NAG Toolbox

nag_lapack_dstegr (f08jl)

1 Purpose

nag_lapack_dstegr (f08jl) computes all the eigenvalues and, optionally, all the eigenvectors of a real n by n symmetric tridiagonal matrix.

2 Syntax

```
[d, e, m, w, z, isuppz, info] = nag_lapack_dstegr(jobz, range, d, e, vl, vu, il,
iu, 'n', n)
```

```
[d, e, m, w, z, isuppz, info] = f08jl(jobz, range, d, e, vl, vu, il, iu, 'n', n)
```

Note: the interface to this routine has changed since earlier releases of the toolbox:

At Mark 23: *abstol* was made an optional input parameter.

3 Description

nag_lapack_dstegr (f08jl) computes all the eigenvalues and, optionally, the eigenvectors, of a real symmetric tridiagonal matrix T. That is, the function 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

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

The function 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 functions which must be called to perform the reduction to tridiagonal form and apply Q are:

full matrix	f08fe and f08fg
full matrix, packed storage	f08ge and f08gg
band matrix	f08he with **vect** = 'V' and .

This function 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. nag_lapack_dstegr (f08jl) 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 functions in this chapter when all the eigenvectors are required, particularly so compared to those functions 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

5.1 Compulsory Input Parameters

1: **jobz** – CHARACTER(1)

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)

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 **il**th through **iu**th eigenvectors will be found.

Constraint: **range** = 'A', 'V' or 'I'.

3: **d**(:) – REAL (KIND=nag_wp) array

The dimension of the array **d** must be at least max(1, **n**)

The n diagonal elements of the tridiagonal matrix T.

4: **e**(:) – REAL (KIND=nag_wp) array

The dimension of the array **e** must be at least max(1, **n**)

e(1 : **n** − 1) contains the subdiagonal elements of the tridiagonal matrix T. **e**(**n**) need not be set.

5: **vl** – REAL (KIND=nag_wp)
6: **vu** – REAL (KIND=nag_wp)

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**.

7: **il** – INTEGER

8: **iu** – INTEGER

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 $\mathbf{n} > 0$, $1 \leq \mathbf{il} \leq \mathbf{iu} \leq \mathbf{n}$;
if **range** $=$ 'I' and $\mathbf{n} = 0$, $\mathbf{il} = 1$ and $\mathbf{iu} = 0$.

5.2 Optional Input Parameters

1: **n** – INTEGER

Default: the dimension of the array **d**.

n, the order of the matrix T.

Constraint: $\mathbf{n} \geq 0$.

5.3 Output Parameters

1: **d**(:) – REAL (KIND=nag_wp) array

The dimension of the array **d** will be $\max(1, \mathbf{n})$

2: **e**(:) – REAL (KIND=nag_wp) array

The dimension of the array **e** will be $\max(1, \mathbf{n})$

3: **m** – INTEGER

The total number of eigenvalues found. $0 \leq \mathbf{m} \leq \mathbf{n}$.

If **range** $=$ 'A', $\mathbf{m} = \mathbf{n}$.

If **range** $=$ 'I', $\mathbf{m} = \mathbf{iu} - \mathbf{il} + 1$.

4: **w**(:) – REAL (KIND=nag_wp) array

The dimension of the array **w** will be $\max(1, \mathbf{n})$

The eigenvalues in ascending order.

5: **z**(ldz, :) – REAL (KIND=nag_wp) array

The first dimension, ldz, of the array **z** will be

if **jobz** $=$ 'V', $ldz = \max(1, \mathbf{n})$;
otherwise $ldz = 1$.

The second dimension of the array **z** will be $\max(1, \mathbf{m})$ if **jobz** $=$ 'V' and 1 otherwise.

If **jobz** $=$ 'V', then if $\mathbf{info} = 0$, the columns of **z** contain the orthonormal eigenvectors of the matrix T, with the ith column of Z holding the eigenvector associated with $\mathbf{w}(i)$.

If **jobz** $=$ 'N', **z** is not referenced.

6: **isuppz**(:) – INTEGER array

The dimension of the array **isuppz** will be $\max(1, 2 \times \mathbf{n})$

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 $\mathbf{isuppz}(2 \times i - 1)$ through $\mathbf{isuppz}(2 \times i)$.

7: **info** – INTEGER

info $= 0$ unless the function detects an error (see Section 6).

6 Error Indicators and Warnings

info $= -i$

If **info** $= -i$, parameter i had an illegal value on entry. The parameters are numbered as follows:

1: **jobz**, 2: **range**, 3: **n**, 4: **d**, 5: **e**, 6: **vl**, 7: **vu**, 8: **il**, 9: **iu**, 10: **abstol**, 11: **m**, 12: **w**, 13: **z**, 14: **ldz**, 15: **isuppz**, 16: **work**, 17: **lwork**, 18: **iwork**, 19: **liwork**, 20: **info**.

It is possible that **info** refers to a parameter that is omitted from the MATLAB interface. This usually indicates that an error in one of the other input parameters has caused an incorrect value to be inferred.

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 alsoSection 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 function is nag_lapack_zstegr (f08jy).

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

```
     function f08jl_example

fprintf('f08jl example results\n\n');

% Tridiagoal matrix A stored as diagonal and off-diagonal
d = [1; 4; 9; 16];
e = [1; 2; 3; 0];

% All eigenvalues and eigenvectors of A
jobz = 'V';
range = 'A';
vl = 0;
vu = 0;
il = nag_int(0);
iu = nag_int(0);
[w, ~, m, w, z, isuppz, info] = ...
f08jl( ...
```

```
       jobz, range, d, e, vl, vu, il, iu);

% Normalize eigenvectors: largest element positive
for j = 1:m
  [~,k] = max(abs(z(:,j)));
  if z(k,j) < 0;
    z(:,j) = -z(:,j);
  end
end

disp('Eigenvalues');
disp(w(1:m)');
disp('Eigenvectors');
disp(z);
```

9.2 Program Results

```
     f08jl example results

Eigenvalues
    0.6476    3.5470    8.6578   17.1477

Eigenvectors
    0.9396    0.3388    0.0494    0.0034
   -0.3311    0.8628    0.3781    0.0545
    0.0853   -0.3648    0.8558    0.3568
   -0.0167    0.0879   -0.3497    0.9326
```