NAG Toolbox

nag_lapack_ztgevc (f08yx)

1 Purpose

nag_lapack_ztgevc (f08yx) computes some or all of the right and/or left generalized eigenvectors of a pair of complex upper triangular matrices (A, B).

2 Syntax

```
[vl, vr, m, info] = nag_lapack_ztgevc(side, howmny, select, a, b, vl, vr, mm,
'n', n)
```

```
[vl, vr, m, info] = f08yx(side, howmny, select, a, b, vl, vr, mm, 'n', n)
```

3 Description

nag_lapack_ztgevc (f08yx) computes some or all of the right and/or left generalized eigenvectors of the matrix pair (A, B) which is assumed to be in upper triangular form. If the matrix pair (A, B) is not upper triangular then the function nag_lapack_zhgeqz (f08xs) should be called before invoking nag_lapack_ztgevc (f08yx).

The right generalized eigenvector x and the left generalized eigenvector y of (A, B) corresponding to a generalized eigenvalue λ are defined by

$$(A - \lambda B)x = 0$$

and

$$y^{\mathrm{H}}(A - \lambda B) = 0.$$

If a generalized eigenvalue is determined as $0/0$, which is due to zero diagonal elements at the same locations in both A and B, a unit vector is returned as the corresponding eigenvector.

Note that the generalized eigenvalues are computed using nag_lapack_zhgeqz (f08xs) but nag_lapack_ztgevc (f08yx) does not explicitly require the generalized eigenvalues to compute eigenvectors. The ordering of the eigenvectors is based on the ordering of the eigenvalues as computed by nag_lapack_ztgevc (f08yx).

If all eigenvectors are requested, the function may either return the matrices X and/or Y of right or left eigenvectors of (A, B), or the products ZX and/or QY, where Z and Q are two matrices supplied by you. Usually, Q and Z are chosen as the unitary matrices returned by nag_lapack_zhgeqz (f08xs). Equivalently, Q and Z are the left and right Schur vectors of the matrix pair supplied to nag_lapack_zhgeqz (f08xs). In that case, QY and ZX are the left and right generalized eigenvectors, respectively, of the matrix pair supplied to nag_lapack_zhgeqz (f08xs).

4 References

Anderson E, Bai Z, Bischof C, Blackford S, Demmel J, Dongarra J J, Du Croz J J, Greenbaum A, Hammarling S, McKenney A and Sorensen D (1999) *LAPACK Users' Guide* (3rd Edition) SIAM, Philadelphia

Golub G H and Van Loan C F (1996) *Matrix Computations* (3rd Edition) Johns Hopkins University Press, Baltimore

Moler C B and Stewart G W (1973) An algorithm for generalized matrix eigenproblems *SIAM J. Numer. Anal.* **10** 241–256

Stewart G W and Sun J-G (1990) *Matrix Perturbation Theory* Academic Press, London

5 Parameters

5.1 Compulsory Input Parameters

1: **side** – CHARACTER(1)

Specifies the required sets of generalized eigenvectors.

side = 'R'
Only right eigenvectors are computed.

side = 'L'
Only left eigenvectors are computed.

side = 'B'
Both left and right eigenvectors are computed.

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

2: **howmny** – CHARACTER(1)

Specifies further details of the required generalized eigenvectors.

howmny = 'A'
All right and/or left eigenvectors are computed.

howmny = 'B'
All right and/or left eigenvectors are computed; they are backtransformed using the input matrices supplied in arrays **vr** and/or **vl**.

howmny = 'S'
Selected right and/or left eigenvectors, defined by the array **select**, are computed.

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

3: **select**(:) – LOGICAL array

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

Specifies the eigenvectors to be computed if **howmny** = 'S'. To select the generalized eigenvector corresponding to the jth generalized eigenvalue, the jth element of **select** should be set to *true*.

Constraint: if **howmny** = 'S', **select**$(j) = \mathit{true}$ or false, for $j = 1, 2, \ldots, n$.

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

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

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

The matrix A must be in upper triangular form. Usually, this is the matrix A returned by nag_lapack_zhgeqz (f08xs).

5: **b**$(ldb, :)$ – COMPLEX (KIND=nag_wp) array

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

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

The matrix B must be in upper triangular form with non-negative real diagonal elements. Usually, this is the matrix B returned by nag_lapack_zhgeqz (f08xs).

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

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

if **side** = 'L' or 'B', $ldvl \geq$ max(1, **n**);
if **side** = 'R', $ldvl \geq 1$.

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

If **howmny** = 'B' and **side** = 'L' or 'B', **vl** must be initialized to an n by n matrix Q. Usually, this is the unitary matrix Q of left Schur vectors returned by nag_lapack_zhgeqz (f08xs).

7: **vr**($ldvr$, :) – COMPLEX (KIND=nag_wp) array

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

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

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

If **howmny** = 'B' and **side** = 'R' or 'B', **vr** must be initialized to an n by n matrix Z. Usually, this is the unitary matrix Z of right Schur vectors returned by nag_lapack_dhgeqz (f08xe).

