NAG Library Routine Document

F08ACF (DGEMQRT)

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

F08ACF (DGEMQRT) multiplies an arbitrary real matrix C by the real orthogonal matrix Q from a QR factorization computed by F08ABF (DGEQRT).

2 Specification

```
SUBROUTINE F08ACF (SIDE, TRANS, M, N, K, NB, V, LDV, T, LDT, C, LDC,       &
                   WORK, INFO)

INTEGER             M, N, K, NB, LDV, LDT, LDC, INFO
REAL (KIND=nag_wp) V(LDV,*), T(LDT,*), C(LDC,*), WORK(*)
CHARACTER(1)        SIDE, TRANS
```

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

3 Description

F08ACF (DGEMQRT) is intended to be used after a call to F08ABF (DGEQRT) which performs a QR factorization of a real matrix A. The orthogonal matrix Q is represented as a product of elementary reflectors.

This routine 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 routine is in solving linear least squares problems, as described in the F08 Chapter Introduction and illustrated in Section 10 in F08ABF (DGEQRT).

4 References

Golub G H and Van Loan C F (2012) *Matrix Computations* (4th Edition) Johns Hopkins University Press, Baltimore

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. Usually $\mathrm{K} = \min(m_A, n_A)$ where m_A, n_A are the dimensions of the matrix A supplied in a previous call to F08ABF (DGEQRT).

Constraints:

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

6: NB – INTEGER *Input*

On entry: the block size used in the QR factorization performed in a previous call to F08ABF (DGEQRT); this value must remain unchanged from that call.

Constraints:

$\mathrm{NB} \geq 1$;
if $\mathrm{K} > 0$, $\mathrm{NB} \leq \mathrm{K}$.

7: V(LDV, $*$) – REAL (KIND=nag_wp) array *Input*

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

On entry: details of the vectors which define the elementary reflectors, as returned by F08ABF (DGEQRT) in the first k columns of its array parameter A.

8: LDV – INTEGER *Input*

On entry: the first dimension of the array V as declared in the (sub)program from which F08ACF (DGEMQRT) is called.

Constraints:

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

9: T(LDT, $*$) – REAL (KIND=nag_wp) array *Input*

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

On entry: further details of the orthogonal matrix Q as returned by F08ABF (DGEQRT). The number of blocks is $b = \left\lceil \frac{k}{\mathrm{NB}} \right\rceil$, where $k = \min(m, n)$ and each block is of order NB except for the

last block, which is of order $k-(b-1)\times$ NB. For the b blocks the upper triangular block reflector factors $\boldsymbol{T}_1, \boldsymbol{T}_2, \ldots, \boldsymbol{T}_b$ are stored in the NB by n matrix T as $\boldsymbol{T} = [\boldsymbol{T}_1|\boldsymbol{T}_2|\ldots|\boldsymbol{T}_b]$.

10: LDT – INTEGER *Input*

On entry: the first dimension of the array T as declared in the (sub)program from which F08ACF (DGEMQRT) is called.

Constraint: LDT $\geq$ NB.

11: C(LDC, $*$) – REAL (KIND=nag_wp) array *Input/Output*

Note: the second dimension of the array C must be at least max(1, N).

On entry: the m by n matrix C.

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

12: LDC – INTEGER *Input*

On entry: the first dimension of the array C as declared in the (sub)program from which F08ACF (DGEMQRT) is called.

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

13: WORK($*$) – REAL (KIND=nag_wp) array *Workspace*

Note: the dimension of the array WORK must be at least N $\times$ NB if SIDE = 'L' and at least M $\times$ NB if SIDE = 'R'.

14: INFO – INTEGER *Output*

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

6 Error Indicators and Warnings

INFO < 0

If INFO $= -i$, argument i 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 Parallelism and Performance

F08ACF (DGEMQRT) is not threaded by NAG in any implementation.

F08ACF (DGEMQRT) makes calls to BLAS and/or LAPACK routines, which may be threaded within the vendor library used by this implementation. Consult the documentation for the vendor library for further information.

Please consult the X06 Chapter Introduction for information on how to control and interrogate the OpenMP environment used within this routine. Please also consult the Users' Note for your implementation for any additional implementation-specific information.

9 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 F08AQF (ZGEMQRT).

10 Example

See Section 10 in F08ABF (DGEQRT).