NAG Library Routine Document

F08BQF (ZTPMQRT)

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

F08BQF (ZTPMQRT) multiplies an arbitrary complex matrix C by the complex unitary matrix Q from a QR factorization computed by F08BPF (ZTPQRT).

2 Specification

```
SUBROUTINE F08BQF (SIDE, TRANS, M, N, K, L, NB, V, LDV, T, LDT, C1,        &
                   LDC1, C2, LDC2, WORK, INFO)
INTEGER               M, N, K, L, NB, LDV, LDT, LDC1, LDC2, INFO
COMPLEX (KIND=nag_wp) V(LDV,*), T(LDT,*), C1(LDC1,*), C2(LDC2,*),          &
                      WORK(*)
CHARACTER(1)          SIDE, TRANS
```

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

3 Description

F08BQF (ZTPMQRT) is intended to be used after a call to F08BPF (ZTPQRT) which performs a QR factorization of a triangular-pentagonal matrix containing an upper triangular matrix A over a pentagonal matrix B. The unitary matrix Q is represented as a product of elementary reflectors.

This routine may be used to form the matrix products

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

where the complex rectangular m_c by n_c matrix C is split into component matrices C_1 and C_2.

If Q is being applied from the left (QC or $Q^{\mathrm{H}}C$) then

$$C = \begin{pmatrix} C_1 \\ C_2 \end{pmatrix}$$

where C_1 is k by n_c, C_2 is m_v by n_c, $m_c = k + m_v$ is fixed and m_v is the number of rows of the matrix V containing the elementary reflectors (i.e., M as passed to F08BPF (ZTPQRT)); the number of columns of V is n_v (i.e., N as passed to F08BPF (ZTPQRT)).

If Q is being applied from the right (CQ or CQ^{H}) then

$$C = \begin{pmatrix} C_1 & C_2 \end{pmatrix}$$

where C_1 is m_c by k, and C_2 is m_c by m_v and $n_c = k + m_v$ is fixed.

The matrices C_1 and C_2 are overwriten by the result of the matrix product.

A common application of this routine is in updating the solution of a linear least squares problem as illustrated in Section 10 in F08BPF (ZTPQRT).

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^{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: the number of rows of the matrix C_2, that is,

if SIDE = 'L'
then m_v, the number of rows of the matrix V;

if SIDE = 'R'
then m_c, the number of rows of the matrix C.

Constraint: M ≥ 0.

4: N – INTEGER *Input*

On entry: the number of columns of the matrix C_2, that is,

if SIDE = 'L'
then n_c, the number of columns of the matrix C;

if SIDE = 'R'
then n_v, the number of columns of the matrix V.

Constraint: N ≥ 0.

5: K – INTEGER *Input*

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

Constraint: K ≥ 0.

6: L – INTEGER *Input*

On entry: l, the number of rows of the upper trapezoidal part of the pentagonal composite matrix V, passed (as B) in a previous call to F08BPF (ZTPQRT). This must be the same value used in the previous call to F08BPF (ZTPQRT) (see L in F08BPF (ZTPQRT)).

Constraint: $0 \leq$ L $\leq$ K.

7: NB – INTEGER *Input*

On entry: nb, the blocking factor used in a previous call to F08BPF (ZTPQRT) to compute the QR factorization of a triangular-pentagonal matrix containing composite matrices A and B.

Constraints:

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

8: V(LDV, $*$) – COMPLEX (KIND=nag_wp) array *Input*

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

On entry: the m_v by n_v matrix V; this should remain unchanged from the array B returned by a previous call to F08BPF (ZTPQRT).

9: LDV – INTEGER *Input*

On entry: the first dimension of the array V as declared in the (sub)program from which F08BQF (ZTPMQRT) is called.

Constraints:

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

10: 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: this must remain unchanged from a previous call to F08BPF (ZTPQRT) (see T in F08BPF (ZTPQRT)).

11: LDT – INTEGER *Input*

On entry: the first dimension of the array T as declared in the (sub)program from which F08BQF (ZTPMQRT) is called.

Constraint: LDT $\geq$ NB.

12: C1(LDC1, $*$) – COMPLEX (KIND=nag_wp) array *Input/Output*

Note: the second dimension of the array C1 must be at least $\max(1, \mathrm{N})$ if SIDE = 'L' and at least $\max(1, \mathrm{K})$ if SIDE = 'R'.

On entry: C_1, the first part of the composite matrix C.

if SIDE = 'L'
then C1 contains the first k rows of C;

if SIDE = 'R'
then C1 contains the first k columns of C.

On exit: C1 is overwritten by the corresponding block of QC or $Q^{\mathrm{H}}C$ or CQ or CQ^{H}.

13: LDC1 – INTEGER *Input*

On entry: the first dimension of the array C1 as declared in the (sub)program from which F08BQF (ZTPMQRT) is called.

Constraints:

if SIDE = 'L', LDC1 $\geq \max(1, \mathrm{K})$;
if SIDE = 'R', LDC1 $\geq \max(1, \mathrm{M})$.

14: C2(LDC2, $*$) – COMPLEX (KIND=nag_wp) array *Input/Output*

Note: the second dimension of the array C2 must be at least $\max(1, \mathrm{N})$.

On entry: C_2, the second part of the composite matrix C.

if SIDE = 'L'
then C2 contains the remaining m_v rows of C;

if SIDE = 'R'
then C2 contains the remaining m_v columns of C;

On exit: C2 is overwritten by the corresponding block of QC or $Q^{\mathrm{H}}C$ or CQ or CQ^{H}.

15: LDC2 – INTEGER *Input*

On entry: the first dimension of the array C2 as declared in the (sub)program from which F08BQF (ZTPMQRT) is called.

Constraint: $\mathrm{LDC2} \geq \max(1, \mathrm{M})$.

16: WORK($*$) – COMPLEX (KIND=nag_wp) array *Workspace*

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

17: 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

F08BQF (ZTPMQRT) is not threaded by NAG in any implementation.

F08BQF (ZTPMQRT) 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 real analogue of this routine is F08BCF (DTPMQRT).

10 Example

See Section 10 in F08BPF (ZTPQRT).