NAG Toolbox for MATLAB

f02fc

1 Purpose

f02fc computes selected eigenvalues, and optionally the corresponding eigenvectors, of a real symmetric matrix.

2 Syntax

```
[a, m, w, z, ifail] = f02fc(job, range, uplo, a, wl, wu, il, iu, mest,
'n', n)
```

3 Description

f02fc computes selected eigenvalues, and optionally the corresponding eigenvectors, of a real symmetric matrix A:

$$Az_i = \lambda_i z_i.$$

The eigenvalues λ_i are selected either by *value* (all the eigenvalues in a half-open interval):

$$w_l \leq \lambda_i < w_u$$

or by *index*, assuming that the eigenvalues are indexed in *ascending* order:

$$i_l \leq i \leq i_u, \qquad \text{where} \qquad \lambda_1 \leq \lambda_2 \leq \cdots \leq \lambda_n.$$

4 References

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

Parlett B N 1998 *The Symmetric Eigenvalue Problem* SIAM, Philadelphia

5 Parameters

5.1 Compulsory Input Parameters

1: **job – string**

Indicates whether eigenvectors are to be computed.

job = 'N'

Only eigenvalues are computed.

job = 'V'

Eigenvalues and eigenvectors are computed.

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

2: **range – string**

Indicates how eigenvalues are to be selected.

range = 'V'

Eigenvalues are selected by value (see **wl** and **wu**);

range = 'I'

Eigenvalues are selected by index (see **il** and **iu**).

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

3: **uplo – string**

Indicates whether the upper or lower triangular part of A is stored.

uplo = 'U'

The upper triangular part of A is stored.

uplo = 'L'

The lower triangular part of A is stored.

Constraint: **uplo** = 'U' or 'L'.

4: **a(lda,∗) – double array**

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

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

The n by n symmetric matrix A.

If **uplo** = 'U', the upper triangle of A must be stored and the elements of the array below the diagonal need not be set.

f **uplo** = 'L', the lower triangle of A must be stored and the elements of the array above the diagonal need not be set.

5: **wl – double scalar**

6: **wu – double scalar**

w_l and w_u, the lower and upper bounds of the interval in which eigenvalues are selected if **range** = 'V'. Not referenced if **range** = 'I'.

Constraint: **wu** > **wl**.

7: **il – int32 scalar**

8: **iu – int32 scalar**

i_l and i_u, the lower and upper bounds of the indices of the eigenvalues which are selected if **range** = 'I'. Not referenced if **range** = 'V'.

Constraint: if $\mathbf{n} > 0$, $1 \leq \mathbf{il} \leq \mathbf{iu} \leq \mathbf{n}$.

9: **mest – int32 scalar**

If **job** = 'V', **mest** must be an upper bound on m, the number of eigenvalues and eigenvectors selected. No eigenvectors are computed if **mest** < m.

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

Constraint: $\mathbf{mest} \geq \max(1, m)$; $\mathbf{mest} \geq \mathbf{iu} - \mathbf{il} + 1$ if **range** = 'I'.

5.2 Optional Input Parameters

1: **n – int32 scalar**

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

n, the order of the matrix A.

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

5.3 Input Parameters Omitted from the MATLAB Interface

lda, ldz, work, lwork, iwork

5.4 Output Parameters

1: **a(lda,$*$) – double array**

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

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

The contents of **a** are overwritten. The diagonal and first off-diagonal contain the upper or lower triangle of the symmetric tridiagonal matrix T (see Section 8).

2: **m – int32 scalar**

m, the number of eigenvalues actually selected.

If **range** = 'I', $m = i_u - i_l + 1$.

3: **w($*$) – double array**

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

The first **m** elements hold the selected eigenvalues in ascending order; elements $\mathbf{m} + 1$ to **n** are used as workspace.

