NAG Fortran Library Routine Document

F08JUF (CPTEQR/ZPTEQR)

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 parameter WORK changed at Mark 20: the length of WORK needs to be increased.

1 Purpose

F08JUF (CPTEQR/ZPTEQR) computes all the eigenvalues, and optionally all the eigenvectors, of a complex Hermitian positive-definite matrix which has been reduced to tridiagonal form.

2 Specification

```
SUBROUTINE F08JUF(COMPZ, N, D, E, Z, LDZ, WORK, INFO)
ENTRY      cpteqr (COMPZ, N, D, E, Z, LDZ, WORK, INFO)
INTEGER            N, LDZ, INFO
real               D(*), E(*), WORK(*)
complex            Z(LDZ,*)
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

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

The routine stores the real orthogonal matrix Z in a ***complex*** array, so that it may be used to compute all the eigenvalues and eigenvectors of a complex Hermitian positive-definite matrix A which has been reduced to tridiagonal form T:

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

In this case, the matrix Q must be formed explicitly and passed to F08JUF, 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	F08FSF (CHETRD/ZHETRD) + F08FTF (CUNGTR/ZUNGTR)
full matrix, packed storage	F08GSF (CHPTRD/ZHPTRD) + F08GTF (CUPGTR/ZUPGTR)
band matrix	F08HSF (CHBTRD/ZHBTRD) with VECT = 'V'.

The routine first factorizes T as LDL^H where L is unit lower bidiagonal and D is diagonal. It forms the bidiagonal matrix $B = LD^{\frac{1}{2}}$, and then calls F08MSF (CBDSQR/ZBDSQR) 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 complex factor of absolute value 1.

4 References

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', only the eigenvalues are computed (and the array Z is not referenced);

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

if 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 matrix T.

Constraint: N ≥ 0.

3: D(∗) – ***real*** 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 T.

On exit: the n eigenvalues in descending order, unless INFO > 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 $\max(1, \mathrm{N} - 1)$.

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

On exit: the array is overwritten.

5: Z(LDZ,*) – ***complex*** 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 if COMPZ = 'N'.

On entry: if COMPZ = 'V', Z must contain the unitary 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 F08JUF (CPTEQR/ZPTEQR) is called.

Constraints:

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

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

Note: the dimension of the array WORK must be at least $\max(1, 4 * \mathrm{N})$.

8: 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 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 $= \mathrm{N} + i$, 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

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

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:

$$\theta(\tilde{z}_i, z_i) \leq \frac{c(n)\epsilon\kappa_2(H)}{\mathit{relgap}_i}$$

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

$$\mathit{relgap}_i = \min_{i \neq j} \frac{|\lambda_i - \lambda_j|}{(\lambda_i + \lambda_j)}.$$

8 Further Comments

The total number of real floating-point operations is typically about $30n^2$ if COMPZ = 'N' and about $12n^3$ 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 real analogue of this routine is F08JGF (SPTEQR/DPTEQR).

9 Example

To compute all the eigenvalues and eigenvectors of the complex Hermitian positive-definite matrix A, where

$$A = \begin{pmatrix} 6.02 + 0.00i & -0.45 + 0.25i & -1.30 + 1.74i & 1.45 - 0.66i \\ -0.45 - 0.25i & 2.91 + 0.00i & 0.05 + 1.56i & -1.04 + 1.27i \\ -1.30 - 1.74i & 0.05 - 1.56i & 3.29 + 0.00i & 0.14 + 1.70i \\ 1.45 + 0.66i & -1.04 - 1.27i & 0.14 - 1.70i & 4.18 + 0.00i \end{pmatrix}.$$

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.

```
*     F08JUF Example Program Text
*     Mark 20 Revised. NAG Copyright 2001.
*     .. Parameters ..
      INTEGER          NIN, NOUT
      PARAMETER        (NIN=5,NOUT=6)
      INTEGER          NMAX, LDA, LWORK, LDZ
      PARAMETER        (NMAX=8,LDA=NMAX,LWORK=64*NMAX,LDZ=NMAX)
*     .. Local Scalars ..
      INTEGER          I, IFAIL, INFO, J, N
      CHARACTER        UPLO
*     .. Local Arrays ..
      complex          A(LDA,NMAX), TAU(NMAX), WORK(LWORK), Z(LDZ,NMAX)
      real             D(NMAX), E(NMAX), RWORK(4*NMAX)
      CHARACTER        CLABS(1), RLABS(1)
*     .. External Subroutines ..
      EXTERNAL         F06TFF, X04DBF, chetrd, cpteqr, cungtr
*     .. Executable Statements ..
      WRITE (NOUT,*) 'F08JUF Example Program Results'
*     Skip heading in data file
      READ (NIN,*)
      READ (NIN,*) N
      IF (N.LE.NMAX) THEN
*
*        Read A from data file
*
         READ (NIN,*) UPLO
         IF (UPLO.EQ.'U') THEN
            READ (NIN,*) ((A(I,J),J=I,N),I=1,N)
         ELSE IF (UPLO.EQ.'L') THEN
            READ (NIN,*) ((A(I,J),J=1,I),I=1,N)
         END IF
*
*        Reduce A to tridiagonal form T = (Q**H)*A*Q
*
         CALL chetrd(UPLO,N,A,LDA,D,E,TAU,WORK,LWORK,INFO)
*
*        Copy A into Z
*
         CALL F06TFF(UPLO,N,N,A,LDA,Z,LDZ)
*
*        Form Q explicitly, storing the result in Z
*
         CALL cungtr(UPLO,N,Z,LDZ,TAU,WORK,LWORK,INFO)
*
*        Calculate all the eigenvalues and eigenvectors of A
*
         CALL cpteqr('V',N,D,E,Z,LDZ,RWORK,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 X04DBF('General',' ',N,N,Z,LDZ,'Bracketed','F7.4',
     +                  'Eigenvectors','Integer',RLABS,'Integer',CLABS,
     +                  80,0,IFAIL)
*
         END IF
       END IF
       STOP
*
99999 FORMAT (8X,4(F7.4,11X,:))
      END
```

9.2 Program Data

```
F08JUF Example Program Data
  4                                                          :Value of N
  'L'                                                        :Value of UPLO
 ( 6.02, 0.00)
 (-0.45,-0.25) ( 2.91, 0.00)
 (-1.30,-1.74) ( 0.05,-1.56) ( 3.29, 0.00)
 ( 1.45, 0.66) (-1.04,-1.27) ( 0.14,-1.70) ( 4.18, 0.00)   :End of matrix A
```

9.3 Program Results

```
 F08JUF Example Program Results

 Eigenvalues
         7.9995            5.9976            2.0003            0.4026

 Eigenvectors
                     1                 2                 3                 4
 1   ( 0.7289, 0.0000) ( 0.2001, 0.4724) (-0.2133, 0.1498) ( 0.0995,-0.3573)
 2   (-0.1651,-0.2067) (-0.2461, 0.3742) ( 0.7308, 0.0000) ( 0.2867,-0.3364)
 3   (-0.4170,-0.1413) ( 0.4476, 0.1455) (-0.3282, 0.0471) ( 0.6890, 0.0000)
 4   ( 0.1748, 0.4175) ( 0.5610, 0.0000) ( 0.5203, 0.1317) ( 0.0659, 0.4336)
```