NAG Toolbox for MATLAB

f08jd

1 Purpose

f08jd computes selected eigenvalues and, optionally, eigenvectors of a real n by n symmetric tridiagonal matrix T. Eigenvalues and eigenvectors can be selected by specifying either a range of values or a range of indices for the desired eigenvalues.

2 Syntax

```
[d, e, m, w, z, isuppz, info] = f08jd(jobz, range, d, e, vl, vu, il, iu,
abstol, 'n', n)
```

3 Description

Whenever possible f08jd computes the eigenspectrum using Relatively Robust Representations. f08jd computes eigenvalues by the **dqds** algorithm, while orthogonal eigenvectors are computed from various 'good' LDL^{T} representations (also known as Relatively Robust Representations). Gram–Schmidt orthogonalisation is avoided as far as possible. More specifically, the various steps of the algorithm are as follows. For the ith unreduced block of T:

(a) compute $T - \sigma_i I = L_i D_i L_i^{\mathrm{T}}$, such that $L_i D_i L_i^{\mathrm{T}}$ is a relatively robust representation,

(b) compute the eigenvalues, λ_j, of $L_i D_i L_i^{\mathrm{T}}$ to high relative accuracy by the dqds algorithm,

(c) if there is a cluster of close eigenvalues, 'choose' σ_i close to the cluster, and go to ,

(d) given the approximate eigenvalue λ_j of $L_i D_i L_i^{\mathrm{T}}$, compute the corresponding eigenvector by forming a rank-revealing twisted factorization.

The desired accuracy of the output can be specified by the parameter **abstol**. For more details, see Dhillon 1997 and Parlett and Dhillon 2000.

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 URL: 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

Demmel J W and Kahan W 1990 Accurate singular values of bidiagonal matrices *SIAM J. Sci. Statist. Comput.* **11** 873–912

Dhillon I 1997 A new On^2 algorithm for the symmetric tridiagonal eigenvalue/eigenvector problem *Computer Science Division Technical Report No. UCB//CSD-97-971* UC Berkeley

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

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 – string**

If **jobz** = 'N', compute eigenvalues only.

If **jobz** = 'V', compute eigenvalues and eigenvectors.

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

2: **range – string**

If **range** = 'A', all eigenvalues will be found.

If **range** = 'V', all eigenvalues in the half-open interval $(\mathbf{vl}, \mathbf{vu}]$ will be found.

If **range** = 'I', the **il**th to **iu**th eigenvalues will be found.

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

3: **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.

4: **e**$(*)$ **– double array**

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

The $(n-1)$ subdiagonal elements of the tridiagonal matrix T.

5: **vl – double scalar**
6: **vu – double scalar**

If **range** = 'V', the lower and upper bounds of the interval to be searched for eigenvalues.

If **range** = 'A' or 'I', **vl** and **vu** are not referenced.

Constraint: if **range** = 'V', $\mathbf{vl} < \mathbf{vu}$.

7: **il – int32 scalar**
8: **iu – int32 scalar**

If **range** = 'I', the indices (in ascending order) of the smallest and largest eigenvalues to be returned.

If **range** = 'A' or 'V', **il** and **iu** are not referenced.

Constraints:

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

9: **abstol – double scalar**

The absolute error tolerance for the eigenvalues. An approximate eigenvalue is accepted as converged when it is determined to lie in an interval $[a, b]$ of width less than or equal to

$$\mathbf{abstol} + \epsilon \max(|a|, |b|),$$

where ϵ is the ***machine precision***. If **abstol** is less than or equal to zero, then $\epsilon \|T\|_1$ will be used in its place. See Demmel and Kahan 1990.

If high relative accuracy is important, set **abstol** to x02am( ), although doing so does not currently guarantee that eigenvalues are computed to high relative accuracy. See Barlow and Demmel 1990 for a discussion of which matrices can define their eigenvalues to high relative accuracy.

5.2 Optional Input Parameters

1: **n – int32 scalar**

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

n, the order of the matrix.

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

5.3 Input Parameters Omitted from the MATLAB Interface

ldz, work, lwork, iwork, liwork

5.4 Output Parameters

1: $\mathbf{d}(*)$ **– double array**

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

May be multiplied by a constant factor chosen to avoid over/underflow in computing the eigenvalues.

2: $\mathbf{e}(*)$ **– double array**

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

May be multiplied by a constant factor chosen to avoid over/underflow in computing the eigenvalues.

3: **m – int32 scalar**

The total number of eigenvalues found.

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

If **range** = 'V', the exact value of **m** is not known in advance, but will satisfy $0 \leq \mathbf{m} \leq \mathbf{n}$.

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

4: $\mathbf{w}(*)$ **– double array**

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

The first **m** elements contain the selected eigenvalues in ascending order.

5: $\mathbf{z}(\mathbf{ldz},*)$ **– double array**

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

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

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

If **jobz** = 'V', then if **info** $= 0$, the first m columns of Z contain the orthonormal eigenvectors of the matrix A corresponding to the selected eigenvalues, with the ith column of Z holding the eigenvector associated with $\mathbf{w}(i)$.

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

Note: you must ensure that at least $\max(1, \mathbf{m})$ columns are supplied in the array **z**; if **range** = 'V', the exact value of **m** is not known in advance and an upper bound must be used.

6: **isuppz**($*$) – **int32 array**

Note: the dimension of the array **isuppz** must be at least $\max(1, 2 \times \mathbf{m})$.

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 **isuppz**$(2 \times i - 1)$ through **isuppz**$(2 \times i)$. Implemented only for **range** = 'A' or 'I' and **iu** − **il** = **n** − 1.

7: **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: **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

An internal error has occurred in this function. Please refer to **info** in f08jj.

7 Accuracy

The computed eigenvalues and eigenvectors are exact for a nearby matrix $(A + E)$, where

$$\|E\|_2 = O(\epsilon)\|A\|_2,$$

and ϵ is the ***machine precision***. See Section 4.7 of Anderson *et al.* 1999 for further details.

8 Further Comments

The total number of floating-point operations is proportional to n^2 if **jobz** = 'N' and is proportional to n^3 if **jobz** = 'V' and **range** = 'A', otherwise the number of floating-point operations will depend upon the number of computed eigenvectors.

9 Example

```
jobz = 'Vectors';
range = 'Indices';
d = [1;
     4;
     9;
     16];
e = [1;
     2;
     3];
vl = 0;
vu = 0;
il = int32(2);
iu = int32(3);
abstol = 0;
[dOut, eOut, m, w, z, isuppz, info] = f08jd(jobz, range, d, e, vl, vu,
```

```
il, iu, abstol)
```

```
dOut =
     1
     4
     9
    16
eOut =
     1
     2
     3
m =
           2
w =
    3.5470
    8.6578
         0
         0
z =
    0.3388    0.0494
    0.8628    0.3781
   -0.3648    0.8558
    0.0879   -0.3497
isuppz =
           0
           0
           0
           0
           0
           0
           0
           0
info =
           0
```