NAG Library Routine Document

F08JCF (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 arguments 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 (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,        &
                   INFO)
INTEGER             N, LDZ, LWORK, IWORK(max(1,LIWORK)), LIWORK, INFO
REAL (KIND=nag_wp) D(*), E(*), Z(LDZ,*), WORK(max(1,LWORK))
CHARACTER(1)        JOB
```

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

3 Description

F08JCF (DSTEVD) 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

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

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

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

4 References

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

5 Arguments

1: JOB – CHARACTER(1) *Input*

On entry: indicates whether eigenvectors are computed.

JOB = 'N'
Only eigenvalues are computed.

JOB = 'V'
Eigenvalues and eigenvectors are computed.

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

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 n diagonal elements of the tridiagonal matrix T.

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

4: E($*$) – REAL (KIND=nag_wp) array *Input/Output*

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

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: E is overwritten with intermediate results.

5: Z(LDZ, $*$) – REAL (KIND=nag_wp) array *Output*

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

Constraints:

if JOB = 'V', $\mathrm{LDZ} \geq \max(1, \mathrm{N})$;
if JOB = 'N', $\mathrm{LDZ} \geq 1$.

7: WORK($\max(1, \mathrm{LWORK})$) – REAL (KIND=nag_wp) array *Workspace*

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 (DSTEVD) is called.

If LWORK = -1, a workspace query is assumed; the routine only calculates the minimum dimension of the WORK array, returns this value as the first entry of the WORK array, and no error message related to LWORK is issued.

Constraints:

if JOB = 'N' or $\mathrm{N} \leq 1$, $\mathrm{LWORK} \geq 1$ or LWORK = -1;
if JOB = 'V' and $\mathrm{N} > 1$, $\mathrm{LWORK} \geq \mathrm{N}^2 + 4 \times \mathrm{N} + 1$ or LWORK = -1.

9: IWORK($\max(1, \mathrm{LIWORK})$) – INTEGER array *Workspace*

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 (DSTEVD) is called.

If LIWORK $= -1$, a workspace query is assumed; the routine only calculates the minimum dimension of the IWORK array, returns this value as the first entry of the IWORK array, and no error message related to LIWORK is issued.

Constraints:

if JOB $=$ 'N' or N ≤ 1, LIWORK ≥ 1 or LIWORK $= -1$;
if JOB $=$ 'V' and N > 1, LIWORK $\geq 5 \times$ 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

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 $= i$ and JOB $=$ 'N', the algorithm failed to converge; i elements of an intermediate tridiagonal form did not converge to zero; if INFO $= i$ and JOB $=$ 'V', then the algorithm failed to compute an eigenvalue while working on the submatrix lying in rows and column $i/(\text{N}+1)$ through i mod $(\text{N}+1)$.

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.

8 Parallelism and Performance

F08JCF (DSTEVD) is threaded by NAG for parallel execution in multithreaded implementations of the NAG Library.

F08JCF (DSTEVD) 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

There is no complex analogue of this routine.

10 Example

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

$$T = \begin{pmatrix} 1.0 & 1.0 & 0.0 & 0.0 \\ 1.0 & 4.0 & 2.0 & 0.0 \\ 0.0 & 2.0 & 9.0 & 3.0 \\ 0.0 & 0.0 & 3.0 & 16.0 \end{pmatrix}.$$

10.1 Program Text

```
    Program f08jcfe

!     F08JCF Example Program Text

!     Mark 26 Release. NAG Copyright 2016.

!     .. Use Statements ..
      Use nag_library, Only: dnrm2, dstevd, nag_wp, x04caf
!     .. Implicit None Statement ..
      Implicit None
!     .. Parameters ..
      Integer, Parameter               :: nin = 5, nout = 6
!     .. Local Scalars ..
      Real (Kind=nag_wp)               :: norm
      Integer                          :: i, ifail, info, ldz, liwork, lwork,  &
                                          n
      Character (1)                    :: job
!     .. Local Arrays ..
      Real (Kind=nag_wp), Allocatable  :: d(:), e(:), work(:), z(:,:)
      Integer, Allocatable             :: iwork(:)
!     .. Executable Statements ..
      Write (nout,*) 'F08JCF Example Program Results'
!     Skip heading in data file
      Read (nin,*)
      Read (nin,*) n
      ldz = n
      liwork = 5*n + 3
      lwork = n*n + 4*n + 1
      Allocate (d(n),e(n-1),work(lwork),z(ldz,n),iwork(liwork))

!     Read T from data file

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

      Read (nin,*) job

!     Calculate all the eigenvalues and eigenvectors of T
!     The NAG name equivalent of dstevd is f08jcf
      Call dstevd(job,n,d,e,z,ldz,work,lwork,iwork,liwork,info)

      Write (nout,*)
      If (info>0) 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
!         The NAG name equivalent of dnrm2 is f06ejf
          norm = dnrm2(n,z(1,i),1)
          If (z(1,i)<0.0_nag_wp) Then
            norm = -norm
          End If
          z(1:n,i) = z(1:n,i)/norm
        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 f08jcfe
```

10.2 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
```

10.3 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
```