NAG Toolbox for MATLAB

f08gu

1 Purpose

f08gu multiplies an arbitrary complex matrix C by the complex unitary matrix Q which was determined by f08gs when reducing a complex Hermitian matrix to tridiagonal form.

2 Syntax

```
[ap, c, info] = f08gu(side, uplo, trans, ap, tau, c, 'm', m, 'n', n)
```

3 Description

f08gu is intended to be used after a call to f08gs, which reduces a complex Hermitian matrix A to real symmetric tridiagonal form T by a unitary similarity transformation: $A = QTQ^{\mathrm{H}}$. f08gs represents the unitary matrix Q as a product of elementary reflectors.

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 Z of eigenvectors of T to the matrix QZ 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: **uplo – string**

This **must** be the same parameter **uplo** as supplied to f08gs.

Constraint: **uplo** = 'U' or 'L'.

3: **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'.

4: **ap**$(*)$ **– complex array**

Note: the dimension of the array **ap** must be at least $\max(1, \mathbf{m} \times (\mathbf{m}+1)/2)$ if **side** = 'L' and at least $\max(1, \mathbf{n} \times (\mathbf{n}+1)/2)$ if **side** = 'R'.

Details of the vectors which define the elementary reflectors, as returned by f08gs.

5: **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 f08gs.

6: **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

ldc, work

5.4 Output Parameters

1: **ap**$(*)$ **– complex array**

Note: the dimension of the array **ap** must be at least $\max(1, \mathbf{m} \times (\mathbf{m}+1)/2)$ if **side** = 'L' and at least $\max(1, \mathbf{n} \times (\mathbf{n}+1)/2)$ if **side** = 'R'.

Is used as internal workspace prior to being restored and hence is unchanged.

2: **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**.

3: **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: **uplo**, 3: **trans**, 4: **m**, 5: **n**, 6: **ap**, 7: **tau**, 8: **c**, 9: **ldc**, 10: **work**, 11: **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 $8m^2n$ if **side** = 'L' and $8mn^2$ if **side** = 'R'.

The real analogue of this function is f08gg.

9 Example

```
side = 'Left';
uplo = 'L';
trans = 'No transpose';
ap = [complex(-2.28, +0);
     complex(-4.33845594653213, +0);
     complex(0.3278606760921924, -0.1251226092264437);
     complex(-0.1412565637506947, -0.366636483973957);
     complex(-0.1284569816493291, +0);
     complex(-2.022594578622617, +0);
     complex(-0.308321908008089, +0.1763226364726777);
     complex(-0.1665932537524081, +0);
     complex(-1.802322978338735, +0);
     complex(-1.924949764598263, +0)];
tau = [complex(1.410284216766754, +0.4679084045148932);
     complex(1.302420369434775, +0.7853320742529579);
     complex(1.093973715923082, -0.9955746786231597)];
c = [complex(0.7298945743917051, +0), complex(-0.2595449733877608, +0);
     complex(0.6258777805557931, +0), complex(-0.04325496258655371, +0);
     complex(0.2513449473644084, +0), complex(0.495247410182068, +0);
     complex(0.1111603864444915, +0), complex(0.8279465065502341, +0)];
[apOut, cOut, info] = f08gu(side, uplo, trans, ap, tau, c)
```

```
apOut =
  -2.2800
  -4.3385
   0.3279 - 0.1251i
```

```
  -0.1413 - 0.3666i
  -0.1285
  -2.0226
  -0.3083 + 0.1763i
  -0.1666
  -1.8023
  -1.9249
cOut =
   0.7299            -0.2595
  -0.1663 - 0.2061i   0.5969 + 0.4214i
  -0.4165 - 0.1417i  -0.2965 - 0.1507i
   0.1743 + 0.4162i   0.3482 + 0.4085i
info =
           0
```