NAG Toolbox for MATLAB

f08jj

1 Purpose

f08jj computes some (or all) of the eigenvalues of a real symmetric tridiagonal matrix, by bisection.

2 Syntax

```
[m, nsplit, w, iblock, isplit, info] = f08jj(range, order, vl, vu, il,
iu, abstol, d, e, 'n', n)
```

3 Description

f08jj uses bisection to compute some or all of the eigenvalues of a real symmetric tridiagonal matrix T.

It searches for zero or negligible off-diagonal elements of T to see if the matrix splits into block diagonal form:

$$T = \begin{pmatrix} T_1 & & & & & \\ & T_2 & & & & \\ & & \cdot & & & \\ & & & \cdot & & \\ & & & & \cdot & \\ & & & & & T_p \end{pmatrix}.$$

It performs bisection on each of the blocks T_i and returns the block index of each computed eigenvalue, so that a subsequent call to f08jk to compute eigenvectors can also take advantage of the block structure.

4 References

Kahan W 1966 Accurate eigenvalues of a symmetric tridiagonal matrix *Report CS41* Stanford University

5 Parameters

5.1 Compulsory Input Parameters

1: **range – string**

Indicates which eigenvalues are required.

range = 'A'

All the eigenvalues are required.

range = 'V'

All the eigenvalues in the half-open interval (**vl**,**vu**] are required.

range = 'I'

Eigenvalues with indices **il** to **iu** are required.

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

2: **order – string**

Indicates the order in which the eigenvalues and their block numbers are to be stored.

order = 'B'

The eigenvalues are to be grouped by split-off block and ordered from smallest to largest within each block.

order = 'E'

The eigenvalues for the entire matrix are to be ordered from smallest to largest.

Constraint: **order** = 'B' or 'E'.

3: **vl – double scalar**

4: **vu – double scalar**

If **range** = 'V', the lower and upper bounds, respectively, of the half-open interval (**vl**,**vu**] within which the required eigenvalues lie.

If **range** = 'A' or 'I', **vl** is not referenced.

Constraint: if **range** = 'V', **vl** $<$ **vu**.

5: **il – int32 scalar**

6: **iu – int32 scalar**

If **range** = 'I', the indices of the first and last eigenvalues, respectively, to be computed (assuming that the eigenvalues are in ascending order).

If **range** = 'A' or 'V', **il** is not referenced.

Constraint: if **range** = 'I', $1 \leq$ **il** $\leq$ **iu** $\leq$ **n**.

7: **abstol – double scalar**

The absolute tolerance to which each eigenvalue is required. An eigenvalue (or cluster) is considered to have converged if it lies in an interval of width $\leq$ **abstol**. If **abstol** ≤ 0.0, then the tolerance is taken as ***machine precision*** $\times \|T\|_1$.

8: **d**$(*)$ **– double array**

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

The diagonal elements of the tridiagonal matrix T.

9: **e**$(*)$ **– double array**

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

The off-diagonal elements of the tridiagonal matrix T.

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: **n** ≥ 0.

5.3 Input Parameters Omitted from the MATLAB Interface

work, iwork

5.4 Output Parameters

1: **m – int32 scalar**

m, the actual number of eigenvalues found.

2: **nsplit – int32 scalar**

The number of diagonal blocks which constitute the tridiagonal matrix T.

3: **w**$(*)$ **– double array**

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

The required eigenvalues of the tridiagonal matrix T stored in $\mathbf{w}(1)$ to $\mathbf{w}(m)$.

4: **iblock**$(*)$ **– int32 array**

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

At each row/column j where $\mathbf{e}(j)$ is zero or negligible, T is considered to split into a block diagonal matrix and $\mathbf{iblock}(i)$ contains the block number of the eigenvalue stored in $\mathbf{w}(i)$, for $i = 1, 2, \ldots, m$. Note that $\mathbf{iblock}(i) < 0$ for some i whenever **info** $= 1$ or 3 (see Section 6) and **range** = 'A' or 'V'.

5: **isplit**$(*)$ **– int32 array**

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

The leading **nsplit** elements contain the points at which T splits up into sub-matrices as follows. The first sub-matrix consists of rows/columns 1 to $\mathbf{isplit}(1)$, the second sub-matrix consists of rows/columns $\mathbf{isplit}(1) + 1$ to $\mathbf{isplit}(2)$, $\ldots$, and the **nsplit**(th) sub-matrix consists of rows/columns $\mathbf{isplit}(\mathbf{nsplit} - 1) + 1$ to $\mathbf{isplit}(\mathbf{nsplit})$ $(= n)$.

6: **info – int32 scalar**

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: **range**, 2: **order**, 3: **n**, 4: **vl**, 5: **vu**, 6: **il**, 7: **iu**, 8: **abstol**, 9: **d**, 10: **e**, 11: **m**, 12: **nsplit**, 13: **w**, 14: **iblock**, 15: **isplit**, 16: **work**, 17: **iwork**, 18: **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 $= 1$

If **range** = 'A' or 'V', the algorithm failed to compute some (or all) of the required eigenvalues to the required accuracy. More precisely, $\mathbf{iblock}(i) < 0$ indicates that eigenvalue i (stored in $\mathbf{w}(i)$) failed to converge.

info $= 2$

If **range** = 'I', the algorithm failed to compute some (or all) of the required eigenvalues. Try calling the function again with **range** = 'A'.

info $= 3$

If **range** = 'I', see the description above for **info** $= 2$.

If **range** = 'A' or 'V', see the description above for **info** $= 1$.

info = 4

No eigenvalues have been computed. The floating-point arithmetic on the computer is not behaving as expected.

If failures with **info** ≥ 1 are causing persistent trouble and you have checked that the function is being called correctly, please contact NAG.

7 Accuracy

The eigenvalues of T are computed to high relative accuracy which means that if they vary widely in magnitude, then any small eigenvalues will be computed more accurately than, for example, with the standard QR method. However, the reduction to tridiagonal form (prior to calling the function) may exclude the possibility of obtaining high relative accuracy in the small eigenvalues of the original matrix if its eigenvalues vary widely in magnitude.

8 Further Comments

There is no complex analogue of this function.

9 Example

```
range = 'I';
order = 'B';
vl = 0;
vu = 0;
il = int32(1);
iu = int32(2);
abstol = 0;
d = [2.07;
     1.474093708197552;
     -0.6491595075457843;
     -1.694934200651768];
e = [-5.825753170191817;
     2.624045178795586;
     0.9162727563219193];
[m, nsplit, w, iblock, isplit, info] = ...
    f08jj(range, order, vl, vu, il, iu, abstol, d, e)
```

```
m =
           2
nsplit =
           1
w =
   -5.0034
   -1.9987
         0
         0
iblock =
           1
           1
           0
           0
isplit =
           4
           0
           0
           0
info =
           0
```