4: **z(ldz,mest) – double array**

If **job** = 'V', the first **m** columns of **z** contain the selected eigenvectors, with the ith column holding the eigenvector z_i associated with the eigenvalue λ_i (stored in $\mathbf{w}(i)$).

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

5: **ifail – int32 scalar**

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

6 Error Indicators and Warnings

Errors or warnings detected by the function:

ifail = 1

On entry, **job** $\neq$ 'N' or 'V',
or **range** $\neq$ 'V' or 'I',
or **uplo** $\neq$ 'U' or 'L',
or $\mathbf{n} < 0$,
or $\mathbf{lda} < \max(1, \mathbf{n})$,
or $\mathbf{wu} \leq \mathbf{wl}$ when **range** = 'V',
or $\mathbf{il} < 1$ when **range** = 'I',
or $\mathbf{iu} > \mathbf{n}$, or $\mathbf{il} > \mathbf{iu}$ and $\mathbf{n} > 0$, when **range** = 'I',
or $\mathbf{mest} < 1$,
or $\mathbf{ldz} < 1$, or $\mathbf{ldz} < \mathbf{n}$ when **job** = 'V',
or $\mathbf{lwork} < \max(1, 8 \times \mathbf{n})$.

ifail = 2

The bisection algorithm failed to compute all the eigenvalues. No eigenvectors have been computed.

ifail $= 3$

There are more than **mest** eigenvalues in the specified range. The actual number of eigenvalues in the range is returned in **m**. No eigenvectors have been computed. Rerun with the second dimension of **z** = **mest** $\geq$ **m**.

ifail $= 4$

Inverse iteration failed to compute all the specified eigenvectors. If an eigenvector failed to converge, the corresponding column of **z** is set to zero.

7 Accuracy

If λ_i is an exact eigenvalue, and $\tilde{\lambda}_i$ is the corresponding computed value, then

$$|\tilde{\lambda}_i - \lambda_i| \leq c(n)\epsilon\|A\|_2,$$

where $c(n)$ is a modestly increasing function of n, and ϵ is the ***machine precision***.

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:

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

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

8 Further Comments

f02fc calls functions from LAPACK in Chapter F08. It first reduces A to tridiagonal form T, using an orthogonal similarity transformation: $A = QTQ^{\mathrm{T}}$. Eigenvalues of T are computed by bisection. If eigenvectors are required, eigenvectors of T are computed by inverse iteration, and are transformed to eigenvectors of A by premultiplying them with the orthogonal matrix Q that was used in the reduction to tridiagonal form.

Each eigenvector z is normalized so that $\|z\|_2 = 1$ and the element of largest absolute value is positive.

The time taken by the function is approximately proportional to n^3.

The function can be used to compute *all* eigenvalues, and optionally *all* eigenvectors, by setting **range** = 'I', **il** = 1 and **iu** = **n**. In some circumstances it may do this more efficiently than f02fa, but this depends on the machine, the size of the problem, and the distribution of eigenvalues. Eigenvectors computed by f02fc may not be orthogonal to as high a degree of accuracy as those computed by f02fa.

9 Example

```
job = 'Vectors';
range = 'Index';
uplo = 'L';
a = [4.16, 0, 0, 0;
     -3.12, 5.03, 0, 0;
     0.56, -0.83, 0.76, 0;
     -0.1, 1.18, 0.34, 1.18];
wl = 0;
wu = 0;
il = int32(1);
iu = int32(2);
mest = int32(3);
[aOut, m, w, z, ifail] = f02fc(job, range, uplo, a, wl, wu, il, iu, mest)
```

```
aOut =
```

```
    4.1600         0         0         0
    3.1714    5.2508         0         0
   -0.0890   -0.9731    1.0936         0
    0.0159   -0.9839    0.5533    0.6255
m =
           2
w =
    0.1239
    1.0051
         0
         0
z =
    0.1859   -0.4209         0
    0.3791   -0.3108         0
    0.6621    0.7210         0
   -0.6192    0.4543         0
ifail =
           0
```