NAG Toolbox for MATLAB

f08ax

1 Purpose

f08ax multiplies an arbitrary complex matrix C by the complex unitary matrix Q from an LQ factorization computed by f08av.

2 Syntax

```
[c, info] = f08ax(side, trans, a, tau, c, 'm', m, 'n', n, 'k', k)
```

3 Description

f08ax is intended to be used after a call to f08av, which performs an LQ factorization of a complex matrix A. The unitary matrix Q is represented 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).

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: **a**(**lda**,$*$) **– complex array**

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

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 f08av.

4: **tau**$(*)$ – **complex array**

Note: the dimension of the array **tau** must be at least $\max(1, \mathbf{k})$.

Further details of the elementary reflectors as returned by f08av.

5: **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.

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.

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

3: **k** – **int32 scalar**

Default: The dimension of the array **tau**.

k, the number of elementary reflectors whose product defines the matrix Q.

Constraints:

if **side** = 'L', $\mathbf{m} \geq \mathbf{k} \geq 0$;
if **side** = 'R', $\mathbf{n} \geq \mathbf{k} \geq 0$.

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: **k**, 6: **a**, 7: **lda**, 8: **tau**, 9: **c**, 10: **ldc**, 11: **work**, 12: **lwork**, 13: **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 $8nk(2m-k)$ if **side** = 'L' and $8mk(2n-k)$ if **side** = 'R'.

The real analogue of this function is f08ak.

9 Example

```
side = 'Left';
trans = 'Conjugate transpose';
a  =  [complex(-2.225511177235469,  -0),  complex(0.2438442594612761,  -
0.3082069932914802), ...
                    complex(-0.2741364985191491,     -0.230966496866882),
complex(0.5807709514701289, +0.3468663602164287);
                     complex(0.8207552577961005,    +1.238457046719376),
complex(1.688100989346069, ...
             -0),    complex(-0.1936415258449461,   +0.5429517855236941),
complex(0.2789084851242076, -0.2203175797458332);
                   complex(0.001033470433007241,    -0.6822252862760427),
complex(0.774751330164006, ...
       -0.6154727158531147),  complex(-1.590258250459319,  -0),  complex(-
0.1267668516113225, +0.1109845202357172)];
tau = [complex(1.12581379184436, +0.1617605895141772);
     complex(1.099053669168954, +0.5468590598466785);
     complex(1.132925657311579, -0.9590540478961487)];
c   =   [complex(0.6066022106781648,   -0.08537364446581577),   complex(-
2.170287909315212, +1.199724372230148);
             complex(5.216747404439203,   -2.512397031673043),   complex(-
2.377178681793996, +2.987456497043377);
                     complex(6.293162138826466,    -7.861098176652019),
complex(0.1214991293563571, +0.458724036761386);
     complex(0, +0), complex(0, +0)];
[cOut, info] = f08ax(side, trans, a, tau, c)
```

```
cOut =
  -2.8501 + 6.4683i  -1.1682 - 1.8886i
   1.6264 - 0.7799i   2.8377 + 0.7654i
   6.9290 + 4.6481i  -1.7610 - 0.7041i
   1.4048 + 3.2400i   1.0518 - 1.6365i
info =
           0
```