NAG Library Routine Document

F08JLF (DSTEGR)

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

F08JLF (DSTEGR) computes all the eigenvalues and, optionally, all the eigenvectors of a real n by n symmetric tridiagonal matrix.

2 Specification

```
SUBROUTINE F08JLF (JOBZ, RANGE, N, D, E, VL, VU, IL, IU, ABSTOL, M, W, Z,       &
                   LDZ, ISUPPZ, WORK, LWORK, IWORK, LIWORK, INFO)

INTEGER            N, IL, IU, M, LDZ, ISUPPZ(*), LWORK,                         &
                   IWORK(max(1,LIWORK)), LIWORK, INFO
REAL (KIND=nag_wp) D(*), E(*), VL, VU, ABSTOL, W(*), Z(LDZ,*),                  &
                   WORK(max(1,LWORK))
CHARACTER(1)       JOBZ, RANGE
```

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

3 Description

F08JLF (DSTEGR) computes all the eigenvalues and, optionally, the eigenvectors, of a real symmetric tridiagonal matrix T. That is, the routine computes the spectral factorization of T given by

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

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

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

The routine may also be used to compute all the eigenvalues and eigenvectors of a real symmetric 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 explicitly applied to the output matrix Z. The routines which must be called to perform the reduction to tridiagonal form and apply Q are:

full matrix	F08FEF (DSYTRD) and F08FGF (DORMTR)
full matrix, packed storage	F08GEF (DSPTRD) and F08GGF (DOPMTR)
band matrix	F08HEF (DSBTRD) with VECT = 'V' and F06YAF (DGEMM).

This routine uses the dqds and the Relatively Robust Representation algorithms to compute the eigenvalues and eigenvectors respectively; see for example Parlett and Dhillon (2000) and Dhillon and Parlett (2004) for further details. F08JLF (DSTEGR) can usually compute all the eigenvalues and eigenvectors in $O(n^2)$ floating point operations and so, for large matrices, is often considerably faster than the other symmetric tridiagonal routines in this chapter when all the eigenvectors are required, particularly so compared to those routines that are based on the QR algorithm.

4 References

Anderson E, Bai Z, Bischof C, Blackford S, Demmel J, Dongarra J J, Du Croz J J, Greenbaum A, Hammarling S, McKenney A and Sorensen D (1999) *LAPACK Users' Guide* (3rd Edition) SIAM, Philadelphia http://www.netlib.org/lapack/lug

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

Dhillon I S and Parlett B N (2004) Orthogonal eigenvectors and relative gaps. *SIAM J. Appl. Math.* **25** 858–899

Parlett B N and Dhillon I S (2000) Relatively robust representations of symmetric tridiagonals *Linear Algebra Appl.* **309** 121–151

5 Parameters

1: JOBZ – CHARACTER(1) *Input*

On entry: indicates whether eigenvectors are computed.

JOBZ = 'N'
Only eigenvalues are computed.

JOBZ = 'V'
Eigenvalues and eigenvectors are computed.

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

2: RANGE – CHARACTER(1) *Input*

On entry: indicates which eigenvalues should be returned.

RANGE = 'A'
All eigenvalues will be found.

RANGE = 'V'
All eigenvalues in the half-open interval $(\mathrm{VL}, \mathrm{VU}]$ will be found.

RANGE = 'I'
The ILth through IUth eigenvectors will be found.

Constraint: RANGE = 'A', 'V' or 'I'.

3: N – INTEGER *Input*

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

Constraint: $\mathrm{N} \geq 0$.

4: 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: D is overwritten.

5: 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: $\mathrm{E}(1 : \mathrm{N}-1)$ contains the subdiagonal elements of the tridiagonal matrix T. E(N) need not be set.

On exit: E is overwritten.

6: VL – REAL (KIND=nag_wp) *Input*
7: VU – REAL (KIND=nag_wp) *Input*

On entry: if RANGE = 'V', VL and VU contain the lower and upper bounds respectively of the interval to be searched for eigenvalues.

