# NAG Fortran 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,
     1                   LWORK, INFO)
      INTEGER            M, N, K, L, LDA, LDC, LWORK, INFO
      complex*16         A(LDA,*), TAU(*), C(LDC,*), WORK(*)
      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 URL: http://www.netlib.org/lapack/lug

## 5 Parameters

1: SIDE – CHARACTER*1 *Input*

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

SIDE = 'L'

$Z$ or $Z^{\mathrm{H}}$ is applied to $C$ from the left.

SIDE = 'R'

$Z$ or $Z^{\mathrm{H}}$ is applied to $C$ from the right.

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

2: TRANS – CHARACTER*1 *Input*

*On entry*: if TRANS = 'N', no transpose, apply $Z$.

If TRANS = 'C', transpose, apply $Z^{H}$.

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

3: M – INTEGER *Input*

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

*Constraint*: $M \geq 0$.

4: N – INTEGER *Input*

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

*Constraint*: $N \geq 0$.

5: K – INTEGER *Input*

*On entry*: $k$, the number of elementary reflectors whose product defines the matrix $Z$.

*Constraints*:

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

6: L – INTEGER *Input*

*On entry*: the number of columns of the matrix $A$ containing the meaningful part of the Householder reflectors.

*Constraints*:

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

7: A(LDA,*) – ***complex*16*** array *Input/Output*

**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 $i$th 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).

*On exit*: is modified by F08BXF (ZUNMRZ) but restored on exit.

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*16*** 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 F08BHF (DTZRZF).

10: C(LDC,*) – ***complex*16*** 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*: is overwritten by $ZC$ or $Z^{H}C$ or $CZ$ or $Z^{H}C$ 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($*$) – ***complex*16*** array *Workspace*

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

*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

Errors or warnings detected by the routine:

INFO $< 0$

If INFO $= -i$, the $i$th 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 $\epsilon$ is the ***machine precision***.

# 8 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).

# 9 Example

See Section 9 of the document for F08BVF (ZTZRZF).