8: **mm** – INTEGER

The number of columns in the arrays **vl** and/or **vr**.

Constraints:

if **howmny** = 'A' or 'B', $\mathbf{mm} \geq \mathbf{n}$;
if **howmny** = 'S', **mm** must not be less than the number of requested eigenvectors.

5.2 Optional Input Parameters

1: **n** – INTEGER

Default: the first dimension of the arrays **vl**, **vr**. (An error is raised if these dimensions are not equal.)

n, the order of the matrices A and B.

Constraint: $\mathbf{n} \geq 0$.

5.3 Output Parameters

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

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

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

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

If **side** = 'L' or 'B', **vl** contains:

if **howmny** = 'A', the matrix Y of left eigenvectors of (A, B);

if **howmny** = 'B', the matrix QY;

if **howmny** = 'S', the left eigenvectors of (A, B) specified by **select**, stored consecutively in the columns of the array **vl**, in the same order as their corresponding eigenvalues.

2: **vr**($ldvr$, :) – COMPLEX (KIND=nag_wp) array

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

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

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

If **side** = 'R' or 'B', **vr** contains:

if **howmny** = 'A', the matrix X of right eigenvectors of (A, B);

if **howmny** = 'B', the matrix ZX;

if **howmny** = 'S', the right eigenvectors of (A, B) specified by **select**, stored consecutively in the columns of the array **vr**, in the same order as their corresponding eigenvalues.

3: **m** – INTEGER

The number of columns in the arrays **vl** and/or **vr** actually used to store the eigenvectors. If **howmny** = 'A' or 'B', **m** is set to **n**. Each selected eigenvector occupies one column.

4: **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: **side**, 2: **howmny**, 3: **select**, 4: **n**, 5: **a**, 6: **lda**, 7: **b**, 8: **ldb**, 9: **vl**, 10: **ldvl**, 11: **vr**, 12: **ldvr**, 13: **mm**, 14: **m**, 15: **work**, 16: **rwork**, 17: **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

It is beyond the scope of this manual to summarise the accuracy of the solution of the generalized eigenvalue problem. Interested readers should consult Section 4.11 of the LAPACK Users' Guide (see Anderson *et al.* (1999)) and Chapter 6 of Stewart and Sun (1990).

8 Further Comments

nag_lapack_ztgevc (f08yx) is the sixth step in the solution of the complex generalized eigenvalue problem and is usually called after nag_lapack_zhgeqz (f08xs).

The real analogue of this function is nag_lapack_dtgevc (f08yk).

9 Example

This example computes the α and β arguments, which defines the generalized eigenvalues and the corresponding left and right eigenvectors, of the matrix pair (A, B) given by

$$A = \begin{pmatrix} 1.0 + 3.0i & 1.0 + 4.0i & 1.0 + 5.0i & 1.0 + 6.0i \\ 2.0 + 2.0i & 4.0 + 3.0i & 8.0 + 4.0i & 16.0 + 5.0i \\ 3.0 + 1.0i & 9.0 + 2.0i & 27.0 + 3.0i & 81.0 + 4.0i \\ 4.0 + 0.0i & 16.0 + 1.0i & 64.0 + 2.0i & 256.0 + 3.0i \end{pmatrix}$$

and

$$B = \begin{pmatrix} 1.0 + 0.0i & 2.0 + 1.0i & 3.0 + 2.0i & 4.0 + 3.0i \\ 1.0 + 1.0i & 4.0 + 2.0i & 9.0 + 3.0i & 16.0 + 4.0i \\ 1.0 + 2.0i & 8.0 + 3.0i & 27.0 + 4.0i & 64.0 + 5.0i \\ 1.0 + 3.0i & 16.0 + 4.0i & 81.0 + 5.0i & 256.0 + 6.0i \end{pmatrix}.$$

To compute generalized eigenvalues, it is required to call five functions: nag_lapack_zggbal (f08wv) to balance the matrix, nag_lapack_zgeqrf (f08as) to perform the QR factorization of B, nag_lapack_ zunmqr (f08au) to apply Q to A, nag_lapack_zgghrd (f08ws) to reduce the matrix pair to the generalized Hessenberg form and nag_lapack_zhgeqz (f08xs) to compute the eigenvalues via the QZ algorithm.

The computation of generalized eigenvectors is done by calling nag_lapack_ztgevc (f08yx) to compute the eigenvectors of the balanced matrix pair. The function nag_lapack_zggbak (f08ww) is called to backward transform the eigenvectors to the user-supplied matrix pair. If both left and right eigenvectors are required then nag_lapack_zggbak (f08ww) must be called twice.

