE (NOUT,99998) 'Failure in DSTEDC. 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 KDMAX too small'
      END IF
*
99999 FORMAT ((3X,8F8.4))
99998 FORMAT (1X,A,I10)
99997 FORMAT ((1X,A,I5))
      END
```

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
```