NAG Fortran Library Routine Document

F08CGF (DORMQL)

Note: before using this routine, please read the Users' Note for your implementation to check the interpretation of ***bold italicised*** terms and other implementation-dependent details.

1 Purpose

F08CGF (DORMQL) multiplies a general real m by n matrix C by the real orthogonal matrix Q from a QL factorization computed by F08CEF (DGEQLF).

2 Specification

```
SUBROUTINE F08CGF (SIDE, TRANS, M, N, K, A, LDA, TAU, C, LDC, WORK,
1                  LWORK, INFO)
INTEGER            M, N, K, LDA, LDC, LWORK, INFO
double precision   A(LDA,*), TAU(*), C(LDC,*), WORK(*)
CHARACTER*1        SIDE, TRANS
```

The routine may be called by its LAPACK name ***dormql***.

3 Description

F08CGF (DORMQL) is intended to be used following a call to F08CEF (DGEQLF), which performs a QL factorization of a real matrix A and represents the orthogonal matrix Q as a product of elementary reflectors.

This routine 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 routine is in solving linear least squares problems, as described in the F08 Chapter Introduction, and illustrated in Section 9 of the document for F08CEF (DGEQLF).

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 URL: http://www.netlib.org/lapack/lug

5 Parameters

1: SIDE – CHARACTER*1 *Input*

On entry: 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 *Input*

On entry: 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: M – INTEGER *Input*

On entry: m, the number of rows of the matrix C.

Constraint: $\mathrm{M} \geq 0$.

4: N – INTEGER *Input*

On entry: n, the number of columns of the matrix C.

Constraint: $\mathrm{N} \geq 0$.

5: K – INTEGER *Input*

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

Constraints:

if SIDE = 'L', $\mathrm{M} \geq \mathrm{K} \geq 0$;
if SIDE = 'R', $\mathrm{N} \geq \mathrm{K} \geq 0$.

6: A(LDA,$*$) – ***double precision*** array *Input*

Note: the second dimension of the array A must be at least $\max(1, \mathrm{K})$.

On entry: details of the vectors which define the elementary reflectors as returned by F08CEF (DGEQLF) and stored in the $(n-k+i)$th columns of A, for $i = 1, 2, \ldots, k$.

7: LDA – INTEGER *Input*

On entry: the first dimension of the array A as declared in the (sub)program from which F08CGF (DORMQL) is called.

Constraints:

if SIDE = 'L', $\mathrm{LDA} \geq \max(1, \mathrm{M})$;
if SIDE = 'R', $\mathrm{LDA} \geq \max(1, \mathrm{N})$.

8: TAU($*$) – ***double precision*** array *Input*

Note: the dimension of the array TAU must be at least $\max(1, \mathrm{K})$.

On entry: further details of the elementary reflectors as returned by F08CEF (DGEQLF).

9: C(LDC,$*$) – ***double precision*** array *Input/Output*

Note: the second dimension of the array C must be at least $\max(1, \mathrm{N})$.

On entry: the m by n matrix C.

On exit: is overwritten by QC or $Q^{\mathrm{T}}C$ or CQ or CQ^{T} as specified by SIDE and TRANS.

10: LDC – INTEGER *Input*

On entry: the first dimension of the array C as declared in the (sub)program from which F08CGF (DORMQL) is called.

Constraint: $\mathrm{LDC} \geq \max(1, \mathrm{M})$.

11: WORK(∗) – ***double precision*** array *Workspace*

Note: the dimension of the array WORK must be at least max(1, LWORK).

On exit: if INFO = 0, WORK(1) contains the minimum value of LWORK required for optimal performance.

12: LWORK – INTEGER *Input*

On entry: the dimension of the array WORK as declared in the (sub)program from which F08CGF (DORMQL) is called.

If LWORK = −1, a workspace query is assumed; the routine only calculates the optimal size of the WORK array, returns this value as the first entry of the WORK array, and no error message related to LWORK is issued.

Suggested value: for optimal performance, LWORK ≥ N × nb if SIDE = 'L' and at least M × nb if SIDE = 'R', where nb is the optimal ***block size***.

Constraints:

if SIDE = 'L', LWORK ≥ max(1, N) or LWORK = −1;
if SIDE = 'R', LWORK ≥ max(1, M) or LWORK = −1.

13: INFO – INTEGER *Output*

On exit: INFO = 0 unless the routine detects an error (see Section 6).

6 Error Indicators and Warnings

Errors or warnings detected by the routine:

INFO < 0

If INFO = −i, the ith parameter had an illegal value. An explanatory message is output, and execution of the program is terminated.

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 routine is F08CUF (ZUNMQL).

9 Example

See Section 9 of the document for F08CEF (DGEQLF).