NAG Toolbox for MATLAB

f08px

1 Purpose

f08px computes selected left and/or right eigenvectors of a complex upper Hessenberg matrix corresponding to specified eigenvalues, by inverse iteration.

2 Syntax

```
[w, vl, vr, m, ifaill, ifailr, info] = f08px(job, eigsrc, initv, select,
h, w, vl, vr, mm, 'n', n)
```

3 Description

f08px computes left and/or right eigenvectors of a complex upper Hessenberg matrix H, corresponding to selected eigenvalues.

The right eigenvector x, and the left eigenvector y, corresponding to an eigenvalue λ, are defined by:

$$Hx = \lambda x \qquad \text{and} \qquad y^{\mathrm{H}}H = \lambda y^{\mathrm{H}}\left(\qquad \text{or } H^{\mathrm{H}}y = \bar{\lambda}y\right).$$

The eigenvectors are computed by inverse iteration. They are scaled so that $\max|\mathrm{Re}(x_i)| + |\mathrm{Im}\, x_i| = 1$.

If H has been formed by reduction of a complex general matrix A to upper Hessenberg form, then the eigenvectors of H may be transformed to eigenvectors of A by a call to f08nu.

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: **job – string**

Indicates whether left and/or right eigenvectors are to be computed.

job = 'R'

Only right eigenvectors are computed.

job = 'L'

Only left eigenvectors are computed.

job = 'B'

Both left and right eigenvectors are computed.

Constraint: **job** = 'R', 'L' or 'B'.

2: **eigsrc – string**

Indicates whether the eigenvalues of H (stored in **w**) were found using f08ps.

eigsrc = 'Q'

The eigenvalues of H were found using f08ps; thus if H has any zero subdiagonal elements (and so is block triangular), then the jth eigenvalue can be assumed to be an eigenvalue of the block containing the jth row/column. This property allows the function to perform inverse iteration on just one diagonal block.

eigsrc = 'N'

No such assumption is made and the function performs inverse iteration using the whole matrix.

Constraint: **eigsrc** = 'Q' or 'N'.

3: **initv – string**

Indicates whether you are supplying initial estimates for the selected eigenvectors.

initv = 'N'

No initial estimates are supplied.

initv = 'U'

Initial estimates are supplied in **vl** and/or **vr**.

Constraint: **initv** = 'N' or 'U'.

4: **select**$(*)$ **– logical array**

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

Specifies which eigenvectors are to be computed. To select the eigenvector corresponding to the eigenvalue **w**(j), **select**(j) must be set to **true**.

5: **h**(**ldh**,$*$) **– complex array**

The first dimension of the array **h** 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 upper Hessenberg matrix H.

6: **w**$(*)$ **– complex array**

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

The eigenvalues of the matrix H. If **eigsrc** = 'Q', the array **must** be exactly as returned by f08ps.

7: **vl**(**ldvl**,$*$) **– complex array**

The first dimension, **ldvl**, of the array **vl** must satisfy

if **job** = 'L' or 'B', **ldvl** $\geq \max(1, \mathbf{n})$;
if **job** = 'R', **ldvl** ≥ 1.

The second dimension of the array must be at least $\max(1, \mathbf{mm})$ if **job** = 'L' or 'B' and at least 1 if **job** = 'R'

If **initv** = 'U' and **job** = 'L' or 'B', **vl** must contain starting vectors for inverse iteration for the left eigenvectors. Each starting vector must be stored in the same column as will be used to store the corresponding eigenvector (see below).

If **initv** = 'N', **vl** need not be set.

8: **vr(ldvr,$*$) – complex array**

The first dimension, **ldvr**, of the array **vr** must satisfy

if **job** $=$ 'R' or 'B', **ldvr** $\geq \max(1, \mathbf{n})$;
if **job** $=$ 'L', **ldvr** ≥ 1.

The second dimension of the array must be at least $\max(1, \mathbf{mm})$ if **job** $=$ 'R' or 'B' and at least 1 if **job** $=$ 'L'

If **initv** $=$ 'U' and **job** $=$ 'R' or 'B', **vr** must contain starting vectors for inverse iteration for the right eigenvectors. Each starting vector must be stored in the same column as will be used to store the corresponding eigenvector (see below).

If **initv** $=$ 'N', **vr** need not be set.

9: **mm – int32 scalar**

The number of columns in the arrays **vl** and/or **vr** . The actual number of columns required, m, is obtained by counting 1 for each selected real eigenvector and 2 for each selected complex eigenvector (see **select**); $0 \leq m \leq n$.

Constraint: **mm** $\geq m$.

5.2 Optional Input Parameters

1: **n – int32 scalar**

Default: The first dimension of the array **h** and the second dimension of the array **h**. (An error is raised if these dimensions are not equal.)

n, the order of the matrix H.

Constraint: **n** ≥ 0.

5.3 Input Parameters Omitted from the MATLAB Interface

ldh, ldvl, ldvr, work, rwork

5.4 Output Parameters

1: **w($*$) – complex array**

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

The real parts of some elements of **w** may be modified, as close eigenvalues are perturbed slightly in searching for independent eigenvectors.

2: **vl(ldvl,$*$) – complex array**

The first dimension, **ldvl**, of the array **vl** must satisfy

if **job** $=$ 'L' or 'B', **ldvl** $\geq \max(1, \mathbf{n})$;
if **job** $=$ 'R', **ldvl** ≥ 1.

The second dimension of the array must be at least $\max(1, \mathbf{mm})$ if **job** $=$ 'L' or 'B' and at least 1 if **job** $=$ 'R'

If **job** $=$ 'L' or 'B', **vl** contains the computed left eigenvectors (as specified by **select**). The eigenvectors are stored consecutively in the columns of the array, in the same order as their eigenvalues.

