F08JGF (SPTEQR/DPTEQR) – NAG Fortran Library Routine Document

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

F08JGF (SPTEQR/DPTEQR) computes all the eigenvalues, and optionally all the eigenvectors, of a real symmetric positive-definite tridiagonal matrix, or of a real symmetric positive-definite matrix which has been reduced to tridiagonal form.

2   Specification

```
SUBROUTINE F08JGF(COMPZ, N, D, E, Z, LDZ, WORK, INFO)
ENTRY        spteqr(COMPZ, N, D, E, Z, LDZ, WORK, INFO)
INTEGER          N, LDZ, INFO
real             D(*), E(*), Z(LDZ,*), WORK("*")
CHARACTER*1      COMPZ
```

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 positive-definite 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

$$Tz_i = \lambda_i z_i \text{ for } i = 1, 2, \ldots, n.$$

The routine may also be used to compute all the eigenvalues and eigenvectors of a real symmetric positive-definite matrix A which has been reduced to tridiagonal form T:

$$\begin{aligned} A \quad &= QTQ^T, \text{ where } Q \text{ is orthogonal} \\ &= (QZ)\Lambda(QZ)^T. \end{aligned}$$

In this case, the matrix Q must be formed explicitly and passed to F08JGF, which is called with COMPZ = 'V'. The routines which must be called to perform the reduction to tridiagonal form and form Q are:

full matrix	F08FEF (SSYTRD/DSYTRD) + F08FFF (SORGTR/DORGTR)
full matrix, packed storage	F08GEF (SSPTRD/DSPTRD) + F08GFF (SOPGTR/DOPGTR)
band matrix	F08HEF (SSBTRD/DSBTRD) with VECT = 'V'.

The routine first factorizes T as LDL^T where L is unit lower bidiagonal and D is diagonal. It forms the bidiagonal matrix $B = LD^{\frac{1}{2}}$, and then calls F08MEF (SBDSQR/DBDSQR) to compute the singular values of B which are the same as the eigenvalues of T. The method used by the routine allows high relative accuracy to be achieved in the small eigenvalues of T. The eigenvectors are normalized so that $\| z_i \|_2 = 1$, but are determined only to within a factor ± 1.

4   References

[1]   Barlow J and Demmel J W (1990) Computing accurate eigensystems of scaled diagonally dominant matrices *SIAM J. Numer. Anal.* **27** 762–791

5 Parameters

1: COMPZ — CHARACTER*1 *Input*

On entry: indicates whether the eigenvectors are to be computed as follows:

if COMPZ = 'N', then only the eigenvalues are computed (and the array Z is not referenced);

if COMPZ = 'I', then the eigenvalues and eigenvectors of T are computed (and the array Z is initialized by the routine);

if COMPZ = 'V', then 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 matrix T.

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 diagonal elements of the tridiagonal matrix T.

On exit: the n eigenvalues in descending order, unlessINFO > 0, in which case the array is overwritten.

4: E(∗) — ***real*** array *Input/Output*

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

On entry: the off-diagonal elements of the tridiagonal matrix T.

On exit: the array is overwritten.

5: Z(LDZ,∗) — ***real*** array *Input/Output*

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

On entry: if COMPZ = 'V', Z must contain the orthogonal matrix Q from the reduction to tridiagonal form. If COMPZ = 'I', Z need not be set.

On exit: if COMPZ = 'I' or 'V', the n required orthonormal eigenvectors stored by columns; the ith column corresponds to the ith eigenvalue, where $i = 1, 2, \ldots, n$, unless INFO > 0.

Z is not referenced if COMPZ = 'N'.

6: LDZ — INTEGER *Input*

On entry: the first dimension of the array Z as declared in the (sub)program from which F08JGF (SPTEQR/DPTEQR) is called.

Constraints:

LDZ $\geq$ 1 if COMPZ = 'N',

LDZ $\geq$ max(1,N) if COMPZ = 'V' or 'I'.

7: WORK(∗) — ***real*** array *Workspace*

Note: the dimension of the array WORK must be at least max(1,4∗(N−1)) if COMPZ = 'V' or 'I', and at least 1 if COMPZ = 'N'.

WORK is not referenced if COMPZ = 'N'.

8: 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 $= -i$, 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 leading minor of order i is not positive-definite and the Cholesky factorization of T could not be completed. Hence T itself is not positive-definite.

