

NAG Toolbox

nag_lapack_zunghr (f08nt)

1 Purpose

nag_lapack_zunghr (f08nt) generates the complex unitary matrix Q which was determined by nag_lapack_zgehrd (f08ns) when reducing a complex general matrix A to Hessenberg form.

2 Syntax

```
[a, info] = nag_lapack_zunghr(ilo, ihi, a, tau, 'n', n)
[a, info] = f08nt(ilo, ihi, a, tau, 'n', n)
```

3 Description

nag_lapack_zunghr (f08nt) is intended to be used following a call to nag_lapack_zgehrd (f08ns), which reduces a complex general matrix A to upper Hessenberg form H by a unitary similarity transformation: $A = QHQ^H$. nag_lapack_zgehrd (f08ns) represents the matrix Q as a product of $i_{\text{hi}} - i_{\text{lo}}$ elementary reflectors. Here i_{lo} and i_{hi} are values determined by nag_lapack_zgebal (f08nv) when balancing the matrix; if the matrix has not been balanced, $i_{\text{lo}} = 1$ and $i_{\text{hi}} = n$.

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_zgehrd (f08ns).

Constraints:

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

- 3: **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**).

Details of the vectors which define the elementary reflectors, as returned by nag_lapack_zgehrd (f08ns).

4: **tau**(:) – COMPLEX (KIND=nag_wp) array
 The dimension of the array **tau** must be at least max(1, **n** – 1)
 Further details of the elementary reflectors, as returned by nag_lapack_zgehrd (f08ns).

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: **n** ≥ 0 .

5.3 Output Parameters

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

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

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

The *n* by *n* unitary 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 unitary matrix by a matrix E such that

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

where ϵ is the *machine precision*.

8 Further Comments

The total number of real floating-point operations is approximately $\frac{16}{3}q^3$, where $q = i_{\text{hi}} - i_{\text{lo}}$.

The real analogue of this function is nag_lapack_dorgrh (f08nf).

9 Example

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

$$A = \begin{pmatrix} -3.97 - 5.04i & -4.11 + 3.70i & -0.34 + 1.01i & 1.29 - 0.86i \\ 0.34 - 1.50i & 1.52 - 0.43i & 1.88 - 5.38i & 3.36 + 0.65i \\ 3.31 - 3.85i & 2.50 + 3.45i & 0.88 - 1.08i & 0.64 - 1.48i \\ -1.10 + 0.82i & 1.81 - 1.59i & 3.25 + 1.33i & 1.57 - 3.44i \end{pmatrix}.$$

Here A is general and must first be reduced to Hessenberg form by nag_lapack_zgehrd (f08ns). The program then calls nag_lapack_zunghr (f08nt) to form Q , and passes this matrix to nag_lapack_zhseqr (f08ps) which computes the Schur factorization of A .

9.1 Program Text

```
function f08nt_example

fprintf('f08nt example results\n\n');

ilo = nag_int(1);
ihi = nag_int(4);
a = [ -3.97 - 5.04i, -4.11 + 3.70i, -0.34 + 1.01i, 1.29 - 0.86i;
      0.34 - 1.50i, 1.52 - 0.43i, 1.88 - 5.38i, 3.36 + 0.65i;
      3.31 - 3.85i, 2.50 + 3.45i, 0.88 - 1.08i, 0.64 - 1.48i;
      -1.10 + 0.82i, 1.81 - 1.59i, 3.25 + 1.33i, 1.57 - 3.44i];

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

% Form Q
[Q, info] = f08nt(ilo, ihi, H, tau);

% Schur factorize H = Y*T*Y' and form Z = QY A = QY*T*(QQY)'
job = 'Schur form';
compz = 'Vectors';
[~, w, z, info] = f08ps( ...
    job, compz, ilo, ihi, H, Q);

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

9.2 Program Results

```
f08nt example results

Eigenvalues of A
-6.0004 - 6.9998i
-5.0000 + 2.0060i
 7.9982 - 0.9964i
 3.0023 - 3.9998i
```
