NAG Toolbox for MATLAB

f02ec

1 Purpose

f02ec computes selected eigenvalues and eigenvectors of a real general matrix.

2 Syntax

```
[a, m, wr, wi, vr, vi, ifail] = f02ec(crit, a, wl, wu, mest, 'n', n)
```

3 Description

f02ec computes selected eigenvalues and the corresponding right eigenvectors of a real general matrix A:

$$Ax_i = \lambda_i x_i.$$

Eigenvalues λ_i may be selected either by *modulus*, satisfying:

$$w_l \leq |\lambda_i| \leq w_u,$$

or by *real part*, satisfying:

$$w_l \leq \mathrm{Re}(\lambda_i) \leq w_u.$$

Note that even though A is real, λ_i and x_i may be complex. If x_i is an eigenvector corresponding to a complex eigenvalue λ_i, then the complex conjugate vector $\bar{x}_i$ is the eigenvector corresponding to the complex conjugate eigenvalue $\bar{\lambda}_i$. The eigenvalues in a complex conjugate pair λ_i and $\bar{\lambda}_i$ are either both selected or both not selected.

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: **crit – string**

Indicates the criterion for selecting eigenvalues.

crit = 'M'

Eigenvalues are selected according to their moduli: $w_l \leq |\lambda_i| \leq w_u$.

crit = 'R'

Eigenvalues are selected according to their real parts: $w_l \leq \mathrm{Re}(\lambda_i) \leq w_u$.

Constraint: **crit** = 'M' or 'R'.

2: **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 general matrix A.

3: **wl – double scalar**

4: **wu – double scalar**

w_l and w_u, the lower and upper bounds on the criterion for the selected eigenvalues (see **crit**).

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

5: **mest – int32 scalar**

mest must be an upper bound on m, the number of eigenvalues and eigenvectors selected. No eigenvectors are computed if **mest** $< m$.

Constraint: **mest** $\geq \max(1, m)$.

5.2 Optional Input Parameters

1: **n – int32 scalar**

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

n, the order of the matrix A.

Constraint: **n** ≥ 0.

5.3 Input Parameters Omitted from the MATLAB Interface

lda, ldvr, ldvi, work, lwork, iwork, bwork

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 $\max(1, \mathbf{n})$

Contains the Hessenberg form of the balanced input matrix A' (see Section 8).

2: **m – int32 scalar**

m, the number of eigenvalues actually selected.

3: **wr($*$) – double array**

4: **wi($*$) – double array**

Note: the dimension of the arrays **wr** and **wi** must be at least $\max(1, \mathbf{n})$.

The first **m** elements of **wr** and **wi** hold the real and imaginary parts, respectively, of the selected eigenvalues; elements $\mathbf{m}+1$ to **n** contain the other eigenvalues. Complex conjugate pairs of eigenvalues are stored in consecutive elements of the arrays, with the eigenvalue having positive imaginary part first. See also Section 8.

5: **vr(ldvr,mest) – double array**

Contains the real parts of the selected eigenvectors, with the ith column holding the real part of the eigenvector associated with the eigenvalue λ_i (stored in **wr**(i) and **wi**(i)).

6: **vi(ldvi,mest) – double array**

Contains the imaginary parts of the selected eigenvectors, with the ith column holding the imaginary part of the eigenvector associated with the eigenvalue λ_i (stored in **wr**(i) and **wi**(i)).

7: **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, **crit** $\neq$ 'M' or 'R',
or **n** < 0,
or **lda** $< \max(1, \mathbf{n})$,
or **wu** $\leq$ **wl**,
or **mest** < 1,
or **ldvr** $< \max(1, \mathbf{n})$,
or **ldvi** $< \max(1, \mathbf{n})$,
or **lwork** $< \max(1, \mathbf{n} \times (\mathbf{n} + 4))$.

ifail $= 2$

The *QR* 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 **vr** and **vi** $=$ **mest** $\geq$ **m**.

ifail $= 4$

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

7 Accuracy

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

$$\left|\tilde{\lambda}_i - \lambda_i\right| \leq \frac{c(n)\epsilon\left\|A'\right\|_2}{s_i},$$

where $c(n)$ is a modestly increasing function of n, ϵ is the ***machine precision***, and s_i is the reciprocal condition number of λ_i; A' is the balanced form of the original matrix A (see Section 8), and $\left\|A'\right\| \leq \|A\|$.

If x_i is the corresponding exact eigenvector, and $\tilde{x}_i$ is the corresponding computed eigenvector, then the angle $\theta(\tilde{x}_i, x_i)$ between them is bounded as follows:

$$\theta(\tilde{x}_i, x_i) \leq \frac{c(n)\epsilon\left\|A'\right\|_2}{sep_i},$$

where sep_i is the reciprocal condition number of x_i.

The condition numbers s_i and sep_i may be computed from the Hessenberg form of the balanced matrix A' which is returned in the array **a**. This requires calling f08pe with **job** $=$ 'S' to compute the Schur form of A', followed by f08ql.

8 Further Comments

f02ec calls functions from LAPACK in Chapter F08. It first balances the matrix, using a diagonal similarity transformation to reduce its norm; and then reduces the balanced matrix A' to upper Hessenberg form H, using an orthogonal similarity transformation: $A' = QHQ^{\mathrm{T}}$. The function uses the Hessenberg *QR* algorithm to compute all the eigenvalues of H, which are the same as the eigenvalues of A. It computes the eigenvectors of H which correspond to the selected eigenvalues, using inverse iteration. It premultiplies the eigenvectors by Q to form the eigenvectors of A'; and finally transforms the eigenvectors to those of the original matrix A.

Each eigenvector x (real or complex) is normalized so that $\|x\|_2 = 1$, and the element of largest absolute value is real and positive.

The inverse iteration function may make a small perturbation to the real parts of close eigenvalues, and this may shift their moduli just outside the specified bounds. If you are relying on eigenvalues being within the bounds, you should test them on return from f02ec.

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

The function can be used to compute *all* eigenvalues and eigenvectors, by setting **wl** large and negative, and **wu** large and positive.

9 Example

```
crit = 'Moduli';
a = [0.35, 0.45, -0.14, -0.17;
     0.09, 0.07000000000000001, -0.54, 0.35;
     -0.44, -0.33, -0.03, 0.17;
     0.25, -0.32, -0.13, 0.11];
wl = 0.2;
wu = 0.5;
mest = int32(3);
[aOut, m, wr, wi, vr, vi, ifail] = f02ec(crit, a, wl, wu, mest)
```

```
aOut =
    0.3500   -0.1160   -0.3886   -0.2942
   -0.5140    0.1225    0.1004    0.1126
   -0.7285    0.6443   -0.1357   -0.0977
    0.4139   -0.1665    0.4262    0.1632
m =
           2
wr =
   -0.0994
   -0.0994
    0.7995
   -0.1007
wi =
    0.4008
   -0.4008
         0
         0
vr =
   -0.1933   -0.1933         0
    0.2519    0.2519         0
    0.0972    0.0972         0
    0.6760    0.6760         0
vi =
    0.2546   -0.2546         0
   -0.5224    0.5224         0
   -0.3084    0.3084         0
         0         0         0
ifail =
           0
```