If **job** $=$ 'R', **vl** is not referenced.

3: **vr(ldvr,$*$) – complex array**

The first dimension, **ldvr**, of the array **vr** must satisfy

if **job** $=$ 'R' or 'B', **ldvr** $\geq \max(1, \mathbf{n})$;
if **job** $=$ 'L', **ldvr** ≥ 1.

The second dimension of the array must be at least $\max(1, \mathbf{mm})$ if **job** $=$ 'R' or 'B' and at least 1 if **job** $=$ 'L'

If **job** $=$ 'R' or 'B', **vr** contains the computed right eigenvectors (as specified by **select**). The eigenvectors are stored consecutively in the columns of the array, in the same order as their eigenvalues.

If **job** $=$ 'L', **vr** is not referenced.

4: **m – int32 scalar**

m, the number of selected eigenvectors.

5: **ifaill($*$) – int32 array**

Note: the dimension of the array **ifaill** must be at least $\max(1, \mathbf{mm})$ if **job** $=$ 'L' or 'B' and at least 1 if **job** $=$ 'R'.

If **job** $=$ 'L' or 'B', then **ifaill**$(i) = 0$ if the selected left eigenvector converged and **ifaill**$(i) = j > 0$ if the eigenvector stored in the ith row or column of **vl** (corresponding to the jth eigenvalue) failed to converge.

If **job** $=$ 'R', **ifaill** is not referenced.

6: **ifailr($*$) – int32 array**

Note: the dimension of the array **ifailr** must be at least $\max(1, \mathbf{mm})$ if **job** $=$ 'R' or 'B' and at least 1 if **job** $=$ 'L'.

If **job** $=$ 'R' or 'B', then **ifailr**$(i) = 0$ if the selected right eigenvector converged and **ifailr**$(i) = j > 0$ if the eigenvector stored in the ith column of **vr** (corresponding to the jth eigenvalue) failed to converge.

If **job** $=$ 'L', **ifailr** is not referenced.

7: **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: **job**, 2: **eigsrc**, 3: **initv**, 4: **select**, 5: **n**, 6: **h**, 7: **ldh**, 8: **w**, 9: **vl**, 10: **ldvl**, 11: **vr**, 12: **ldvr**, 13: **mm**, 14: **m**, 15: **work**, 16: **rwork**, 17: **ifaill**, 18: **ifailr**, 19: **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 parameters **ifaill** and/or **ifailr** above) failed to converge. The corresponding columns of **vl** and/or **vr** contain no useful information.

7 Accuracy

Each computed right eigenvector x_i is the exact eigenvector of a nearby matrix $A + E_i$, such that $\|E_i\| = O(\epsilon)\|A\|$. Hence the residual is small:

$$\|Ax_i - \lambda_i x_i\| = O(\epsilon)\|A\|.$$

However, eigenvectors corresponding to close or coincident eigenvalues may not accurately span the relevant subspaces.

Similar remarks apply to computed left eigenvectors.

8 Further Comments

The real analogue of this function is f08pk.

9 Example

```
job = 'Right';
eigsrc = 'QR';
initv = 'No initial vectors';
select = [true;
     true;
     false;
     false];
h    =    [complex(-3.97,    -5.04),    complex(-1.131805187339771,    -
2.56930489882744), ...
            complex(-4.602742437533554,   -0.142631904083292),   complex(-
1.424912289366528, +1.732983703342187);
          complex(-5.479653273702635,  +0),  complex(1.858472820765587,  -
1.55018070644029), ...
            complex(4.414465526917012,   -0.7638237115550983),   complex(-
0.4805261336990153, -1.197599997332747);
                      complex(0.6932222118146283,    -0.4828752762602551),
complex(6.267276818064224, ...
      +0), complex(-0.4503809403345012, -0.02898183259817966), complex(-
1.346684450078734, +1.65792489538873);
                     complex(-0.2112946907920694,    +0.0864412259893682),
complex(0.1242146188766495, ...
               -0.2289276049796828),   complex(-3.499985837393258,   +0),
complex(2.561908119568915, -3.370837460961531)];
w = [complex(-6.000425342949246, -6.999843371570391);
     complex(-5.000033457596968, +2.006027162316515);
     complex(7.998194516208248, -0.9963650913929011);
     complex(3.002264284337973, -3.999818699353226)];
vl = [complex(0, +0)];
vr = [complex(0, +0), complex(0, +0), complex(0, +0), complex(0, +0);
     complex(0, +0), complex(0, +0), complex(0, +0), complex(0, +0);
     complex(0, +0), complex(0, +0), complex(0, +0), complex(0, +0);
     complex(0, +0), complex(0, +0), complex(0, +0), complex(0, +0)];
mm = int32(4);
[wOut, vlOut, vrOut, m, ifaill, ifailr, info] = ...
    f08px(job, eigsrc, initv, select, h, w, vl, vr, mm)
```

```
wOut =
  -6.0004 - 6.9998i
  -5.0000 + 2.0060i
   7.9982 - 0.9964i
   3.0023 - 3.9998i
vlOut =
     0
vrOut =
   0.3815 - 0.6185i  -0.2691 - 0.5053i        0                  0
  -0.1142 - 0.3555i   0.5760 - 0.2995i        0                  0
   0.2385 + 0.0716i  -0.9708 + 0.0292i        0                  0
   0.0932 - 0.0102i  -0.3048 - 0.2032i        0                  0
```

```
m =
           2
ifaill =
           0
ifailr =
           0
           0
           0
           0
info =
           0
```