NAG Library Function Document

nag_zunmrz (f08bxc)

1 Purpose

nag_zunmrz (f08bxc) multiplies a general complex \( m \) by \( n \) matrix \( C \) by the complex unitary matrix \( Z \) from an \( RZ \) factorization computed by nag_ztzrzf (f08bvc).

2 Specification

```c
#include <nag.h>
#include <nagf08.h>
void nag_zunmrz (Nag_OrderType order, Nag_SideType side,
    Nag_TransType trans, Integer m, Integer n, Integer k, Integer l,
    const Complex a[], Integer pda, const Complex tau[], Complex c[],
    Integer pdc, NagError *fail)
```

3 Description

nag_zunmrz (f08bxc) is intended to be used following a call to nag_ztzrzf (f08bvc), 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 function may be used to form one of the matrix products

\[
ZC, \quad Z^H C, \quad CZ, \quad CZ^H,
\]

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

4 References


5 Arguments

1: \textbf{order} -- Nag_OrderType \hspace{1cm} \textit{Input}

\textit{On entry}: the \textbf{order} argument specifies the two-dimensional storage scheme being used, i.e., row-major ordering or column-major ordering. C language defined storage is specified by \textbf{order} = Nag_RowMajor. See Section 3.2.1.3 in the Essential Introduction for a more detailed explanation of the use of this argument.

\textit{Constraint}: \textbf{order} = Nag_RowMajor or Nag_ColMajor.

2: \textbf{side} -- Nag_SideType \hspace{1cm} \textit{Input}

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

\textbf{side} = Nag_LeftSide
\hspace{1cm} \( Z \) or \( Z^H \) is applied to \( C \) from the left.

\textbf{side} = Nag_RightSide
\hspace{1cm} \( Z \) or \( Z^H \) is applied to \( C \) from the right.

\textit{Constraint}: \textbf{side} = Nag_LeftSide or Nag_RightSide.
3:  trans – Nag_TransType  
   On entry: indicates whether $Z$ or $Z^H$ is to be applied to $C$.
   trans = Nag_NoTrans  
       $Z$ is applied to $C$.
   trans = Nag_ConjTrans  
       $Z^H$ is applied to $C$.
   Constraint: trans = Nag_NoTrans or Nag_ConjTrans.

4:  m – Integer  
   On entry: $m$, the number of rows of the matrix $C$.
   Constraint: $m \geq 0$.

5:  n – Integer  
   On entry: $n$, the number of columns of the matrix $C$.
   Constraint: $n \geq 0$.

6:  k – Integer  
   On entry: $k$, the number of elementary reflectors whose product defines the matrix $Z$.
   Constraints:
   if side = Nag_LeftSide, $m \geq k \geq 0$;
   if side = Nag_RightSide, $n \geq k \geq 0$.

7:  l – Integer  
   On entry: $l$, the number of columns of the matrix $A$ containing the meaningful part of the Householder reflectors.
   Constraints:
   if side = Nag_LeftSide, $m \geq l \geq 0$;
   if side = Nag_RightSide, $n \geq l \geq 0$.

8:  a[dim] – const Complex  
   Input
   Note: the dimension, dim, of the array a must be at least
   $\max(1, pda \times m)$ when side = Nag_LeftSide and order = Nag_ColMajor;
   $\max(1, k \times pda)$ when side = Nag_LeftSide and order = Nag_RowMajor;
   $\max(1, pda \times n)$ when side = Nag_RightSide and order = Nag_ColMajor;
   $\max(1, k \times pda)$ when side = Nag_RightSide and order = Nag_RowMajor.
   The $(i,j)$th element of the matrix $A$ is stored in
   $\begin{cases} a[(j - 1) \times pda + i - 1] & \text{when order = Nag_ColMajor;} \\ a[(i - 1) \times pda + j - 1] & \text{when order = Nag_RowMajor.} \end{cases}$
   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 nag_ztzrzf (f08bvc).

9:  pda – Integer  
   Input
   On entry: the stride separating row or column elements (depending on the value of order) in the array a.
Constraints:

if order = Nag_ColMajor, pda ≥ max(1, k);
if order = Nag_RowMajor,
  if side = Nag_LeftSide, pda ≥ max(1, m);
  if side = Nag_RightSide, pda ≥ max(1, n).

