NAG Library Routine Document

F08AQF (ZGEMQRT)

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

F08AQF (ZGEMQRT) multiplies an arbitrary complex matrix C by the complex unitary matrix Q from a QR factorization computed by F08APF (ZGEQRT).

2 Specification

```
SUBROUTINE F08AQF (SIDE, TRANS, M, N, K, NB, V, LDV, T, LDT, C, LDC,       &
                   WORK, INFO)

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

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

3 Description

F08AQF (ZGEMQRT) is intended to be used after a call to F08APF (ZGEQRT), which performs a QR factorization of a complex matrix A. The unitary matrix Q is represented as a product of elementary reflectors.

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

$$QC, Q^{\mathrm{H}}C, CQ \text{ or } CQ^{\mathrm{H}},$$

overwriting the result on C (which may be any complex 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 F08APF (ZGEQRT).

4 References

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

5 Arguments

1: SIDE – CHARACTER(1) *Input*

On entry: indicates how Q or Q^{H} is to be applied to C.

SIDE = 'L'
Q or Q^{H} is applied to C from the left.

SIDE = 'R'
Q or Q^{H} is applied to C from the right.

Constraint: SIDE = 'L' or 'R'.

2: TRANS – CHARACTER(1) *Input*

On entry: indicates whether Q or Q^{H} is to be applied to C.

TRANS = 'N'
Q is applied to C.

TRANS = 'C'
Q^{H} is applied to C.

Constraint: TRANS = 'N' or 'C'.

3: M – INTEGER *Input*

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

Constraint: M ≥ 0.

4: N – INTEGER *Input*

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

Constraint: N ≥ 0.

5: K – INTEGER *Input*

On entry: k, the number of elementary reflectors whose product defines the matrix Q. Usually K $= \min(m_A, n_A)$ where m_A, n_A are the dimensions of the matrix A supplied in a previous call to F08APF (ZGEQRT).

Constraints:

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

6: NB – INTEGER *Input*

On entry: the block size used in the QR factorization performed in a previous call to F08APF (ZGEQRT); this value must remain unchanged from that call.

Constraints:

NB ≥ 1;
if K > 0, NB $\leq$ K.

7: V(LDV, $*$) – COMPLEX (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 F08APF (ZGEQRT) in the first k columns of its array argument A.

8: LDV – INTEGER *Input*

On entry: the first dimension of the array V as declared in the (sub)program from which F08AQF (ZGEMQRT) is called.

Constraints:

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

9: T(LDT, $*$) – COMPLEX (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 unitary matrix Q as returned by F08APF (ZGEQRT). 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 F08AQF (ZGEMQRT) is called.

Constraint: LDT $\geq$ NB.

11: C(LDC, $*$) – COMPLEX (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 $Q^{\mathrm{H}}C$ or CQ or CQ^{H} 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 F08AQF (ZGEMQRT) is called.

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

13: WORK($*$) – COMPLEX (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

F08AQF (ZGEMQRT) 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 real floating-point operations is approximately $8nk(2m-k)$ if SIDE = 'L' and $8mk(2n-k)$ if SIDE = 'R'.

The real analogue of this routine is F08ACF (DGEMQRT).

10 Example

See Section 10 in F08APF (ZGEQRT).