NAG Toolbox

nag_eigen_complex_gen_eigsys (f02gc)

1 Purpose

nag_eigen_complex_gen_eigsys (f02gc) computes selected eigenvalues and eigenvectors of a complex general matrix.

2 Syntax

```
[a, m, w, v, ifail] = nag_eigen_complex_gen_eigsys(crit, a, wl, wu, mest, 'n', n)
[a, m, w, v, ifail] = f02gc(crit, a, wl, wu, mest, 'n', n)
```

3 Description

nag_eigen_complex_gen_eigsys (f02gc) computes selected eigenvalues and the corresponding right eigenvectors of a complex 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

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

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** – CHARACTER(1)

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**$(lda, :)$ – COMPLEX (KIND=nag_wp) array

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

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

The n by n general matrix A.

3: **wl** – REAL (KIND=nag_wp)

4: **wu** – REAL (KIND=nag_wp)

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

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

5: **mest** – INTEGER

The second dimension of the array **v**. **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** – INTEGER

Default: the first dimension of the array **a** and the second dimension of the array **a**.

n, the order of the matrix A.

Constraint: **n** ≥ 0.

5.3 Output Parameters

1: **a**$(lda, :)$ – COMPLEX (KIND=nag_wp) array

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

The second dimension of the array **a** will be $\max(1, \mathbf{n})$.

Contains the Hessenberg form of the balanced input matrix A' (see Section 9).

2: **m** – INTEGER

m, the number of eigenvalues actually selected.

3: **w**$(\mathbf{n})$ – COMPLEX (KIND=nag_wp) array

The first **m** elements of **w** hold the selected eigenvalues; elements $\mathbf{m}+1$ to **n** contain the other eigenvalues.

4: **v**$(ldv, \mathbf{mest})$ – COMPLEX (KIND=nag_wp) array

Contains the selected eigenvectors, with the ith column holding the eigenvector associated with the eigenvalue λ_i (stored in **w**(i)).

5: **ifail** – INTEGER

ifail $= 0$ unless the function detects an error (see Section 5).

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 $ldv < \max(1, \mathbf{n})$,
or $lwork < \max(1, \mathbf{n} \times (\mathbf{n}+2))$.

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 **v** = **mest** $\geq$ **m**.

ifail $= 4$ (*warning*)

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

ifail $= -99$

An unexpected error has been triggered by this routine. Please contact NAG.

ifail $= -399$

Your licence key may have expired or may not have been installed correctly.

ifail $= -999$

Dynamic memory allocation failed.

7 Accuracy

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

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

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 9), and $\|A'\| \leq \|A\|$.

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:

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

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 nag_lapack_zhseqr (f08ps) with **job** = 'S' to compute the Schur form of A', followed by nag_lapack_ztrsna (f08qy).

8 Further Comments

nag_eigen_complex_gen_eigsys (f02gc) 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 a unitary similarity transformation: $A' = QHQ^{\mathrm{H}}$. 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 is normalized so that $\|x\|_2 = 1$, and the element of largest absolute value is real.

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 nag_eigen_complex_gen_eigsys (f02gc).

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

This example computes those eigenvalues of the matrix A which lie in the range $[-5.5, +5.5]$, and their corresponding eigenvectors, where

$$A = \begin{pmatrix} -3.97 - 5.04i & -4.11 + 3.70i & -0.34 + 1.01i & 1.29 - 0.86i \\ 0.34 - 1.50i & 1.52 - 0.43i & 1.88 - 5.38i & 3.36 + 0.65i \\ 3.31 - 3.85i & 2.50 + 3.45i & 0.88 - 1.08i & 0.64 - 1.48i \\ -1.10 + 0.82i & 1.81 - 1.59i & 3.25 + 1.33i & 1.57 - 3.44i \end{pmatrix}.$$

9.1 Program Text

```
function f02gc_example

fprintf('f02gc example results\n\n');

a = [ -3.97 - 5.04i, -4.11 + 3.70i, -0.34 + 1.01i,  1.29 - 0.86i;
       0.34 - 1.50i,  1.52 - 0.43i,  1.88 - 5.38i,  3.36 + 0.65i;
       3.31 - 3.85i,  2.50 + 3.45i,  0.88 - 1.08i,  0.64 - 1.48i;
      -1.10 + 0.82i,  1.81 - 1.59i,  3.25 + 1.33i,  1.57 - 3.44i];

% Find eigenvalues of A such that |eig| <= 5.5
crit = 'Moduli';
wl = 0.0;
wu = 5.5;
mest = nag_int(3);

[a, m, w, v, ifail] = f02gc( ...
                             crit, a, wl, wu, mest);

evals = w(1:m)';
disp('Eigenvalues');
disp(evals);

evecs = v(:,1:m);
disp('Eigenvectors');
disp(evecs);
```

9.2 Program Results

```
f02gc example results

Eigenvalues
  -5.0000 - 2.0060i   3.0023 + 3.9998i

Eigenvectors
  -0.3865 + 0.1732i  -0.0356 - 0.1782i
  -0.3539 + 0.4529i   0.1264 + 0.2666i
   0.6124 + 0.0000i   0.0129 - 0.2966i
  -0.0859 - 0.3284i   0.8898 + 0.0000i
```