NAG Library Routine Document

F08BXF (ZUNMRZ)

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

F08BXF (ZUNMRZ) multiplies a general complex m by n matrix C by the complex unitary matrix Z from an RZ factorization computed by F08BVF (ZTZRZF).

2 Specification

```
SUBROUTINE F08BXF (SIDE, TRANS, M, N, K, L, A, LDA, TAU, C, LDC, WORK,     &
                   LWORK, INFO)

INTEGER                 M, N, K, L, LDA, LDC, LWORK, INFO
COMPLEX (KIND=nag_wp) A(LDA,*), TAU(*), C(LDC,*), WORK(max(1,LWORK))
CHARACTER(1)            SIDE, TRANS
```

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

3 Description

F08BXF (ZUNMRZ) is intended to be used following a call to F08BVF (ZTZRZF), which performs an RZ factorization of a real upper trapezoidal matrix A and represents the unitary matrix Z as a product of elementary reflectors.

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

$$ZC, \quad Z^{\mathrm{H}}C, \quad CZ, \quad CZ^{\mathrm{H}},$$

overwriting the result on C, which may be any complex rectangular m by n matrix.

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 Arguments

1: SIDE – CHARACTER(1) *Input*

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

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

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

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

2: TRANS – CHARACTER(1) *Input*

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

TRANS = 'N'
Z is applied to C.

TRANS = 'C'
Z^{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 Z.

Constraints:

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

6: L – INTEGER *Input*

On entry: l, the number of columns of the matrix A containing the meaningful part of the Householder reflectors.

Constraints:

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

7: A(LDA, $*$) – COMPLEX (KIND=nag_wp) array *Input*

Note: the second dimension of the array A must be at least max(1, M) if SIDE = 'L' and at least max(1, N) if SIDE = 'R'.

On entry: the ith row of A must contain the vector which defines the elementary reflector H_i, for $i = 1, 2, \ldots, k$, as returned by F08BVF (ZTZRZF).

8: LDA – INTEGER *Input*

On entry: the first dimension of the array A as declared in the (sub)program from which F08BXF (ZUNMRZ) is called.

Constraint: LDA $\geq$ max(1, K).

9: TAU($*$) – COMPLEX (KIND=nag_wp) array *Input*

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

On entry: TAU(i) must contain the scalar factor of the elementary reflector H_i, as returned by F08BVF (ZTZRZF).

10: 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 ZC or $Z^{\mathrm{H}}C$ or CZ or $Z^{\mathrm{H}}C$ as specified by SIDE and TRANS.

11: LDC – INTEGER *Input*

On entry: the first dimension of the array C as declared in the (sub)program from which F08BXF (ZUNMRZ) is called.

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

12: WORK(max(1, LWORK)) – COMPLEX (KIND=nag_wp) array *Workspace*

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

13: LWORK – INTEGER *Input*

On entry: the dimension of the array WORK as declared in the (sub)program from which F08BXF (ZUNMRZ) 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 $\geq \mathrm{N} \times nb$ if SIDE $=$ 'L' and at least $\mathrm{M} \times nb$ if SIDE $=$ 'R', where nb is the optimal ***block size***.

Constraints:

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

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.

If INFO $= -999$, dynamic memory allocation failed.

See Section 3.7 in How to Use the NAG Library and its Documentation for further information.

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

F08BXF (ZUNMRZ) 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 $16nlk$ if SIDE = 'L' and $16mlk$ if SIDE = 'R'.

The real analogue of this routine is F08BKF (DORMRZ).

10 Example

See Section 10 in F08BVF (ZTZRZF).