

NAG Toolbox

nag_lapack_dgghrd (f08we)

1 Purpose

`nag_lapack_dgghrd (f08we)` reduces a pair of real matrices (A, B), where B is upper triangular, to the generalized upper Hessenberg form using orthogonal transformations.

2 Syntax

```
[a, b, q, z, info] = nag_lapack_dgghrd(compq, compz, ilo, ihi, a, b, q, z, 'n', n)
[a, b, q, z, info] = f08we(compq, compz, ilo, ihi, a, b, q, z, 'n', n)
```

3 Description

`nag_lapack_dgghrd (f08we)` is the third step in the solution of the real generalized eigenvalue problem

$$Ax = \lambda Bx.$$

The (optional) first step balances the two matrices using `nag_lapack_dggbal (f08wh)`. In the second step, matrix B is reduced to upper triangular form using the QR factorization function `nag_lapack_dgeqr (f08ae)` and this orthogonal transformation Q is applied to matrix A by calling `nag_lapack_dormqr (f08ag)`.

`nag_lapack_dgghrd (f08we)` reduces a pair of real matrices (A, B), where B is upper triangular, to the generalized upper Hessenberg form using orthogonal transformations. This two-sided transformation is of the form

$$\begin{aligned} Q^T A Z &= H \\ Q^T B Z &= T \end{aligned}$$

where H is an upper Hessenberg matrix, T is an upper triangular matrix and Q and Z are orthogonal 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

$$\begin{aligned} Q_1 A Z_1^T &= (Q_1 Q) H (Z_1 Z)^T, \\ Q_1 B Z_1^T &= (Q_1 Q) T (Z_1 Z)^T. \end{aligned}$$

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 orthogonal matrix Q .

compq = 'N'

Do not compute Q .

compq = 'I'

The orthogonal matrix Q is returned.

compq = 'V'

q must contain an orthogonal 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 orthogonal matrix Z .

compz = 'N'

Do not compute Z .

compz = 'I'

The orthogonal matrix Z is returned.

compz = 'V'

z must contain an orthogonal matrix Z_1 , and the product Z_1Z is returned.

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

3: **ilo** – INTEGER4: **ihii** – INTEGER

i_{lo} and i_{hi} as determined by a previous call to nag_lapack_dggbal (f08wh). Otherwise, they should be set to 1 and n , respectively.

Constraints:

if $n > 0$, $1 \leq ilo \leq ihi \leq n$;
 if $n = 0$, **ilo** = 1 and **ihii** = 0.

5: **a**(*lda*,:) – REAL (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 of the matrix pair (A, B) . Usually, this is the matrix A returned by nag_lapack_dormqr (f08ag).

6: **b**(*ldb*,:) – REAL (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 upper triangular matrix B of the matrix pair (A, B) . Usually, this is the matrix B returned by the QR factorization function nag_lapack_dgeqrf (f08ae).

7: **q**(*ldq*,:) – REAL (KIND=nag_wp) array

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

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

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

If **compq** = 'V', **q** must contain an orthogonal matrix Q_1 .

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

8: **$\mathbf{z}(ldz,:)$** – REAL (KIND=nag_wp) array

The first dimension, ldz , of the array **\mathbf{z}** must satisfy

if **compz** = 'V' or 'T', $ldz \geq \max(1, n)$;
 if **compz** = 'N', $ldz \geq 1$.

The second dimension of the array **\mathbf{z}** must be at least $\max(1, n)$ if **compz** = 'V' or 'T' and at least 1 if **compz** = 'N'.

If **compz** = 'V', **\mathbf{z}** must contain an orthogonal matrix Z_1 .

If **compz** = 'N', **\mathbf{z}** is not referenced.

5.2 Optional Input Parameters

1: **\mathbf{n}** – INTEGER

Default: the first dimension of the array **\mathbf{z}** .

n , the order of the matrices A and B .

Constraint: $n \geq 0$.

5.3 Output Parameters

1: **$\mathbf{a}(lda,:)$** – REAL (KIND=nag_wp) array

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

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

\mathbf{a} stores the upper Hessenberg matrix H .

2: **$\mathbf{b}(ldb,:)$** – REAL (KIND=nag_wp) array

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

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

\mathbf{b} stores the upper triangular matrix T .

3: **$\mathbf{q}(ldq,:)$** – REAL (KIND=nag_wp) array

The first dimension, ldq , of the array **\mathbf{q}** will be

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

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

If **compq** = 'I', **\mathbf{q}** contains the orthogonal matrix Q .

If **compq** = 'V', **\mathbf{q}** stores $Q_1 Q$.

4: **$\mathbf{z}(ldz,:)$** – REAL (KIND=nag_wp) array

The first dimension, ldz , of the array **\mathbf{z}** will be

if **compz** = 'V' or 'T', $ldz = \max(1, n)$;
 if **compz** = 'N', $ldz = 1$.

The second dimension of the array **\mathbf{z}** will be $\max(1, n)$ if **compz** = 'V' or 'T' and at least 1 if **compz** = 'N'.

If **compz** = 'T', **\mathbf{z}** contains the orthogonal matrix Z .

If **compz** = 'V', **\mathbf{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 orthogonal transformations which are backward stable.

8 Further Comments

This function is usually followed by nag_lapack_dhgeqz (f08xe) which implements the QZ algorithm for computing generalized eigenvalues of a reduced pair of matrices.

The complex analogue of this function is nag_lapack_zgghrd (f08ws).

9 Example

See Section 10 in nag_lapack_dhgeqz (f08xe) and nag_lapack_dtgevc (f08yk).

9.1 Program Text

```
function f08we_example

fprintf('f08we example results\n\n');

a = [ 1.0    1.0    1.0    1.0    1.0;
      2.0    4.0    8.0   16.0   32.0;
      3.0    9.0   27.0   81.0  243.0;
      4.0   16.0   64.0  256.0 1024.0;
      5.0   25.0  125.0 625.0 3125.0];
b = a';

%' Balance A and B
job = 'B';
[a, b, ilo, ihi, lscale, rscale, info] = ...
f08wh( ...
job, a, b);

bbal = b(ilo:ihi,ilo:ihi);
abal = a(ilo:ihi,ilo:ihi);

% QR factorize balanced B
[QR, tau, info] = f08ae(bbal);

% Perform C = Q^T * A
side = 'Left';
trans = 'Transpose';
[c, info] = f08ag( ...
side, trans, QR, tau, abal);

% Generalized Hessenberg form (C,R) -> (H,T)
```

```

compq = 'No Q';
compz = 'No Z';
z = eye(4);
q = eye(4);
jlo = nag_int(1);
jhi = nag_int(ihi-ilo+1);
[H, T, ~, ~, info] = ...
f08we( ...
    compq, compz, jlo, jhi, c, QR, q, z);

% Find eigenvalues of generalized Hessenberg form
%   = eigenvalues of (A,B).
job = 'Eigenvalues';
[~, ~, alphar, alphai, beta, ~, ~, info] = ...
f08xe( ...
    job, compq, compz, jlo, jhi, H, T, q, z);

disp('Generalized eigenvalues of (A,B):');
w = complex(alphar+i*alphai);
disp(w./beta);

```

9.2 Program Results

f08we example results

```

Generalized eigenvalues of (A,B):
-2.4367 + 0.0000i
 0.6069 + 0.7948i
 0.6069 - 0.7948i
 1.0000 + 0.0000i
-0.4104 + 0.0000i

```
