NAG Toolbox for MATLAB

f08jx

1 Purpose

f08jx computes the eigenvectors of a real symmetric tridiagonal matrix corresponding to specified eigenvalues, by inverse iteration, storing the eigenvectors in a ***complex*** array.

2 Syntax

```
[z, ifailv, info] = f08jx(d, e, m, w, iblock, isplit, 'n', n)
```

3 Description

f08jx computes the eigenvectors of a real symmetric tridiagonal matrix T corresponding to specified eigenvalues, by inverse iteration (see Jessup and Ipsen 1992). It is designed to be used in particular after the specified eigenvalues have been computed by f08jj with **order** = 'B', but may also be used when the eigenvalues have been computed by other functions in Chapters F08 or F02.

The eigenvectors of T are real, but are stored by this function in a ***complex*** array. If T has been formed by reduction of a full complex Hermitian matrix A to tridiagonal form, then eigenvectors of T may be transformed to (complex) eigenvectors of A by a call to f08fu or f08gu.

f08jj determines whether the matrix T splits into block diagonal form:

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

and passes details of the block structure to this function in the arrays **iblock** and **isplit**. This function can then take advantage of the block structure by performing inverse iteration on each block T_i separately, which is more efficient than using the whole matrix.

4 References

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

Jessup E and Ipsen I C F 1992 Improving the accuracy of inverse iteration *SIAM J. Sci. Statist. Comput.* **13** 550–572

5 Parameters

5.1 Compulsory Input Parameters

1: **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.

2: **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.

3: **m – int32 scalar**

m, the number of eigenvectors to be returned.

Constraint: $0 \leq \mathbf{m} \leq \mathbf{n}$.

4: **w**$(*)$ **– double array**

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

The eigenvalues of the tridiagonal matrix T stored in $\mathbf{w}(1)$ to $\mathbf{w}(m)$, as returned by f08jj with **order** = 'B'. Eigenvalues associated with the first sub-matrix must be supplied first, in nondecreasing order; then those associated with the second sub-matrix, again in nondecreasing order; and so on.

Constraint: $\mathbf{w}(i) \leq \mathbf{w}(i+1)$ if $\mathbf{iblock}(i) = \mathbf{iblock}(i+1)$, for $i = 1, 2, \ldots, \mathbf{m} - 1$.

5: **iblock**$(*)$ **– int32 array**

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

The first *m* elements must contain the sub-matrix indices associated with the specified eigenvalues, as returned by f08jj with **order** = 'B'. If the eigenvalues were not computed by f08jj with **order** = 'B', set $\mathbf{iblock}(i)$ to 1, for $i = 1, 2, \ldots, m$.

Constraint: $\mathbf{iblock}(i) \leq \mathbf{iblock}(i+1)$, for $i = 1, 2, \ldots, \mathbf{m} - 1$.

6: **isplit**$(*)$ **– int32 array**

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

The points at which T breaks up into sub-matrices, as returned by f08jj with **order** = 'B'. If the eigenvalues were not computed by f08jj with **order** = 'B', set $\mathbf{isplit}(1)$ to **n**.

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, iwork

5.4 Output Parameters

1: **z**(**ldz**,$*$) **– complex array**

The first dimension of the array **z** must be at least $\max(1, \mathbf{n})$

The second dimension of the array must be at least $\max(1, \mathbf{m})$

The *m* eigenvectors, stored as columns of Z; the *i*th column corresponds to the *i*th specified eigenvalue, unless **info** > 0 (in which case see Section 6).

2: **ifailv**$(*)$ **– int32 array**

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

If $\mathbf{info} = i > 0$, the first i elements of **ifailv** contain the indices of any eigenvectors which have failed to converge. The rest of the first **m** elements of **ifailv** are set to 0.

3: **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: **n**, 2: **d**, 3: **e**, 4: **m**, 5: **w**, 6: **iblock**, 7: **isplit**, 8: **z**, 9: **ldz**, 10: **work**, 11: **iwork**, 12: **ifailv**, 13: **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** $= i$, then i eigenvectors (as indicated by the parameter **ifailv** above) each failed to converge in five iterations. The current iterate after five iterations is stored in the corresponding column of **z**.

7 Accuracy

Each computed eigenvector z_i is the exact eigenvector of a nearby matrix $A + E_i$, such that

$$\|E_i\| = O(\epsilon)\|A\|,$$

where ϵ is the ***machine precision***. Hence the residual is small:

$$\|Az_i - \lambda_i z_i\| = O(\epsilon)\|A\|.$$

However, a set of eigenvectors computed by this function may not be orthogonal to so high a degree of accuracy as those computed by f08js.

8 Further Comments

The real analogue of this function is f08jk.

9 Example

```
d = [-2.28;
     -0.1284569816493291;
     -0.1665932537524081;
     -1.9249497645982631;
e = [-4.33845594653213;
     -2.022594578622617;
     -1.802322978338735];
m = int32(2);
w = [-6.00018545434124;
     -3.003033650781957;
     0;
     0];
iblock = [int32(1);
     int32(1);
     int32(0);
     int32(0)];
isplit = [int32(4);
     int32(0);
     int32(0);
     int32(0)];
[z, ifailv, info] = f08jx(d, e, m, w, iblock, isplit)
```

```
z =
```

```
    0.7299   -0.2595
    0.6259   -0.0433
    0.2513    0.4952
    0.1112    0.8279
ifailv =
           0
           0
info =
           0
```