> If INFO $= N + i$, the algorithm to compute the singular values of the Cholesky factor B failed to converge; i off-diagonal elements did not converge to zero.

7   Accuracy

The eigenvalues and eigenvectors of T are computed to high relative accuracy which means that if they vary widely in magnitude, then any small eigenvalues (and corresponding eigenvectors) will be computed more accurately than, for example, with the standard QR method. However, the reduction to tridiagonal form (prior to calling the routine) may exclude the possibility of obtaining high relative accuracy in the small eigenvalues of the original matrix if its eigenvalues vary widely in magnitude.

To be more precise, let H be the tridiagonal matrix defined by $H = DTD$, where D is diagonal with $d_{ii} = t_{ii}^{-\frac{1}{2}}$, and $h_{ii} = 1$ for all i. If λ_i is an exact eigenvalue of T and $\tilde{\lambda}_i$ is the corresponding computed value, then

$$|\tilde{\lambda}_i - \lambda_i| \le c(n)\epsilon\kappa_2(H)\lambda_i$$

where $c(n)$ is a modestly increasing function of n, ϵ is the ***machine precision***, and $\kappa_2(H)$ is the condition number of H with respect to inversion defined by: $\kappa_2(H) = \| H \| \cdot \| H^{-1} \|$.

If z_i is the corresponding exact eigenvector of T, and $\tilde{z}_i$ is the corresponding computed eigenvector, then the angle $\theta(\tilde{z}_i, z_i)$ between them is bounded as follows:

$$\theta(\tilde{z}_i, z_i) \le \frac{c(n)\epsilon\kappa_2(H)}{relgap_i}$$

where $relgap_i$ is the relative gap between λ_i and the other eigenvalues, defined by

$$relgap_i = \min_{i \ne j} \frac{|\lambda_i - \lambda_j|}{(\lambda_i + \lambda_j)}.$$

8   Further Comments

The total number of floating-point operations is typically about $30n^2$ if COMPZ = 'N' and about $6n^3$ if COMPZ = 'V' or 'I', but depends on how rapidly the algorithm converges. When COMPZ = 'N', the operations are all performed in scalar mode; the additional operations to compute the eigenvectors when COMPZ = 'V' or 'I' can be vectorized and on some machines may be performed much faster.

The complex analogue of this routine is F08JUF (CPTEQR/ZPTEQR).

9   Example

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

$$T = \begin{pmatrix} 4.16 & 3.17 & 0.00 & 0.00 \\ 3.17 & 5.25 & -0.97 & 0.00 \\ 0.00 & -0.97 & 1.09 & 0.55 \\ 0.00 & 0.00 & 0.55 & 0.62 \end{pmatrix}.$$

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.

```
*       F08JGF Example Program Text
*       Mark 16 Release. NAG Copyright 1992.
*       .. Parameters ..
        INTEGER           NIN, NOUT
        PARAMETER         (NIN=5,NOUT=6)
        INTEGER           NMAX, LDZ
        PARAMETER         (NMAX=8,LDZ=NMAX)
*       .. Local Scalars ..
        INTEGER           I, IFAIL, INFO, N
*       .. Local Arrays ..
        real              D(NMAX), E(NMAX-1), WORK(4*NMAX-4), Z(LDZ,NMAX)
*       .. External Subroutines ..
        EXTERNAL          spteqr, X04CAF
*       .. Executable Statements ..
        WRITE (NOUT,*) 'F08JGF 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)
*
*          Calculate all the eigenvalues and eigenvectors of T
*
           CALL spteqr('I',N,D,E,Z,LDZ,WORK,INFO)
*
           WRITE (NOUT,*)
           IF (INFO.GT.0 .AND. INFO.LE.N) THEN
              WRITE (NOUT,*) 'T is not positive-definite.'
           ELSE IF (INFO.GT.N) 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

```
F08JGF Example Program Data
  4                           :Value of N
  4.16   5.25   1.09   0.62
  3.17  -0.97   0.55         :End of matrix T
```

9.3   Program Results

```
F08JGF Example Program Results

Eigenvalues
    8.0023  1.9926  1.0014  0.1237

Eigenvectors
          1        2        3        4
1    0.6326   0.6245  -0.4191   0.1847
2    0.7668  -0.4270   0.4176  -0.2352
3   -0.1082   0.6071   0.4594  -0.6393
4   -0.0081   0.2432   0.6625   0.7084
```