

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,
1          LDZ, ISUPPZ, WORK, LWORK, IWORK, LIWORK, INFO)

  INTEGER          N, IL, IU, M, LDZ, ISUPPZ(*), LWORK,
1          IWORK(max(1,LIWORK)), LIWORK, INFO
  double precision D(*), E(*), VL, VU, ABSTOL, W(*), Z(LDZ,*),
1          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

$$T = Z\Lambda Z^T,$$

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

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

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.

In the future this routine may be extended to allow for the computation of selected eigenvalues and eigenvectors.

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: if JOBZ = 'N', compute eigenvalues only.
 If JOBZ = 'V', compute eigenvalues and eigenvectors.
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 (VL, 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: $N \geq 0$.

- 4: D(*) – **double precision** array *Input/Output*
Note: the dimension of the array D must be at least $\max(1, N)$.
On entry: the n diagonal elements of the tridiagonal matrix T .
On exit: D is overwritten.

- 5: E(*) – **double precision** array *Input/Output*
Note: the dimension of the array E must be at least $\max(1, N)$.
On entry: E(1 : N – 1) contains the subdiagonal elements of the tridiagonal matrix T . E(N) need not be set.
On exit: E is overwritten.

- 6: VL – **double precision** *Input*
 7: VU – **double precision** *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 VL are not referenced.
Constraint: 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** – *double precision* *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(*)** – *double precision* 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,*)** – *double precision* array *Output*
Note: the second dimension of the array Z must be at least $\max(1, N)$.
On exit: if JOBZ = 'V', then if INFO = 0, the columns of Z contain the orthonormal eigenvectors of the matrix T, with the *i*th column of Z holding the eigenvector associated with $W(i)$.
 If JOBZ = 'N', Z is not referenced.
- 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 $LDZ \geq 1$.
- 15: **ISUPPZ(*)** – INTEGER array *Output*
Note: the dimension of the array ISUPPZ must be at least $\max(1, 2 \times M)$.
On exit: the support of the eigenvectors in Z, i.e., the indices indicating the nonzero elements in Z. The *i*th eigenvector is nonzero only in elements ISUPPZ($2 \times i - 1$) through ISUPPZ($2 \times i$).
- 16: **WORK(max(1, LWORK))** – *double precision* 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: $LWORK \geq \max(1, 18 \times N)$ or LWORK = -1.
- 18: **IWORK(max(1, 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: $LIWORK \geq \max(1, 10 \times N)$ 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

```
*      F08JLF Example Program Text
*      Mark 21 Release. NAG Copyright 2004.
*      .. Parameters ..
      INTEGER                NIN, NOUT
      PARAMETER              (NIN=5,NOUT=6)
      INTEGER                NMAX
      PARAMETER              (NMAX=10)
      INTEGER                LDZ, LIWORK, LWORK
      PARAMETER              (LDZ=NMAX,LIWORK=10*NMAX,LWORK=18*NMAX)
      INTEGER                IL, IU
      PARAMETER              (IL=0,IU=0)
      DOUBLE PRECISION       VL, VU
      PARAMETER              (VL=0.0D0,VU=0.0D0)
      CHARACTER              RANGE
```

```

      PARAMETER      (RANGE='A')
*    .. Local Scalars ..
      DOUBLE PRECISION ABSTOL
      INTEGER          I, IFAIL, INFO, M, N
      CHARACTER        JOBZ
*    .. Local Arrays ..
      DOUBLE PRECISION D(NMAX), E(NMAX-1), W(NMAX), WORK(LWORK),
+      Z(LDZ,NMAX)
      INTEGER          ISUPPZ(2*NMAX), IWORK(LIWORK)
*    .. External Subroutines ..
      EXTERNAL         DSTEGR, X04CAF
*    .. Executable Statements ..
      WRITE (NOUT,*) 'F08JLF Example Program Results'
      WRITE (NOUT,*)
*    Skip heading in data file
      READ (NIN,*)
      READ (NIN,*) N
      IF (N.LE.NMAX) THEN
*
*        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(I),I=1,N)
      READ (NIN,*) (E(I),I=1,N-1)
      READ (NIN,*) JOBZ
*
*        Calculate all the eigenvalues of T. Set ABSTOL to zero so that
*        the default value is used.
*
      ABSTOL = 0.0D0
      CALL DSTEGR(JOBZ,RANGE,N,D,E,VL,VU,IL,IU,ABSTOL,M,W,Z,LDZ,
+      ISUPPZ,WORK,LWORK,IWORK,LIWORK,INFO)
*
      IF (INFO.EQ.0) THEN
*
*        Print eigenvalues and eigenvectors
*
*
      WRITE (NOUT,*) 'Eigenvalues'
      WRITE (NOUT,99999) (W(I),I=1,M)
*
      WRITE (NOUT,*)
      IFAIL = 0
      CALL X04CAF('General',' ',N,M,Z,LDZ,'Eigenvectors',IFAIL)
*
      ELSE
*
      WRITE (NOUT,99998)
+      'Failure to compute an eigenvalue, INFO = ', INFO
      END IF
      ELSE
*
      WRITE (NOUT,*) 'NMAX too small'
      END IF
*
99999 FORMAT ((3X,8F8.4))
99998 FORMAT (1X,A,I10)
      END

```

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
