NAG C Library Function Document

nag_ztr_copy (f16tec)

1 Purpose

nag_ztr_copy (f16tec) copies a complex triangular matrix.

2 Specification

```c
void nag_ztr_copy (Nag_OrderType order, Nag_UploType uplo, Nag_TransType trans,
Nag_DiagType diag, Integer n, const Complex a[], Integer pda, Complex b[],
Integer pdb, NagError *fail)
```

3 Description

nag_ztr_copy (f16tec) performs the triangular matrix copy operations

\[ B \leftarrow A, \quad B \leftarrow A^T \text{ or } B \leftarrow A^H. \]

where \( A \) and \( B \) are \( n \) by \( n \) complex triangular matrices.

4 References


5 Parameters

1: \texttt{order} – Nag_OrderType

\textit{Input}

\textit{On entry:} the \texttt{order} parameter specifies the two-dimensional storage scheme being used, i.e., row-major ordering or column-major ordering. C language defined storage is specified by \texttt{order = Nag_RowMajor}. See Section 2.2.1.4 of the Essential Introduction for a more detailed explanation of the use of this parameter.

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

2: \texttt{uplo} – Nag_UploType

\textit{Input}

\textit{On entry:} specifies whether the upper or lower triangular part of \( A \) is stored as follows:

- if \texttt{uplo = Nag_Upper}, the upper triangular part of \( A \) is stored;
- if \texttt{uplo = Nag_Lower}, the lower triangular part of \( A \) is stored.

\textit{Constraint:} \texttt{uplo = Nag_Upper} or \texttt{Nag_Lower}.

3: \texttt{trans} – Nag_TransType

\textit{Input}

\textit{On entry:} specifies the operation to be performed as follows:

- if \texttt{trans = Nag_NoTrans}, \( B \leftarrow A \);
- if \texttt{trans = Nag_Trans}, \( B \leftarrow A^T \);
- if \texttt{trans = Nag_ConjTrans}, \( B \leftarrow A^H \).

\textit{Constraint:} \texttt{trans = Nag_NoTrans, Nag_Trans or Nag_ConjTrans}.

4: \texttt{diag} – Nag_DiagType

\textit{Input}

\textit{On entry:} specifies whether \( A \) has non-unit or unit diagonal elements, as follows:
if \( \text{diag} = \text{Nag\_NonUnitDiag} \), the diagonal elements are stored explicitly;

if \( \text{diag} = \text{Nag\_UnitDiag} \), the diagonal elements are assumed to be 1, and are not referenced.

\[ \text{Constraint: } \text{diag} = \text{Nag\_NonUnitDiag} \text{ or } \text{Nag\_UnitDiag}. \]

5: \( n \) – Integer

On entry: \( n \), the order of the matrices \( A \) and \( B \).

\[ \text{Constraint: } n \geq 0. \]

6: \( a[\text{dim}] \) – const Complex

\[ \text{Input} \]

Note: the dimension, \( \text{dim} \), of the array \( a \) must be at least \( \max(1, pda \times n) \).

If \( \text{order} = \text{Nag\_ColMajor} \), the \((i, j)\)th element of the matrix \( A \) is stored in \( a[(j - 1) \times pda + i - 1] \) and if \( \text{order} = \text{Nag\_RowMajor} \), the \((i, j)\)th element of the matrix \( A \) is stored in \( a[(i - 1) \times pda + j - 1] \).

On entry: the \( n \) by \( n \) triangular matrix \( A \). If \( \text{uplo} = \text{Nag\_Upper} \) \( A \) is upper triangular and the elements of the array below the diagonal are not referenced; if \( \text{uplo} = \text{Nag\_Lower} \) \( A \) is lower triangular and the elements of the array above the diagonal are not referenced.

7: \( pda \) – Integer

\[ \text{Input} \]

On entry: the stride separating matrix row or column elements (depending on the value of \( \text{order} \)) in the array \( a \).

\[ \text{Constraint: } pda \geq \max(1, n). \]

8: \( b[\text{dim}] \) – Complex

\[ \text{Output} \]

Note: the dimension, \( \text{dim} \), of the array \( b \) must be at least \( \max(1, pdb \times n) \).

If \( \text{order} = \text{Nag\_ColMajor} \), the \((i, j)\)th element of the matrix \( B \) is stored in \( b[(j - 1) \times pdb + i - 1] \) and if \( \text{order} = \text{Nag\_RowMajor} \), the \((i, j)\)th element of the matrix \( B \) is stored in \( b[(i - 1) \times pdb + j - 1] \).

On exit: the \( n \) by \( n \) triangular matrix \( B \). If \( \text{uplo} = \text{Nag\_Upper} \) \( B \) is upper triangular and the elements of the array below the diagonal are not set; if \( \text{uplo} = \text{Nag\_Lower} \) \( B \) is lower triangular and the elements of the array above the diagonal are not set.

9: \( pdb \) – Integer

\[ \text{Input} \]

On entry: the stride separating matrix row or column elements (depending on the value of \( \text{order} \)) in the array \( b \).

\[ \text{Constraint: } pdb \geq \max(1, n). \]

10: \( \text{fail} \) – NagError *

\[ \text{Input/Output} \]

The NAG error parameter (see the Essential Introduction).

6 Error Indicators and Warnings

\( \text{NE\_INT} \)

On entry, \( n = \langle \text{value} \rangle \).

Constraint: \( n \geq 0 \).

On entry, \( pda = \langle \text{value} \rangle \).

Constraint: \( pda \geq \max(1, n) \).

On entry, \( pdb = \langle \text{value} \rangle \).

Constraint: \( pdb \geq \max(1, n) \).

\( \text{NE\_BAD\_PARAM} \)

On entry, parameter \( \langle \text{value} \rangle \) had an illegal value.
7 Accuracy

The BLAS standard requires accurate implementations which avoid unnecessary over/underflow (see section 2.7 of The BLAS Technical Forum Standard (2001)).

8 Further Comments

None.

9 Example

None.