If RANGE = 'A' or 'I', VL and VU are not referenced.

Constraint: if RANGE = 'V', VL $<$ VU.

8: IL – INTEGER *Input*
9: IU – INTEGER *Input*

On entry: if RANGE = 'I', IL and IU contains the indices (in ascending order) of the smallest and largest eigenvalues to be returned, respectively.

If RANGE = 'A' or 'V', IL and IU are not referenced.

Constraints:

if RANGE = 'I' and N $>$ 0, $1 \leq$ IL $\leq$ IU $\leq$ N;
if RANGE = 'I' and N = 0, IL = 1 and IU = 0.

10: ABSTOL – REAL (KIND=nag_wp) *Input*

On entry: in earlier versions, this argument was the absolute error tolerance for the eigenvalues/ eigenvectors. It is now deprecated, and only included for backwards-compatibility.

11: M – INTEGER *Output*

On exit: the total number of eigenvalues found. $0 \leq$ M $\leq$ N.

If RANGE = 'A', M = N.

If RANGE = 'I', M = IU − IL + 1.

12: W$(*)$ – REAL (KIND=nag_wp) array *Output*

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

On exit: the eigenvalues in ascending order.

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

Note: the second dimension of the array Z must be at least max$(1,$ M$)$ if JOBZ = 'V', and at least 1 otherwise.

On exit: if JOBZ = 'V', then if INFO = 0, the columns of Z contain the orthonormal eigenvectors of the matrix T, with the ith column of Z holding the eigenvector associated with W(i).

If JOBZ = 'N', Z is not referenced.

Note: you must ensure that at least max$(1,$ M$)$ columns are supplied in the array Z; if RANGE = 'V', the exact value of M is not known in advance and an upper bound of at least N must be used.

14: LDZ – INTEGER *Input*

On entry: the first dimension of the array Z as declared in the (sub)program from which F08JLF (DSTEGR) is called.

Constraints:

if JOBZ = 'V', LDZ $\geq$ max$(1,$ N$)$;
otherwise LDZ ≥ 1.

15: ISUPPZ($*$) – INTEGER array *Output*

Note: the dimension of the array ISUPPZ must be at least $\max(1, 2 \times \mathrm{M})$.

On exit: the support of the eigenvectors in Z, i.e., the indices indicating the nonzero elements in Z. The ith eigenvector is nonzero only in elements ISUPPZ$(2 \times i - 1)$ through ISUPPZ$(2 \times i)$.

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

On exit: if INFO $= 0$, WORK(1) returns the minimum LWORK.

17: LWORK – INTEGER *Input*

On entry: the dimension of the array WORK as declared in the (sub)program from which F08JLF (DSTEGR) is called.

If LWORK $= -1$, a workspace query is assumed; the routine only calculates the minimum sizes of the WORK and IWORK arrays, returns these values as the first entries of the WORK and IWORK arrays, and no error message related to LWORK or LIWORK is issued.

Constraint: LWORK $\geq \max(1, 18 \times \mathrm{N})$ or LWORK $= -1$.

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

On exit: if INFO $= 0$, WORK(1) returns the minimum LIWORK.

19: LIWORK – INTEGER *Input*

On entry: the dimension of the array IWORK as declared in the (sub)program from which F08JLF (DSTEGR) is called.

If LIWORK $= -1$, a workspace query is assumed; the routine only calculates the minimum sizes of the WORK and IWORK arrays, returns these values as the first entries of the WORK and IWORK arrays, and no error message related to LWORK or LIWORK is issued.

Constraint: LIWORK $\geq \max(1, 10 \times \mathrm{N})$ or LIWORK $= -1$.

20: 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 $= -i$, argument i had an illegal value. An explanatory message is output, and execution of the program is terminated.

INFO > 0

If INFO $= 1$, the dqds algorithm failed to converge, if INFO $= 2$, inverse iteration failed to converge.

