NAG Toolbox for MATLAB

f08ak

1 Purpose

f08ak multiplies an arbitrary real matrix C by the real orthogonal matrix Q from an LQ factorization computed by f08ah.

2 Syntax

```
[c, info] = f08ak(side, trans, a, tau, c, 'm', m, 'n', n, 'k', k)
```

3 Description

f08ak is intended to be used after a call to f08ah, which performs an LQ factorization of a real matrix A. The orthogonal matrix Q is represented as a product of elementary reflectors.

This function may be used to form one of the matrix products

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

overwriting the result on C (which may be any real 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^{T} is to be applied to C.

side = 'L'

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

side = 'R'

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

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

2: **trans – string**

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

trans = 'N'

Q is applied to C.

trans = 'T'

Q^{T} is applied to C.

Constraint: **trans** = 'N' or 'T'.

3: **a**(**lda**,$*$) **– double 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 f08ah.

4: **tau**$(*)$ – **double 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 f08ah.

5: **c**(**ldc**,$*$) – **double 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**,$*$) – **double 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 $Q^{\mathrm{T}}C$ or CQ or CQ^{T} 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 floating-point operations is approximately $2nk(2m-k)$ if **side** = 'L' and $2mk(2n-k)$ if **side** = 'R'.

The complex analogue of this function is f08ax.

9 Example

```
side = 'Left';
trans = 'Transpose';
a   =   [7.629239804856051,   -0.2513556382632653,   0.2820087648807367,   -
0.02069086046679318, ...
     -0.1578636020799776, -0.03525109561009208;
            0.1206804378352315,   6.484792689972742,   0.2614412741025646,
0.1032576018728744, ...
     0.4200952042988654, 0.007010589007406895;
           0.1021464811610684,   -1.661861465322362,   -5.426581112124191,
0.605137487741764, ...
     -0.5386683512562429, 0.1685937776381602;
            1.476634669791005,   0.1088236918688469,   0.4222885621904238,
6.255531937917222, ...
     -0.1704479056684627, -0.3498629724431616];
tau = [1.710424647623495;
     1.592936535590086;
     1.187099434147982;
     1.736930441655314];
c = [-0.3761842691290461, -0.6855204625592025;
     0.2583580019290119, 0.05747738245248697;
     0.5624570664128612, -0.9076696447114656;
     0.1182716488230685, -1.122317523043419;
     0, 0;
     0, 0];
[cOut, info] = f08ak(side, trans, a, tau, c)
```

```
cOut =
    0.2371    0.7383
   -0.4575    0.0158
   -0.0085   -0.0161
   -0.5192    1.0768
    0.0239   -0.6436
   -0.0543   -0.6613
info =
```

0