NAG Toolbox

nag_lapack_dormqr (f08ag)

1 Purpose

nag_lapack_dormqr (f08ag) multiplies an arbitrary real matrix C by the real orthogonal matrix Q from a QR factorization computed by nag_lapack_dgeqrf (f08ae), nag_lapack_dgeqpf (f08be) or nag_lapack_dgeqp3 (f08bf).

2 Syntax

```
[c, info] = nag_lapack_dormqr(side, trans, a, tau, c, 'm', m, 'n', n, 'k', k)
[c, info] = f08ag(side, trans, a, tau, c, 'm', m, 'n', n, 'k', k)
```

3 Description

nag_lapack_dormqr (f08ag) is intended to be used after a call to nag_lapack_dgeqrf (f08ae), nag_lapack_dgeqpf (f08be) or nag_lapack_dgeqp3 (f08bf) which perform a QR 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).

A common application of this function is in solving linear least squares problems, as described in the F08 Chapter Introduction and illustrated in Section 10 in nag_lapack_dgeqrf (f08ae).

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** – CHARACTER(1)

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** – CHARACTER(1)

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, :) – REAL (KIND=nag_wp) array

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

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

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

Details of the vectors which define the elementary reflectors, as returned by nag_lapack_dgeqrf (f08ae), nag_lapack_dgeqpf (f08be) or nag_lapack_dgeqp3 (f08bf).

4: **tau**(:) – REAL (KIND=nag_wp) array

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

Further details of the elementary reflectors, as returned by nag_lapack_dgeqrf (f08ae), nag_lapack_dgeqpf (f08be) or nag_lapack_dgeqp3 (f08bf).

5: **c**(ldc, :) – REAL (KIND=nag_wp) array

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

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

The m by n matrix C.

5.2 Optional Input Parameters

1: **m** – INTEGER

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

m, the number of rows of the matrix C.

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

2: **n** – INTEGER

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

n, the number of columns of the matrix C.

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

3: **k** – INTEGER

Default: the second dimension of the arrays **a**, **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 Output Parameters

1: **c**(ldc, :) – REAL (KIND=nag_wp) array

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

The second dimension of the array **c** will be $\max(1, \mathbf{n})$.

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

2: **info** – INTEGER

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

6 Error Indicators and Warnings

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 nag_lapack_zunmqr (f08au).

9 Example

See Section 10 in nag_lapack_dgeqrf (f08ae).

9.1 Program Text

```
function f08ag_example

fprintf('f08ag example results\n\n');

a = [-0.57, -1.28, -0.39,  0.25;
     -1.93,  1.08, -0.31, -2.14;
      2.30,  0.24,  0.40, -0.35;
     -1.93,  0.64, -0.66,  0.08;
      0.15,  0.30,  0.15, -2.13;
     -0.02,  1.03, -1.43,  0.50];

% Compute the QR Factorisation of A
[a, tau, info] = f08ae(a);

% Apply Q^T to C from Left, i.e. B = Q^T*C
side = 'Left';
trans = 'Transpose';
c = [-2.67,  0.41;
     -0.55, -3.10;
      3.34, -4.01;
     -0.77,  2.76;
      0.48, -6.17;
```

```
      4.10,  0.21];
[b, info] = f08ag(side, trans, a, tau, c);

disp('B = Q^T*C:');
disp(b);
```

9.2 Program Results

```
     f08ag example results

B = Q^T*C:
    3.2456   -2.6896
   -4.5885    3.0574
   -2.1500   -2.3696
   -0.0931   -7.0279
   -0.0085    0.0009
    0.0204    0.0138
```