NAG Fortran Library Routine Document

F08JCF (SSTEVD/DSTEVD)

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.

Warning. The specification of the parameters LWORK and LIWORK changed at Mark 20 in the case where JOB = 'V' and N > 1: the minimum dimension of the array WORK has been reduced whereas the minimum dimension of the array IWORK has been increased.

1 Purpose

F08JCF (SSTEVD/DSTEVD) computes all the eigenvalues, and optionally all the eigenvectors, of a real symmetric tridiagonal matrix. If the eigenvectors are requested, then it uses a divide and conquer algorithm to compute eigenvalues and eigenvectors. However, if only eigenvalues are required, then it uses the Pal–Walker–Kahan variant of the QL or QR algorithm.

2 Specification

```
 SUBROUTINE F08JCF(JOB, N, D, E, Z, LDZ, WORK, LWORK, IWORK, LIWORK,
1                  INFO)
 ENTRY      sstevd (JOB, N, D, E, Z, LDZ, WORK, LWORK, IWORK, LIWORK,
1                  INFO)
 INTEGER           N, LDZ, LWORK, IWORK(*), LIWORK, INFO
 real              D(*), E(*), Z(LDZ,*), WORK(*)
 CHARACTER*1       JOB
```

The ENTRY statement enables the routine to be called by its LAPACK name.

3 Description

This routine computes all the eigenvalues, and optionally all the eigenvectors, of a real symmetric tridiagonal matrix T. In other words, it can compute the spectral factorization of T as

$$T = Z\Lambda Z^T,$$

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

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

4 References

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

5 Parameters

1: JOB – CHARACTER*1 *Input*

On entry: indicates whether eigenvectors are computed as follows:

if JOB = 'N', only eigenvalues are computed;

if JOB = 'V', eigenvalues and eigenvectors are computed.

Constraint: JOB = 'N' or 'V'.

2: N – INTEGER *Input*

On entry: n, the order of the matrix A.

Constraint: N $\geq$ 0.

3: D(*) – ***real*** array *Input/Output*

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

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

On exit: the eigenvalues of the matrix T in ascending order.

4: E(*) – ***real*** array *Input/Output*

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

On entry: the $n - 1$ off-diagonal elements of the tridiagonal matrix T. The nth element of this array is used as workspace.

On exit: the array is overwritten with intermediate results.

5: Z(LDZ,*) – ***real*** array *Output*

Note: the second dimension of the array Z must be at least max(1, N) if JOB = 'V', and at least 1 if JOB = 'N'.

On exit: if JOB = 'V', Z is overwritten by the orthogonal matrix Z which contains the eigenvectors of T.

If JOB = '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 F08JCF (SSTEVD/DSTEVD) is called.

Constraints:

LDZ ≥ max(1, N) if JOB = 'V',
LDZ ≥ 1 if JOB = 'N'.

7: WORK(*) – ***real*** array *Workspace*

Note: the dimension of the array WORK must be at least max(1, LWORK).

On exit: if INFO = 0, WORK(1) contains the required minimal size of LWORK.

8: LWORK – INTEGER *Input*

On entry: the dimension of the array WORK as declared in the (sub)program from which F08JCF (SSTEVD/DSTEVD) is called, unless LWORK = −1, in which case a workspace query is assumed and the routine only calculates the minimum dimension of WORK.

Constraints:

if JOB = 'N' or N ≤ 1, LWORK ≥ 1 or LWORK = −1,
if JOB = 'V' and N > 1, LWORK ≥ $\mathrm{N}^2 + 4 \times \mathrm{N} + 1$ or LWORK = −1.

9: 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) contains the required minimal size of LIWORK.

10: LIWORK – INTEGER *Input*

On entry: the dimension of the array IWORK as declared in the (sub)program from which F08JCF (SSTEVD/DSTEVD) is called, unless LIWORK = −1, in which case a workspace query is assumed and the routine only calculates the minimum dimension of IWORK.

Constraints:

if JOB = 'N' or N ≤ 1, LIWORK ≥ 1 or LIWORK = −1,
if JOB = 'V' and N > 1, LIWORK ≥ 5 × N + 3 or LIWORK = −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 $= -i$, the ith parameter had an illegal value. An explanatory message is output, and execution of the program is terminated.

INFO > 0

If INFO $= i$, the algorithm failed to converge; i indicates the number of elements of an intermediate tridiagonal form which did not converge to zero.

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

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

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:

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

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

8 Further Comments

There is no complex analogue of this routine.

9 Example

To compute all the eigenvalues and eigenvectors of the symmetric tridiagonal matrix T, where

$$T = \begin{pmatrix} 1.0 & 1.0 & 0.0 & 0.0 \\ 1.0 & 4.0 & 1.0 & 0.0 \\ 0.0 & 1.0 & 9.0 & 1.0 \\ 0.0 & 0.0 & 1.0 & 16.0 \end{pmatrix}.$$

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.

```
*     F08JCF Example Program Text.
*     Mark 20 Revised. NAG Copyright 2001.
*     .. Parameters ..
      INTEGER          NIN, NOUT
      PARAMETER        (NIN=5,NOUT=6)
      INTEGER          NMAX, LDZ
      PARAMETER        (NMAX=8,LDZ=NMAX)
      INTEGER          LWORK, LIWORK
      PARAMETER        (LWORK=NMAX*NMAX+4*NMAX+1,LIWORK=5*NMAX+3)
*     .. Local Scalars ..
      INTEGER          I, IFAIL, INFO, N
      CHARACTER        JOB
*     .. Local Arrays ..
      real             D(NMAX), E(NMAX), WORK(LWORK), Z(LDZ,NMAX)
      INTEGER          IWORK(LIWORK)
*     .. External Subroutines ..
      EXTERNAL         sstevd, X04CAF
*     .. Executable Statements ..
      WRITE (NOUT,*) 'F08JCF Example Program Results'
*     Skip heading in data file
      READ (NIN,*)
      READ (NIN,*) N
      IF (N.LE.NMAX) THEN
*
*        Read T from data file
*
         READ (NIN,*) (D(I),I=1,N)
         READ (NIN,*) (E(I),I=1,N-1)
*
         READ (NIN,*) JOB
*
*        Calculate all the eigenvalues and eigenvectors of T
*
         CALL sstevd(JOB,N,D,E,Z,LDZ,WORK,LWORK,IWORK,LIWORK,INFO)
*
         WRITE (NOUT,*)
         IF (INFO.GT.0) THEN
            WRITE (NOUT,*) 'Failure to converge.'
         ELSE
*
*           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)
*
         END IF
      END IF
      STOP
*
99999 FORMAT (3X,(8F8.4))
      END
```

9.2 Program Data

```
F08JCF Example Program Data
  4                         :Value of N
  1.0  4.0  9.0  16.0
  1.0  2.0  3.0             :End of T
  'V'                       :Value of JOB
```

9.3 Program Results

```
 F08JCF Example Program Results

 Eigenvalues
     0.6476  3.5470  8.6578 17.1477

 Eigenvectors
          1       2       3       4
 1   0.9396  0.3388  0.0494  0.0034
 2  -0.3311  0.8628  0.3781  0.0545
 3   0.0853 -0.3648  0.8558  0.3568
 4  -0.0167  0.0879 -0.3497  0.9326
```