10: tau[dim] – const Complex

*Input*

Note: the dimension, dim, of the array tau must be at least max(1, k).

On entry: tau[i – 1] must contain the scalar factor of the elementary reflector Hi, as returned by nag_ztzrzf (f08bvc).

11: c[dim] – Complex

*Input/Output*

Note: the dimension, dim, of the array c must be at least

max(1, pdc × n) when order = Nag_ColMajor;
max(1, m × pdc) when order = Nag_RowMajor.

The (i, j)th element of the matrix C is stored in

\[c[(j - 1) \times pdc + i - 1] \text{ when order = Nag_ColMajor};\]
\[c[(i - 1) \times pdc + j - 1] \text{ when order = Nag_RowMajor}.\]

On entry: the m by n matrix C.

On exit: c is overwritten by ZC or ZH C or CZ or ZH C as specified by side and trans.

12: pdc – Integer

*Input*

On entry: the stride separating row or column elements (depending on the value of order) in the array c.

Constraints:

if order = Nag_ColMajor, pdc ≥ max(1, m);
if order = Nag_RowMajor, pdc ≥ max(1, n).

13: fail – NagError *

*Input/Output*

The NAG error argument (see Section 3.6 in the Essential Introduction).

6 Error Indicators and Warnings

NE_ALLOC_FAIL

Dynamic memory allocation failed.
See Section 3.2.1.2 in the Essential Introduction for further information.

NE_BAD_PARAM

On entry, argument \langle value\rangle had an illegal value.

NE_ENUM_INT_3

On entry, side = \langle value\rangle, m = \langle value\rangle, n = \langle value\rangle and k = \langle value\rangle.
Constraint: if side = Nag_LeftSide, m ≥ k ≥ 0;
if side = Nag_RightSide, n ≥ k ≥ 0.

On entry, side = \langle value\rangle, m = \langle value\rangle, n = \langle value\rangle and l = \langle value\rangle.
Constraint: if side = Nag_LeftSide, m ≥ l ≥ 0;
if side = Nag_RightSide, n ≥ l ≥ 0.
On entry, $\text{side} = \langle value \rangle$, $\text{pda} = \langle value \rangle$, $\text{m} = \langle value \rangle$ and $\text{n} = \langle value \rangle$.
Constraint: if $\text{side} = \text{Nag\_LeftSide}$, $\text{pda} \geq \max(1, m)$;
if $\text{side} = \text{Nag\_RightSide}$, $\text{pda} \geq \max(1, n)$. 

**NE\_INT** 
On entry, $\text{m} = \langle value \rangle$.
Constraint: $\text{m} \geq 0$.
On entry, $\text{n} = \langle value \rangle$.
Constraint: $\text{n} \geq 0$.
On entry, $\text{pda} = \langle value \rangle$.
Constraint: $\text{pda} > 0$.
On entry, $\text{pdc} = \langle value \rangle$.
Constraint: $\text{pdc} > 0$.

**NE\_INT\_2** 
On entry, $\text{pda} = \langle value \rangle$ and $\text{k} = \langle value \rangle$.
Constraint: $\text{pda} \geq \max(1, k)$.
On entry, $\text{pdc} = \langle value \rangle$ and $\text{m} = \langle value \rangle$.
Constraint: $\text{pdc} \geq \max(1, m)$.
On entry, $\text{pdc} = \langle value \rangle$ and $\text{n} = \langle value \rangle$.
Constraint: $\text{pdc} \geq \max(1, n)$.

**NE\_INTERNAL\_ERROR**
An internal error has occurred in this function. Check the function call and any array sizes. If the call is correct then please contact NAG for assistance.
An unexpected error has been triggered by this function. Please contact NAG.
See Section 3.6.6 in the Essential Introduction for further information.

**NE\_NO\_LICENCE**
Your licence key may have expired or may not have been installed correctly.
See Section 3.6.5 in the Essential Introduction for further information.

### 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 Parallelism and Performance 

*nag\_zunmrz* (f08bxc) is not threaded by NAG in any implementation.

*nag\_zunmrz* (f08bxc) 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 function. 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 = Nag_LeftSide and $16mlk$ if side = Nag_RightSide.

The real analogue of this function is nag_dormrz (f08bkc).

10 Example

See Section 10 in nag_ztzrzf (f08bvc).