

NAG Toolbox

nag_matop_complex_tri_pack (f01zb)

1 Purpose

nag_matop_complex_tri_pack (f01zb) copies a complex triangular matrix stored in a packed one-dimensional array into an unpacked two-dimensional array, or vice versa.

2 Syntax

```
[a, b, ifail] = nag_matop_complex_tri_pack(job, uplo, diag, a, b, 'n', n)
[a, b, ifail] = f01zb(job, uplo, diag, a, b, 'n', n)
```

3 Description

nag_matop_complex_tri_pack (f01zb) unpacks a triangular matrix stored in a vector into a two-dimensional array, or packs a triangular matrix stored in a two-dimensional array into a vector. The matrix is packed by column. This function is intended for possible use in conjunction with functions from Chapters F07 and F08, where some functions that use triangular matrices store them in the packed form described below.

4 References

None.

5 Parameters

5.1 Compulsory Input Parameters

1: **job** – CHARACTER(1)

Specifies whether the triangular matrix is to be packed or unpacked.

job = 'P' (Pack)

The matrix is to be packed into array **b**.

job = 'U' (Unpack)

The matrix is to be unpacked into array **a**.

Constraint: **job** = 'P' or 'U'.

2: **uplo** – CHARACTER(1)

Specifies the type of the matrix to be copied

uplo = 'L' (Lower)

The matrix is lower triangular. In this case the packed vector holds, or will hold on exit, the matrix elements in the following order: (1,1), (2,1), ..., (n,1), (2,2), (3,2), ..., (n,2), etc..

uplo = 'U' (Upper)

The matrix is upper triangular. In this case the packed vector holds, or will hold on exit, the matrix elements in the following order: (1,1), (1,2), (2,2), (1,3), (2,3), (3,3), (1,4), etc..

Constraint: **uplo** = 'L' or 'U'.

3: **diag** – CHARACTER(1)

Must specify whether the diagonal elements of the matrix are to be copied.

diag = 'B' (Blank)

The diagonal elements of the matrix are not referenced and not copied.

diag = 'U' (Unit diagonal)

The diagonal elements of the matrix are not referenced, but are assumed all to be unity, and are copied as such.

diag = 'N' (Non-unit diagonal)

The diagonal elements of the matrix are referenced and copied.

Constraint: **diag** = 'B', 'U' or 'N'.

4: **a**(*lda*, **n**) – COMPLEX (KIND=nag_wp) array

lda, the first dimension of the array, must satisfy the constraint $lda \geq n$.

If **job** = 'P', then the leading **n** by **n** part of **a** must contain the matrix to be copied, stored in unpacked form, in the upper or lower triangle depending on argument **uplo**. The opposite triangle of **a** is not referenced and need not be assigned.

5: **b**((**n** × (**n** + 1))/2) – COMPLEX (KIND=nag_wp) array

If **job** = 'U', then **b** must contain the triangular matrix packed by column.

5.2 Optional Input Parameters

1: **n** – INTEGER

Default: the first dimension of the array **a** and the second dimension of the array **a**. (An error is raised if these dimensions are not equal.)

n must specify the number of rows and columns of the triangular matrix.

Constraint: **n** > 0.

5.3 Output Parameters

1: **a**(*lda*, **n**) – COMPLEX (KIND=nag_wp) array

If **job** = 'U', then the leading **n** by **n** part of array **a** contains the copied matrix, stored in unpacked form, in the upper or lower triangle depending on argument **uplo**. The opposite triangle of **a** is not referenced.

2: **b**((**n** × (**n** + 1))/2) – COMPLEX (KIND=nag_wp) array

If **job** = 'P', then **b** contains the triangular matrix packed by column.

Note that **b** must have space for the diagonal elements of the matrix, even if these are not stored.

3: **ifail** – INTEGER

ifail = 0 unless the function detects an error (see Section 5).

6 Error Indicators and Warnings

Errors or warnings detected by the function:

ifail = 1

On entry, **job** ≠ 'P' or 'U'.

ifail = 2On entry, **uplo** ≠ 'L' or 'U'.**ifail** = 3On entry, **diag** ≠ 'N', 'U' or 'B'.**ifail** = 4On entry, **n** < 1.**ifail** = 5On entry, **lda** < **n**.**ifail** = -99

An unexpected error has been triggered by this routine. Please contact NAG.

ifail = -399

Your licence key may have expired or may not have been installed correctly.

ifail = -999

Dynamic memory allocation failed.

7 Accuracy

Not applicable.

8 Further Comments

None.

9 Example

This example reads in a triangular matrix A , and copies it to the packed matrix B .

9.1 Program Text

```
function f01zb_example

fprintf('f01zb example results\n\n');

job = 'Pack';
uplo = 'U';
diag = 'N';
a = [ 1.1 + 1.1i,  1.2 + 1.2i,  1.3 + 1.3i,  1.4 + 1.4i;
      0    + 0i,     2.2 + 2.2i,  2.3 + 2.3i,  2.4 + 2.4i;
      0    + 0i,     0    + 0i,     3.3 + 3.3i,  3.4 + 3.4i;
      0    + 0i,     0    + 0i,     0    + 0i,     4.4 + 4.4i];
b = complex(zeros(10, 1));

[A, ap, ifail] = f01zb(job, uplo, diag, a, b);

disp('Unpacked Matrix A:');
disp(A);
fprintf('\n');
disp('Packed Vector ap:');
disp(ap);
```

9.2 Program Results

f01zb example results

Unpacked Matrix A:

1.1000 + 1.1000i	1.2000 + 1.2000i	1.3000 + 1.3000i	1.4000 + 1.4000i
0.0000 + 0.0000i	2.2000 + 2.2000i	2.3000 + 2.3000i	2.4000 + 2.4000i
0.0000 + 0.0000i	0.0000 + 0.0000i	3.3000 + 3.3000i	3.4000 + 3.4000i
0.0000 + 0.0000i	0.0000 + 0.0000i	0.0000 + 0.0000i	4.4000 + 4.4000i

Packed Vector ap:

1.1000 + 1.1000i
1.2000 + 1.2000i
2.2000 + 2.2000i
1.3000 + 1.3000i
2.3000 + 2.3000i
3.3000 + 3.3000i
1.4000 + 1.4000i
2.4000 + 2.4000i
3.4000 + 3.4000i
4.4000 + 4.4000i