7 Accuracy

See the description for ABSTOL. See also Section 4.7 of Anderson *et al.* (1999) and Barlow and Demmel (1990) for further details.

8 Further Comments

The total number of floating point operations required to compute all the eigenvalues and eigenvectors is approximately proportional to n^2.

The complex analogue of this routine is F08JYF (ZSTEGR).

9 Example

This example finds all the eigenvalues and eigenvectors of the symmetric tridiagonal matrix

$$T = \begin{pmatrix} 1.0 & 1.0 & 0 & 0 \\ 1.0 & 4.0 & 2.0 & 0 \\ 0 & 2.0 & 9.0 & 3.0 \\ 0 & 0 & 3.0 & 16.0 \end{pmatrix}.$$

ABSTOL is set to zero so that the default tolerance of $n\epsilon\|T\|_1$ is used.

9.1 Program Text

```
    Program f08jlfe

!     F08JLF Example Program Text

!     Mark 24 Release. NAG Copyright 2012.

!     .. Use Statements ..
      Use nag_library, Only: dstegr, nag_wp, x04caf
!     .. Implicit None Statement ..
      Implicit None
!     .. Parameters ..
      Real (Kind=nag_wp), Parameter    :: vl = 0.0E0_nag_wp
      Real (Kind=nag_wp), Parameter    :: vu = 0.0E0_nag_wp
      Integer, Parameter               :: il = 0, iu = 0, nin = 5, nout = 6
      Character (1), Parameter         :: range = 'A'
!     .. Local Scalars ..
      Real (Kind=nag_wp)               :: abstol
      Integer                          :: i, ifail, info, ldz, liwork, lwork,  &
                                          m, n
      Character (1)                    :: jobz
!     .. Local Arrays ..
      Real (Kind=nag_wp), Allocatable  :: d(:), e(:), w(:), work(:), z(:,:)
      Integer, Allocatable             :: isuppz(:), iwork(:)
!     .. Executable Statements ..
      Write (nout,*) 'F08JLF Example Program Results'
      Write (nout,*)
!     Skip heading in data file
      Read (nin,*)
      Read (nin,*) n
      ldz = n
      liwork = 10*n
      lwork = 18*n
      Allocate (d(n),e(n),w(n),work(lwork),z(ldz,n),isuppz(2*n),iwork(liwork))

!     Read the symmetric tridiagonal matrix T from data file, first
!     the diagonal elements, then the off diagonal elements and then
!     JOBV ('N' - eigenvalues only, 'V' - vectors as well)

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

!     Calculate all the eigenvalues of T. Set ABSTOL to zero so that
!     the default value is used.

      abstol = 0.0E0_nag_wp
!     The NAG name equivalent of dstegr is f08jlf
      Call dstegr(jobz,range,n,d,e,vl,vu,il,iu,abstol,m,w,z,ldz,isuppz,work, &
        lwork,iwork,liwork,info)

      If (info==0) Then

!       Print eigenvalues and eigenvectors
```

```
        Write (nout,*) 'Eigenvalues'
        Write (nout,99999) w(1:m)

        Write (nout,*)
        Flush (nout)

!       Standardize the eigenvectors so that first elements are non-negative.
        Do i = 1, m
          If (z(1,i)<0.0_nag_wp) z(1:n,i) = -z(1:n,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,m,z,ldz,'Eigenvectors',ifail)

      Else
        Write (nout,99998) 'Failure to compute an eigenvalue, INFO = ', info
      End If

99999 Format ((3X,8F8.4))
99998 Format (1X,A,I10)
    End Program f08jlfe
```

9.2 Program Data

```
F08JLF Example Program Data

  4                    :Value of N

  1.0  4.0  9.0  16.0  :End of D
  1.0  2.0  3.0        :End of E

  'V'                  :Value of JOBZ
```

9.3 Program Results

```
 F08JLF 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
```