9.1 Program Text

```
     function f08yx_example

fprintf('f08yx example results\n\n');

% Generalized eigenvalues of matrix pair (A,B) , where
a = [ 1.0+3.0i  1.0+4.0i  1.0+5.0i   1.0+6.0i;
      2.0+2.0i  4.0+3.0i  8.0+4.0i  16.0+5.0i;
      3.0+1.0i  9.0+2.0i 27.0+3.0i  81.0+4.0i;
      4.0+0.0i 16.0+1.0i 64.0+2.0i 256.0+3.0i];
b = [ 1.0+0.0i  2.0+1.0i  3.0+2.0i   4.0+3.0i;
      1.0+1.0i  4.0+2.0i  9.0+3.0i  16.0+4.0i;
      1.0+2.0i  8.0+3.0i 27.0+4.0i  64.0+5.0i;
      1.0+3.0i 16.0+4.0i 81.0+5.0i 256.0+6.0i];

% Balance matrix pair
job = 'B';
[a, b, ilo, ihi, lscale, rscale, info] = ...
  f08wv(job, a, b);
bbal = b(ilo:ihi,ilo:ihi);
abal = a(ilo:ihi,ilo:ihi);

% QR factorize balanced B
[QR, tau, info] = f08as(bbal);

% Perform C = Q^H*A
side = 'Left';
trans = 'Conjugate transpose';
[c, info] = f08au(...
                  side, trans, QR, tau, abal);

% Generalized Hessenberg form (C,R) -> (H,T)
compq = 'Vectors Q';
compz = 'Vectors Z';

% Form Q explicitly and let Z = I.
[q, info] = f08at(QR, tau);
z = complex(eye(4));

jlo = nag_int(1);
jhi = nag_int(ihi-ilo+1);
[H, T, q, z, info] = ...
  f08ws(...
        compq, compz, jlo, jhi, c, QR, q, z);

% Find eigenvalues of generalized Hessenberg form
%    = eigenvalues of (A,B).
% and return Schur form for computing eigenvectors
job = 'Schur form';
ilo = nag_int(1);
ihi = nag_int(4);
[HS, TS, alpha, beta, q, z, info] = ...
  f08xs(...
        job, compq, compz, jlo, jhi, H, T, q, z);

disp('Generalized eigenvalues of (A,B):');
disp(alpha./beta);

% Obtain scaled eigenvectors from Schur form
```

```
side = 'Both sides';
howmny = 'Backtransformed using Q and Z';
select = [false];
[q, z, m, info] = f08yx(...
                        side, howmny, select, HS, TS, q, z, jhi);

% rescale to obtain left and right eigenvectors of (A,B)
job  = 'Back scale';
side = 'Left';
[VL, info] = f08ww( ...
                    job, side, jlo, jhi, lscale, rscale, q);
side = 'Right';
[VR, info] = f08ww( ...
                    job, side, jlo, jhi, lscale, rscale, z);

% Normalize eigenvectors to have Euclidean norm 1 and largest component real.
incv = nag_int(1);
n = nag_int(4);
for j = 1:n
  [k, r] = f16js(n, VL(:,j), incv);
  scal = conj(VL(k,j))/abs(VL(k,j))/norm(VL(:,j),2);
  VL(:,j) = VL(:,j)*scal;
end
for j = 1:n
  [k, r] = f16js(n, VR(:,j), incv);
  scal = conj(VR(k,j))/abs(VR(k,j))/norm(VR(:,j),2);
  VR(:,j) = VR(:,j)*scal;
end

disp('Right Eigenvectors');
disp(VR);

disp('Left Eigenvectors');
disp(VL);
```

9.2 Program Results

```
     f08yx example results

Generalized eigenvalues of (A,B):
  -0.6354 + 1.6529i
   0.4580 - 0.8426i
   0.4934 + 0.9102i
   0.6744 - 0.0499i

Right Eigenvectors
   0.7186 + 0.0000i  -0.4649 + 0.0156i  -0.3946 + 0.0246i  -0.6788 - 0.1233i
  -0.6208 - 0.2009i   0.7652 + 0.0000i   0.7921 + 0.0000i   0.7184 + 0.0000i
   0.2251 + 0.0762i  -0.4275 - 0.0912i  -0.4554 + 0.0334i  -0.0886 - 0.0067i
  -0.0372 - 0.0088i   0.0707 + 0.0442i   0.0824 - 0.0322i  -0.0048 + 0.0006i

Left Eigenvectors
   0.7397 + 0.0000i  -0.3722 - 0.0016i  -0.3240 - 0.1559i  -0.4118 - 0.2276i
  -0.5812 + 0.2589i   0.8003 + 0.0000i   0.8063 - 0.0000i   0.8681 - 0.0000i
   0.1875 - 0.1097i  -0.4606 - 0.0279i  -0.4523 + 0.0903i  -0.1564 + 0.0136i
  -0.0219 + 0.0195i   0.0839 + 0.0311i   0.0755 - 0.0453i   0.0206 - 0.0038i
```