NAG Toolbox for MATLAB

f08aj

1 Purpose

f08aj generates all or part of the real orthogonal matrix Q from an LQ factorization computed by f08ah.

2 Syntax

```
[a, info] = f08aj(a, tau, 'm', m, 'n', n, 'k', k)
```

3 Description

f08aj 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 generate Q explicitly as a square matrix, or to form only its leading rows.

Usually Q is determined from the LQ factorization of a p by n matrix A with $p \leq n$. The whole of Q may be computed by:

```
[a, info] = f08aj(a, tau);
```

(note that the array **a** must have at least n rows) or its leading p rows by:

```
[a, info] = f08aj(a(1:p,:), tau);
```

The rows of Q returned by the last call form an orthonormal basis for the space spanned by the rows of A; thus f08ah followed by f08aj can be used to orthogonalise the rows of A.

The information returned by the LQ factorization functions also yields the LQ factorization of the leading k rows of A, where $k < p$. The orthogonal matrix arising from this factorization can be computed by:

```
[a, info] = f08aj(a, tau);
```

or its leading k rows by:

```
[a, info] = f08aj(a(1:k,:), tau);
```

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

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

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

Details of the vectors which define the elementary reflectors, as returned by f08ah.

2: **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.

5.2 Optional Input Parameters

1: **m – int32 scalar**

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

m, the number of rows of the matrix Q.

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

2: **n – int32 scalar**

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

n, the number of columns of the matrix Q.

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

3: **k – int32 scalar**

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

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

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

5.3 Input Parameters Omitted from the MATLAB Interface

lda, work, lwork

5.4 Output Parameters

1: **a(lda,$*$) – double array**

The first dimension of the array **a** 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 Q.

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: **m**, 2: **n**, 3: **k**, 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 orthogonal matrix by a matrix E such that

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

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

8 Further Comments

The total number of floating-point operations is approximately $4mnk - 2(m+n)k^2 + \frac{4}{3}k^3$; when $m = k$, the number is approximately $\frac{2}{3}m^2(3n-m)$.

The complex analogue of this function is f08aw.

9 Example

```
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];
[aOut, info] = f08aj(a, tau)
```

```
aOut =
   -0.7104    0.4299   -0.4824    0.0354    0.2700    0.0603
   -0.2412   -0.5323   -0.4845   -0.1595   -0.6311   -0.0027
    0.1287   -0.2619   -0.2108   -0.7447    0.5227   -0.2063
   -0.3403   -0.0921    0.4546   -0.3869   -0.0465    0.7191
info =
           0
```