NAG Toolbox

nag_lapack_zhsein (f08px)

1 Purpose

nag_lapack_zhsein (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] = nag_lapack_zhsein(job, eigsrc, initv,
select, h, w, vl, vr, mm, 'n', n)
```

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

3 Description

nag_lapack_zhsein (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:

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

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 nag_lapack_zunmhr (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** – CHARACTER(1)

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** – CHARACTER(1)

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

eigsrc = 'Q'
The eigenvalues of H were found using nag_lapack_zhseqr (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** – CHARACTER(1)

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

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 $\mathbf{w}(j)$, $\mathbf{select}(j)$ must be set to *true*.

5: $\mathbf{h}(ldh, :)$ – COMPLEX (KIND=nag_wp) array

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

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

The n by n upper Hessenberg matrix H.

6: **w**(:) – COMPLEX (KIND=nag_wp) array

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 nag_lapack_zhseqr (f08ps).

7: $\mathbf{vl}(ldvl, :)$ – COMPLEX (KIND=nag_wp) array

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

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

The second dimension of the array **vl** 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 (KIND=nag_wp) array

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

if **job** = 'R' or 'B', $ldvr \geq$ **n**;
if **job** = 'L', $ldvr \geq 1$.

The second dimension of the array **vr** must be at least max(1, **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** – INTEGER

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** – INTEGER

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 Output Parameters

1: **w**(:) – COMPLEX (KIND=nag_wp) array

The dimension of the array **w** will be max(1, **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 (KIND=nag_wp) array

The first dimension, $ldvl$, of the array **vl** will be

if **job** = 'L' or 'B', $ldvl =$ **n**;
if **job** = 'R', $ldvl = 1$.

The second dimension of the array **vl** will be max(1, **mm**) if **job** = 'L' or 'B' and at least 1 if **job** = 'R'.

The second dimension of the array **vl** will be max(1, **m**) 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 (KIND=nag_wp) array

The first dimension, $ldvr$, of the array **vr** will be

if **job** = 'R' or 'B', $ldvr$ = **n**;
if **job** = 'L', $ldvr = 1$.

The second dimension of the array **vr** will be max(1, **mm**) if **job** = 'R' or 'B' and at least 1 if **job** = 'L'.

The second dimension of the array **vr** will be max(1, **m**) 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** – INTEGER

m, the number of selected eigenvectors.

5: **ifaill**(:) – INTEGER array

The dimension of the array **ifaill** will be max(1, **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**(:) – INTEGER array

The dimension of the array **ifailr** will be max(1, **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** – INTEGER

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 arguments **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 nag_lapack_dhsein (f08pk).

9 Example

See Section 10 in nag_lapack_zunmhr (f08nu).

9.1 Program Text

```
     function f08px_example

fprintf('f08px example results\n\n');

% Complex matrix A
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];

% Reduce (all of) A to upper Hessenberg Form
n   = nag_int(4);
ilo = nag_int(1);
ihi = n;
[H, tau, info] = f08ns(ilo, ihi, a);

% Form Q
[Q, info] = f08nt(ilo, ihi, H, tau);

% Schur factorize H = Y*T*Y^H
job   = 'Schur form';
compz = 'Vectors';
[~, w, ~, info] = f08ps( ...
                         job, compz, ilo, ihi, H, Q);

disp('Eigenvalues of A');
disp(w);

% Calculate eigenvectors of H for negative real part eigenvalues
select = (real(w) < 0);

job = 'Right';
eigsrc = 'QR';
initv = 'No initial vectors';
vl = [];
vr = complex(zeros(n,n));
[~, ~, VR, m, ifaill, ifailr, info] = ...
f08px( ...
       job, eigsrc, initv, select, H, w, vl, vr, n);

% Eigenvectors of A = Q*VR
side = 'Left';
trans = 'No transpose';
[z, info] = f08nu( ...
                   side, trans, ilo, ihi, H, tau, VR);
```

```
% Normalize eigenvectors: largest elements are real
for i = 1:m
  [~,k] = max(abs(real(z(:,i)))+abs(imag(z(:,i))));
  z(:,i) = z(:,i)*conj(z(k,i))/abs(z(k,i));
end

disp('Eigenvectors corresponding to eigenvalues with negative real part');
% Normalize eigenvectors before printing
disp(z/diag(z(1,:)));
```

9.2 Program Results

```
     f08px example results

Eigenvalues of A
  -6.0004 - 6.9998i
  -5.0000 + 2.0060i
   7.9982 - 0.9964i
   3.0023 - 3.9998i

Eigenvectors corresponding to eigenvalues with negative real part
   1.0000 + 0.0000i   1.0000 + 0.0000i
  -0.0210 + 0.3590i   1.1997 - 0.6339i
   0.1035 + 0.3683i  -1.3192 - 0.5912i
  -0.0664 - 0.3436i  -0.1319 + 0.7904i
```
