NAG Toolbox for MATLAB

f08jc

1 Purpose

f08jc 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 Syntax

```
[d, e, z, info] = f08jc(job, d, e, 'n', n)
```

3 Description

f08jc 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

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

4 References

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

5 Parameters

5.1 Compulsory Input Parameters

1: **job – string**

Indicates whether eigenvectors are computed.

job = 'N'

Only eigenvalues are computed.

job = 'V'

Eigenvalues and eigenvectors are computed.

Constraint: **job** = 'N' or 'V'.

2: **d**($*$) **– double array**

Note: the dimension of the array **d** must be at least $\max(1, \mathbf{n})$.

The n diagonal elements of the tridiagonal matrix T.

3: **e**($*$) – **double array**

Note: the dimension of the array **e** must be at least $\max(1, \mathbf{n})$.

The $n - 1$ off-diagonal elements of the tridiagonal matrix T. The nth element of this array is used as workspace.

5.2 Optional Input Parameters

1: **n** – **int32 scalar**

Default: The first dimension of the array **d** and the second dimension of the array **d**. (An error is raised if these dimensions are not equal.)

n, the order of the matrix T.

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

5.3 Input Parameters Omitted from the MATLAB Interface

ldz, work, lwork, iwork, liwork

5.4 Output Parameters

1: **d**($*$) – **double array**

Note: the dimension of the array **d** must be at least $\max(1, \mathbf{n})$.

The eigenvalues of the matrix T in ascending order.

2: **e**($*$) – **double array**

Note: the dimension of the array **e** must be at least $\max(1, \mathbf{n})$.

e is overwritten with intermediate results.

3: **z**(**ldz**,$*$) – **double array**

The first dimension, **ldz**, of the array **z** must satisfy

if **job** = 'V', **ldz** $\geq \max(1, \mathbf{n})$;
if **job** = 'N', **ldz** ≥ 1.

The second dimension of the array must be at least $\max(1, \mathbf{n})$ if **job** = 'V' and at least 1 if **job** = 'N'

If **job** = 'V', **z** contains the orthogonal matrix Z which contains the eigenvectors of T.

If **job** = 'N', **z** is not referenced.

4: **info** – **int32 scalar**

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

6 Error Indicators and Warnings

Errors or warnings detected by the function:

info $= -i$

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

1: **job**, 2: **n**, 3: **d**, 4: **e**, 5: **z**, 6: **ldz**, 7: **work**, 8: **lwork**, 9: **iwork**, 10: **liwork**, 11: **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** $= i$ 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 $i/(\mathbf{n}+1)$ through $\mathrm{mod}(i, \mathbf{n}+1)$.

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:

$$\theta(\tilde{z}_i, z_i) \leq \frac{c(n)\epsilon\|T\|_2}{\min_{i \neq j}\left|\lambda_i - \lambda_j\right|}.$$

Thus the accuracy of a computed eigenvector depends on the gap between its eigenvalue and all the other eigenvalues.

8 Further Comments

There is no complex analogue of this function.

9 Example

```
job = 'V';
d = [1;
     4;
     9;
     16];
e = [1;
     2;
     3;
     0];
[dOut, eOut, z, info] = f08jc(job, d, e)
```

```
dOut =
    0.6476
    3.5470
    8.6578
   17.1477
eOut =
     0
     0
     0
     0
z =
    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
info =
           0
```