NAG Toolbox

nag_lapack_zgghrd (f08ws)

1 Purpose

nag_lapack_zgghrd (f08ws) reduces a pair of complex matrices (A, B), where B is upper triangular, to the generalized upper Hessenberg form using unitary transformations.

2 Syntax

```
[a, b, q, z, info] = nag_lapack_zgghrd(compq, compz, ilo, ihi, a, b, q, z, 'n', n)
```

```
[a, b, q, z, info] = f08ws(compq, compz, ilo, ihi, a, b, q, z, 'n', n)
```

3 Description

nag_lapack_zgghrd (f08ws) is usually the third step in the solution of the complex generalized eigenvalue problem

$$Ax = \lambda Bx.$$

The (optional) first step balances the two matrices using nag_lapack_zggbal (f08wv). In the second step, matrix B is reduced to upper triangular form using the QR factorization function nag_lapack_zgeqrf (f08as) and this unitary transformation Q is applied to matrix A by calling nag_lapack_zunmqr (f08au).

nag_lapack_zgghrd (f08ws) reduces a pair of complex matrices (A, B), where B is triangular, to the generalized upper Hessenberg form using unitary transformations. This two-sided transformation is of the form

$$Q^{\mathrm{H}}AZ = H$$
$$Q^{\mathrm{H}}BZ = T$$

where H is an upper Hessenberg matrix, T is an upper triangular matrix and Q and Z are unitary matrices determined as products of Givens rotations. They may either be formed explicitly, or they may be postmultiplied into input matrices Q_1 and Z_1, so that

$$Q_1 A Z_1^{\mathrm{H}} = (Q_1 Q) H (Z_1 Z)^{\mathrm{H}},$$
$$Q_1 B Z_1^{\mathrm{H}} = (Q_1 Q) T (Z_1 Z)^{\mathrm{H}}.$$

4 References

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

5 Parameters

5.1 Compulsory Input Parameters

1: **compq** – CHARACTER(1)

Specifies the form of the computed unitary matrix Q.

compq = 'N'
Do not compute Q.

compq = 'I'
The unitary matrix Q is returned.

compq = 'V'
q must contain a unitary matrix Q_1, and the product Q_1Q is returned.

Constraint: **compq** = 'N', 'I' or 'V'.

2: **compz** – CHARACTER(1)

Specifies the form of the computed unitary matrix Z.

compz = 'N'
Do not compute Z.

compz = 'V'
z must contain a unitary matrix Z_1, and the product Z_1Z is returned.

compz = 'I'
The unitary matrix Z is returned.

Constraint: **compz** = 'N', 'V' or 'I'.

3: **ilo** – INTEGER
4: **ihi** – INTEGER

i_{lo} and i_{hi} as determined by a previous call to nag_lapack_zggbal (f08wv). Otherwise, they should be set to 1 and n, respectively.

Constraints:

if $\mathbf{n} > 0$, $1 \leq \mathbf{ilo} \leq \mathbf{ihi} \leq \mathbf{n}$;
if $\mathbf{n} = 0$, $\mathbf{ilo} = 1$ and $\mathbf{ihi} = 0$.

5: $\mathbf{a}(lda,:)$ – 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 matrix A of the matrix pair (A,B). Usually, this is the matrix A returned by nag_lapack_zunmqr (f08au).

6: $\mathbf{b}(ldb,:)$ – COMPLEX (KIND=nag_wp) array

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

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

The upper triangular matrix B of the matrix pair (A,B). Usually, this is the matrix B returned by the QR factorization function nag_lapack_zgeqrf (f08as).

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

The first dimension, ldq, of the array **q** must satisfy

if **compq** = 'I' or 'V', $ldq \geq \max(1,\mathbf{n})$;
if **compq** = 'N', $ldq \geq 1$.

The second dimension of the array **q** must be at least $\max(1,\mathbf{n})$ if **compq** = 'I' or 'V' and at least 1 if **compq** = 'N'.

If **compq** = 'V', **q** must contain a unitary matrix Q_1.

If **compq** = 'N', **q** is not referenced.

8: **z**(ldz, :) – COMPLEX (KIND=nag_wp) array

The first dimension, ldz, of the array **z** must satisfy

if **compz** = 'V' or 'I', $ldz \geq \max(1, \mathbf{n})$;
if **compz** = 'N', $ldz \geq 1$.

The second dimension of the array **z** must be at least $\max(1, \mathbf{n})$ if **compz** = 'V' or 'I' and at least 1 if **compz** = 'N'.

If **compz** = 'V', **z** must contain a unitary matrix Z_1.

If **compz** = 'N', **z** is not referenced.

5.2 Optional Input Parameters

1: **n** – INTEGER

Default: the first dimension of the arrays **a**, **b** and the second dimension of the arrays **a**, **b**.

n, the order of the matrices A and B.

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

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 $\max(1, \mathbf{n})$.

a stores the upper Hessenberg matrix H.

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

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

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

b stores the upper triangular matrix T.

3: **q**(ldq, :) – COMPLEX (KIND=nag_wp) array

The first dimension, ldq, of the array **q** will be

if **compq** = 'I' or 'V', $ldq = \max(1, \mathbf{n})$;
if **compq** = 'N', $ldq = 1$.

The second dimension of the array **q** will be $\max(1, \mathbf{n})$ if **compq** = 'I' or 'V' and at least 1 if **compq** = 'N'.

If **compq** = 'I', **q** contains the unitary matrix Q.

Iif **compq** = 'V', **q** stores $Q_1 Q$.

4: **z**(ldz, :) – COMPLEX (KIND=nag_wp) array

The first dimension, ldz, of the array **z** will be

if **compz** = 'V' or 'I', $ldz = \max(1, \mathbf{n})$;
if **compz** = 'N', $ldz = 1$.

The second dimension of the array **z** will be $\max(1, \mathbf{n})$ if **compz** = 'V' or 'I' and at least 1 if **compz** = 'N'.

If **compz** = 'I', **z** contains the unitary matrix Z.

If **compz** = 'V', **z** stores $Z_1 Z$.

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: **compq**, 2: **compz**, 3: **n**, 4: **ilo**, 5: **ihi**, 6: **a**, 7: **lda**, 8: **b**, 9: **ldb**, 10: **q**, 11: **ldq**, 12: **z**, 13: **ldz**, 14: **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

The reduction to the generalized Hessenberg form is implemented using unitary transformations which are backward stable.

8 Further Comments

This function is usually followed by nag_lapack_zhgeqz (f08xs) which implements the QZ algorithm for computing generalized eigenvalues of a reduced pair of matrices.

The real analogue of this function is nag_lapack_dgghrd (f08we).

9 Example

See Section 10 in nag_lapack_zhgeqz (f08xs) and nag_lapack_ztgevc (f08yx).

9.1 Program Text

```
     function f08ws_example

fprintf('f08ws 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 = 'Balance';
[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 = 'No Q';
compz = 'No Z';
z = complex(eye(4));
q = complex(eye(4));
jlo = nag_int(1);
jhi = nag_int(ihi-ilo+1);
[H, T, ~, ~, info] = ...
  f08ws( ...
  compq, compz, jlo, jhi, c, QR, q, z);

% Find eigenvalues of generalized Hessenberg form
%    = eigenvalues of (A,B).
job = 'Eigenvalues';
[~, ~, alpha, beta, ~, ~, info] = ...
  f08xs( ...
  job, compq, compz, jlo, jhi, H, T, q, z);

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

9.2 Program Results

```
     f08ws example results

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