NAG Toolbox for MATLAB

f08nu

1 Purpose

f08nu multiplies an arbitrary complex matrix C by the complex unitary matrix Q which was determined by f08ns when reducing a complex general matrix to Hessenberg form.

2 Syntax

```
[c, info] = f08nu(side, trans, ilo, ihi, a, tau, c, 'm', m, 'n', n)
```

3 Description

f08nu is intended to be used following a call to f08ns, which reduces a complex general matrix A to upper Hessenberg form H by a unitary similarity transformation: $A = QHQ^{\mathrm{H}}$. f08ns 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 f08nv 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 form one of the matrix products

$$QC, Q^{\mathrm{H}}C, CQ \text{ or } CQ^{\mathrm{H}},$$

overwriting the result on C (which may be any complex rectangular matrix).

A common application of this function is to transform a matrix V of eigenvectors of H to the matrix QV of eigenvectors of A.

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: **side – string**

Indicates how Q or Q^{H} is to be applied to C.

side = 'L'

Q or Q^{H} is applied to C from the left.

side = 'R'

Q or Q^{H} is applied to C from the right.

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

2: **trans – string**

Indicates whether Q or Q^{H} is to be applied to C.

trans = 'N'

Q is applied to C.

trans = 'C'

Q^{H} is applied to C.

Constraint: **trans** = 'N' or 'C'.

3: **ilo – int32 scalar**

4: **ihi – int32 scalar**

These **must** be the same parameters **ilo** and **ihi**, respectively, as supplied to f08ns.

Constraints:

if **side** = 'L' and $\mathbf{m} > 0$, $1 \leq \mathbf{ilo} \leq \mathbf{ihi} \leq \mathbf{m}$;
if **side** = 'L' and $\mathbf{m} = 0$, $\mathbf{ilo} = 1$ and $\mathbf{ihi} = 0$;
if **side** = 'R' and $\mathbf{n} > 0$, $1 \leq \mathbf{ilo} \leq \mathbf{ihi} \leq \mathbf{n}$;
if **side** = 'R' and $\mathbf{n} = 0$, $\mathbf{ilo} = 1$ and $\mathbf{ihi} = 0$.

5: **a(lda,∗) – complex array**

The first dimension, **lda**, of the array **a** must satisfy

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

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

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

6: **tau(∗) – complex array**

Note: the dimension of the array **tau** must be at least $\max(1, \mathbf{m} - 1)$ if **side** = 'L' and at least $\max(1, \mathbf{n} - 1)$ if **side** = 'R'.

Further details of the elementary reflectors, as returned by f08ns.

7: **c(ldc,∗) – complex array**

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

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

The m by n matrix C.

5.2 Optional Input Parameters

1: **m – int32 scalar**

Default: The first dimension of the array **c**.

m, the number of rows of the matrix C; m is also the order of Q if **side** = 'L'.

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

2: **n – int32 scalar**

Default: The second dimension of the array **c**.

n, the number of columns of the matrix C; n is also the order of Q if **side** = 'R'.

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

5.3 Input Parameters Omitted from the MATLAB Interface

lda, ldc, work, lwork

5.4 Output Parameters

1: **c**(**ldc**,∗) **– complex array**

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

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

c contains QC or $Q^{\mathrm{H}}C$ or CQ or CQ^{H} as specified by **side** and **trans**.

2: **info – int32 scalar**

info $= 0$ unless the function detects an error (see Section 6).

6 Error Indicators and Warnings

Errors or warnings detected by the function:

info $= -i$

If **info** $= -i$, parameter i had an illegal value on entry. The parameters are numbered as follows:

1: **side**, 2: **trans**, 3: **m**, 4: **n**, 5: **ilo**, 6: **ihi**, 7: **a**, 8: **lda**, 9: **tau**, 10: **c**, 11: **ldc**, 12: **work**, 13: **lwork**, 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 computed result differs from the exact result by a matrix E such that

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

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

8 Further Comments

The total number of real floating-point operations is approximately $8nq^2$ if **side** = 'L' and $8mq^2$ if **side** = 'R', where $q = i_{\mathrm{hi}} - i_{\mathrm{lo}}$.

The real analogue of this function is f08ng.

9 Example

```
side = 'Left';
trans = 'No transpose';
ilo = int32(1);
ihi = int32(4);
a    =    [complex(-3.97,    -5.04),    complex(-1.131805187339771,    -
2.56930489882744), ...
           complex(-4.602742437533554,   -0.142631904083292),   complex(-
1.424912289366528, +1.732983703342187);
         complex(-5.479653273702635,  +0),  complex(1.858472820765587,  -
1.55018070644029), ...
           complex(4.414465526917012,   -0.7638237115550983),   complex(-
0.4805261336990153, -1.197599997332747);
                    complex(0.6932222118146283,    -0.4828752762602551),
```

```
complex(6.267276818064224, ...
      +0), complex(-0.4503809403345012, -0.02898183259817966), complex(-
1.346684450078734, +1.65792489538873);
                    complex(-0.2112946907920694,    +0.0864412259893682),
complex(0.1242146188766495, ...
              -0.2289276049796828),    complex(-3.499985837393258,    +0),
complex(2.561908119568915, -3.370837460961531)];
tau = [complex(1.062047721455606, -0.2737399475982613);
     complex(1.805921371640585, +0.3479067029848286);
     complex(1.181823471041003, +0.9833311880432766)];
c       =       [complex(0.9999999999999999,       -6.374287734989295e-17),
complex(0.2613040867512706, +0.5283875825457147);
            complex(0.3338962373760938,  -0.3906135999457942),  complex(-
0.6020569110082319, +0.2903496482826593);
                    complex(0.08840188787511427,    +0.3310704781636432),
complex(0.9999999999999999, +1.120737926335775e-17);
                    complex(0.07925597470461895,    +0.1017388061223024),
complex(0.3074174862711929, +0.2185870109734625)];
[cOut, info] = f08nu(side, trans, ilo, ihi, a, tau, c)
```

```
cOut =
   1.0000 - 0.0000i   0.2613 + 0.5284i
  -0.0210 + 0.3590i   0.6485 + 0.4683i
   0.1035 + 0.3683i  -0.0323 - 0.8516i
  -0.0664 - 0.3436i  -0.4521 + 0.1368i
info =
           0
```