NAG Library Routine Document

F08JGF (DPTEQR)

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

F08JGF (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)
INTEGER              N, LDZ, INFO
REAL (KIND=nag_wp) D(*), E(*), Z(LDZ,*), WORK(4*N)
CHARACTER(1)         COMPZ
```

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

3 Description

F08JGF (DPTEQR) 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

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

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 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:

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

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

F08JGF (DPTEQR) 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 (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

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.

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

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

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

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

2: N – INTEGER *Input*

On entry: n, the order of the 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 T.

On exit: the n eigenvalues in descending order, unless INFO > 0, in which case D is overwritten.

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 off-diagonal elements of the tridiagonal matrix T.

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 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 = 'V' or 'I', the n required orthonormal eigenvectors stored as columns of Z; the ith column corresponds to the ith eigenvalue, where $i = 1, 2, \ldots, n$, unless INFO > 0.

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 F08JGF (DPTEQR) is called.

Constraints:

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

7: WORK($4 \times$ N) – REAL (KIND=nag_wp) array *Workspace*

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 $= -i$, argument i 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

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

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: $\kappa_2(H) = \|H\| \cdot \left\|H^{-1}\right\|$.

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:

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

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

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

8 Parallelism and Performance

F08JGF (DPTEQR) is threaded by NAG for parallel execution in multithreaded implementations of the NAG Library.

F08JGF (DPTEQR) makes calls to BLAS and/or LAPACK routines, which may be threaded within the vendor library used by this implementation. Consult the documentation for the vendor library for further information.

Please consult the X06 Chapter Introduction for information on how to control and interrogate the OpenMP environment used within this routine. Please also consult the Users' Note for your implementation for any additional implementation-specific information.

9 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 (ZPTEQR).

10 Example

This example computes all the eigenvalues and eigenvectors of the symmetric positive definite tridiagonal matrix T, where

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

10.1 Program Text

```
    Program f08jgfe

!     F08JGF Example Program Text

!     Mark 25 Release. NAG Copyright 2014.

!     .. Use Statements ..
      Use nag_library, Only: dpteqr, nag_wp, x04caf
!     .. Implicit None Statement ..
      Implicit None
!     .. Parameters ..
      Integer, Parameter               :: nin = 5, nout = 6
!     .. Local Scalars ..
      Integer                          :: i, ifail, info, ldz, n
!     .. Local Arrays ..
      Real (Kind=nag_wp), Allocatable  :: d(:), e(:), work(:), z(:,:)
!     .. Executable Statements ..
      Write (nout,*) 'F08JGF Example Program Results'
!     Skip heading in data file
      Read (nin,*)
      Read (nin,*) n
      ldz = n
      Allocate (d(n),e(n-1),work(4*n),z(ldz,n))

!     Read T from data file

      Read (nin,*) d(1:n)
      Read (nin,*) e(1:n-1)

!     Calculate all the eigenvalues and eigenvectors of T
!     The NAG name equivalent of dpteqr is f08jgf
      Call dpteqr('I',n,d,e,z,ldz,work,info)

      Write (nout,*)
      If (info>0 .And. info<=n) Then
        Write (nout,*) 'T is not positive definite.'
      Else If (info>n) Then
        Write (nout,*) 'Failure to converge.'
      Else

!       Print eigenvalues and eigenvectors
```

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

!       Normalize the eigenvectors
        Do i = 1, n
          z(1:n,i) = z(1:n,i)/z(1,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)

      End If

99999 Format (3X,(8F8.4))
    End Program f08jgfe
```

10.2 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
```

10.3 Program Results

```
 F08JGF Example Program Results

 Eigenvalues
     8.0023  1.9926  1.0014  0.1237

 Eigenvectors
             1          2          3          4
 1      1.0000     1.0000     1.0000     1.0000
 2      1.2121    -0.6837    -0.9964    -1.2733
 3     -0.1711     0.9721    -1.0962    -3.4611
 4     -0.0127     0.3895    -1.5807     3.8354
```