NAG Toolbox

nag_lapack_dormql (f08cg)

1 Purpose

nag_lapack_dormql (f08cg) multiplies a general real m by n matrix C by the real orthogonal matrix Q from a QL factorization computed by nag_lapack_dgeqlf (f08ce).

2 Syntax

```
[c, info] = nag_lapack_dormql(side, trans, a, tau, c, 'm', m, 'n', n, 'k', k)
[c, info] = f08cg(side, trans, a, tau, c, 'm', m, 'n', n, 'k', k)
```

3 Description

nag_lapack_dormql (f08cg) is intended to be used following a call to nag_lapack_dgeqlf (f08ce), which performs a QL factorization of a real matrix A and represents the orthogonal matrix Q as a product of elementary reflectors.

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

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

overwriting the result on C, which may be any real rectangular m by n 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_dgeqlf (f08ce).

4 References

Anderson E, Bai Z, Bischof C, Blackford S, Demmel J, Dongarra J J, Du Croz J J, Greenbaum A, Hammarling S, McKenney A and Sorensen D (1999) *LAPACK Users' Guide* (3rd Edition) SIAM, Philadelphia http://www.netlib.org/lapack/lug

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_dgeqlf (f08ce).

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_dgeqlf (f08ce).

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_zunmql (f08cu).

9 Example

See Section 10 in nag_lapack_dgeqlf (f08ce).

9.1 Program Text

```
function f08cg_example

fprintf('f08cg example results\n\n');

a = [-0.57, -1.28, -0.39,  0.25;
     -1.93,  1.08, -0.31, -2.14;
      2.3,   0.24,  0.4,  -0.35;
     -1.93,  0.64, -0.66,  0.08;
      0.15,  0.3,   0.15, -2.13;
     -0.02,  1.03, -1.43,  0.5];
b = [-2.67,  0.41;
     -0.55, -3.10;
      3.34, -4.01;
     -0.77,  2.76;
      0.48, -6.17;
      4.10,  0.21];
% Compute the QL factorization of a
[a, tau, info] = f08ce(a);

% Compute C = (C1) = (Q^T)*b
%             (C2)
[c, info] = f08cg( ...
    'Left', 'Transpose', a, tau, b);

% Compute least-squares solutions by backsubstitution in L*X = C2
[b, info] = f07te( ...
    'Lower', 'No Transpose', 'Non-Unit', a(3:6,:), c(3:6,:));
```

```
if (info > 0)
  fprintf('The lower triangular factor, L, of A is singular,\n');
  fprintf('the least squares solution could not be computed.\n');
else
  % Print least-squares solutions
  [ifail] = x04ca( ...
    'General', ' ', b, 'Least-squares solution(s)');
  % Compute and print estimates of the square roots of the residual
  % sums of squares
  rnorm = zeros(2,1);
  for j=1:2
    rnorm(j) = norm(c(1:2,j));
  end
  fprintf('\nSquare root(s) of the residual sum(s) of squares\n');
  fprintf('\t%11.2e    %11.2e\n', rnorm(1), rnorm(2));
end
```

9.2 Program Results

```
     f08cg example results

 Least-squares solution(s)
             1          2
 1      1.5339    -1.5753
 2      1.8707     0.5559
 3     -1.5241     1.3119
 4      0.0392     2.9585

Square root(s) of the residual sum(s) of squares
    2.22e-02       1.38e-02
```