NAG Toolbox

nag_lapack_ztrevc (f08qx)

1 Purpose

nag_lapack_ztrevc (f08qx) computes selected left and/or right eigenvectors of a complex upper triangular matrix.

2 Syntax

```
[t, vl, vr, m, info] = nag_lapack_ztrevc(job, howmny, select, t, vl, vr, mm, 'n',
n)
```

```
[t, vl, vr, m, info] = f08qx(job, howmny, select, t, vl, vr, mm, 'n', n)
```

3 Description

nag_lapack_ztrevc (f08qx) computes left and/or right eigenvectors of a complex upper triangular matrix T. Such a matrix arises from the Schur factorization of a complex general matrix, as computed by nag_lapack_zhseqr (f08ps), for example.

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

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

The function can compute the eigenvectors corresponding to selected eigenvalues, or it can compute all the eigenvectors. In the latter case the eigenvectors may optionally be pre-multiplied by an input matrix Q. Normally Q is a unitary matrix from the Schur factorization of a matrix A as $A = QTQ^{\mathrm{H}}$; if x is a (left or right) eigenvector of T, then Qx is an eigenvector of A.

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

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: **howmny** – CHARACTER(1)

Indicates how many eigenvectors are to be computed.

howmny = 'A'
All eigenvectors (as specified by **job**) are computed.

howmny = 'B'
All eigenvectors (as specified by **job**) are computed and then pre-multiplied by the matrix Q (which is overwritten).

howmny = 'S'
Selected eigenvectors (as specified by **job** and **select**) are computed.

Constraint: **howmny** = 'A', 'B' or 'S'.

3: **select**(:) – LOGICAL array

The dimension of the array **select** must be at least **n** if **howmny** = 'S', and at least 1 otherwise

Specifies which eigenvectors are to be computed if **howmny** = 'S'. To obtain the eigenvector corresponding to the eigenvalue λ_j, **select**(j) must be set *true*.

If **howmny** = 'A' or 'B', **select** is not referenced.

4: **t**(ldt, :) – COMPLEX (KIND=nag_wp) array

The first dimension of the array **t** must be at least max(1, **n**).

The second dimension of the array **t** must be at least **n**.

The n by n upper triangular matrix T, as returned by nag_lapack_zhseqr (f08ps).

5: **vl**($ldvl$, :) – COMPLEX (KIND=nag_wp) array

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

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

The second dimension of the array **vl** must be at least **mm** if **job** = 'L' or 'B' and at least 1 if **job** = 'R'.

If **howmny** = 'B' and **job** = 'L' or 'B', **vl** must contain an n by n matrix Q (usually the matrix of Schur vectors returned by nag_lapack_zhseqr (f08ps)).

If **howmny** = 'A' or 'S', **vl** need not be set.

6: **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 **mm** if **job** = 'R' or 'B' and at least 1 if **job** = 'L'.

If **howmny** = 'B' and **job** = 'R' or 'B', **vr** must contain an n by n matrix Q (usually the matrix of Schur vectors returned by nag_lapack_zhseqr (f08ps)).

If **howmny** = 'A' or 'S', **vr** need not be set.

7: **mm** – INTEGER

The number of columns in the arrays **vl** and/or **vr**. The precise number of columns required, m, is n if **howmny** = 'A' or 'B'; if **howmny** = 'S', m is the number of selected eigenvectors (see **select**), in which case $0 \leq m \leq n$.

Constraints:

if **howmny** $=$ 'A' or 'B', **mm** $\geq$ **n**;
otherwise **mm** $\geq m$.

5.2 Optional Input Parameters

1: **n** – INTEGER

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

n, the order of the matrix T.

Constraint: **n** ≥ 0.

5.3 Output Parameters

1: **t**$(ldt,:)$ – COMPLEX (KIND=nag_wp) array

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

The second dimension of the array **t** will be **n**.

Is used as internal workspace prior to being restored and hence is unchanged.

2: **vl**$(ldvl,:)$ – COMPLEX (KIND=nag_wp) array

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

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

The second dimension of the array **vl** will be **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 **howmny** and **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', $ldvr = \mathbf{n}$;
if **job** $=$ 'L', $ldvr = 1$.

The second dimension of the array **vr** will be **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 **howmny** and **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. If **howmny** $=$ 'A' or 'B', **m** is set to n.

5: **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: **howmny**, 3: **select**, 4: **n**, 5: **t**, 6: **ldt**, 7: **vl**, 8: **ldvl**, 9: **vr**, 10: **ldvr**, 11: **mm**, 12: **m**, 13: **work**, 14: **rwork**, 15: **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.

7 Accuracy

If x_i is an exact right 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:

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

where sep_i is the reciprocal condition number of x_i.

The condition number sep_i may be computed by calling nag_lapack_ztrsna (f08qy).

8 Further Comments

The real analogue of this function is nag_lapack_dtrevc (f08qk).

9 Example

See Section 10 in nag_lapack_zgebal (f08nv).

9.1 Program Text

```
function f08qx_example

fprintf('f08qx example results\n\n');

n = nag_int(4);
a = [ 1.50 - 2.75i,  0    + 0i,     0    + 0i,      0    + 0i;
     -8.06 - 1.24i, -2.50 - 0.50i,  0    + 0i,    -0.75 + 0.50i;
     -2.09 + 7.56i,  1.39 + 3.97i, -1.25 + 0.75i, -4.82 - 5.67i;
      6.18 + 9.79i, -0.92 - 0.62i,  0    + 0i,    -2.50 - 0.50i];

% Balance a
[a, ilo, ihi, scale, info] = f08nv( ...
     'Both', a);

% Reduce a to upper Hessenberg form
[H, tau, info] = f08ns( ...
 ilo, ihi, a);

% Form Q explicitly, storing result in vr
[Q, info] = f08nt( ...
    ilo, ihi, H, tau);

% Calculate the eigenvalues and Schur factorisation of A
[S, w, VR, info] = f08ps( ...
   'Schur Form', 'Vectors', ilo, ihi, H, Q);

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

% Calculate the eigenvectors of A
select = [false];
```

```
vl = [complex(0)];

[T, ~, Z, m, info] = ...
  f08qx( ...
  'Right', 'Backtransform', select, H, vl, VR, n);

% Scale
[V, info] = f08nw( ...
    'Both', 'Right', ilo, ihi, scale, Z);

% Normalize eigenvectors: largest element is real
for i = 1:m
  [~,k] = max(abs(real(V(:,i)))+abs(imag(V(:,i))));
  V(:,i) = V(:,i)*conj(V(k,i))/abs(V(k,i));
end

disp('Right eigenvectors of A:');
disp(V);
```

9.2 Program Results

```
     f08qx example results

Eigenvalues of A:
  -1.2500 + 0.7500i
  -1.5000 - 0.4975i
  -3.5000 - 0.5025i
   1.5000 - 2.7500i

Right eigenvectors of A:
   0.0000 + 0.0000i   0.0000 + 0.0000i   0.0000 + 0.0000i   0.1778 - 0.0372i
   0.0000 + 0.0000i   0.1902 - 0.0910i   0.1898 - 0.0908i   0.4382 - 0.1021i
   1.0000 + 0.0000i   0.7492 + 0.0000i   0.7479 + 0.0000i   0.8449 + 0.0000i
   0.0000 + 0.0000i  -0.2314 - 0.0337i  -0.2310 - 0.0336i   0.1950 - 0.3509i
```