NAG Toolbox

nag_lapack_dorghr (f08nf)

1 Purpose

nag_lapack_dorghr (f08nf) generates the real orthogonal matrix Q which was determined by nag_lapack_dgehrd (f08ne) when reducing a real general matrix A to Hessenberg form.

2 Syntax

```
[a, info] = nag_lapack_dorghr(ilo, ihi, a, tau, 'n', n)
[a, info] = f08nf(ilo, ihi, a, tau, 'n', n)
```

3 Description

nag_lapack_dorghr (f08nf) is intended to be used following a call to nag_lapack_dgehrd (f08ne), which reduces a real general matrix A to upper Hessenberg form H by an orthogonal similarity transformation: $A = QHQ^{\mathrm{T}}$. nag_lapack_dgehrd (f08ne) represents the matrix Q as a product of $i_{\mathrm{hi}} - i_{\mathrm{lo}}$ elementary reflectors. Here i_{lo} and i_{hi} are values determined by nag_lapack_dgebal (f08nh) when balancing the matrix; if the matrix has not been balanced, $i_{\mathrm{lo}} = 1$ and $i_{\mathrm{hi}} = n$.

This function may be used to generate Q explicitly as a square matrix. Q has the structure:

$$Q = \begin{pmatrix} I & 0 & 0 \\ 0 & Q_{22} & 0 \\ 0 & 0 & I \end{pmatrix}$$

where Q_{22} occupies rows and columns i_{lo} to i_{hi}.

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: **ilo** – INTEGER
2: **ihi** – INTEGER

These **must** be the same arguments **ilo** and **ihi**, respectively, as supplied to nag_lapack_dgehrd (f08ne).

Constraints:

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

3: $\mathbf{a}(lda,:)$ – REAL (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 $\max(1,\mathbf{n})$.

Details of the vectors which define the elementary reflectors, as returned by nag_lapack_dgehrd (f08ne).

4: **tau**(:) – REAL (KIND=nag_wp) array

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

Further details of the elementary reflectors, as returned by nag_lapack_dgehrd (f08ne).

5.2 Optional Input Parameters

1: **n** – INTEGER

Default: the first dimension of the array **a** and the second dimension of the array **a**.

n, the order of the matrix Q.

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

5.3 Output Parameters

1: **a**(lda, :) – REAL (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 $\max(1, \mathbf{n})$.

The n by n orthogonal matrix Q.

2: **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: **n**, 2: **ilo**, 3: **ihi**, 4: **a**, 5: **lda**, 6: **tau**, 7: **work**, 8: **lwork**, 9: **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 computed matrix Q differs from an exactly orthogonal matrix by a matrix E such that

$$\|E\|_2 = O(\epsilon),$$

where ϵ is the ***machine precision***.

8 Further Comments

The total number of floating-point operations is approximately $\frac{4}{3}q^3$, where $q = i_{\mathrm{hi}} - i_{\mathrm{lo}}$.

The complex analogue of this function is nag_lapack_zunghr (f08nt).

9 Example

This example computes the Schur factorization of the matrix A, where

$$A = \begin{pmatrix} 0.35 & 0.45 & -0.14 & -0.17 \\ 0.09 & 0.07 & -0.54 & 0.35 \\ -0.44 & -0.33 & -0.03 & 0.17 \\ 0.25 & -0.32 & -0.13 & 0.11 \end{pmatrix}.$$

Here A is general and must first be reduced to Hessenberg form by nag_lapack_dgehrd (f08ne). The program then calls nag_lapack_dorghr (f08nf) to form Q, and passes this matrix to nag_lapack_dhseqr (f08pe) which computes the Schur factorization of A.

9.1 Program Text

```
     function f08nf_example

fprintf('f08nf example results\n\n');

ilo = nag_int(1);
ihi = nag_int(4);
a = [ 0.35,  0.45, -0.14, -0.17;
      0.09,  0.07, -0.54,  0.35;
     -0.44, -0.33, -0.03,  0.17;
      0.25, -0.32, -0.13,  0.11];

% Reduce A to upper Hessenberg Form A = QHQ^T
[H, tau, info] = f08ne( ...
                        ilo, ihi, a);

% Form Q
[Q, info] = f08nf( ...
                   ilo, ihi, H, tau);

% Schur factorize H = Y*T*Y' and form Z = QY  A = QY*T*(QQY)'
job   = 'Schur form';
compz = 'Vectors';
[~, wr, wi, Z, info] = f08pe( ...
                              job, compz, ilo, ihi, H, Q);

w = wr + i*wi;
disp('Eigenvalues of A');
disp(w);
```

9.2 Program Results

```
     f08nf example results

Eigenvalues of A
   0.7995 + 0.0000i
  -0.0994 + 0.4008i
  -0.0994 - 0.4008i
  -0.1007 + 0.0000i
```