

NAG Toolbox

nag_matop_dtftr (f01vg)

1 Purpose

nag_matop_dtftr (f01vg) unpacks a real triangular matrix, stored in a Rectangular Full Packed (RFP) format array, to a full format array.

2 Syntax

```
[a, info] = nag_matop_dtftr(transr, uplo, n, ar)
[a, info] = f01vg(transr, uplo, n, ar)
```

3 Description

nag_matop_dtftr (f01vg) unpacks a real n by n triangular matrix A , stored in RFP format to conventional storage in a full format array. This function is intended for possible use in conjunction with functions from Chapters F07 and F16 where some functions that use triangular matrices store them in RFP format. The RFP storage format is described in Section 3.2.3 in the F07 Chapter Introduction.

4 References

Gustavson F G, Waśniewski J, Dongarra J J and Langou J (2010) Rectangular full packed format for Cholesky's algorithm: factorization, solution, and inversion *ACM Trans. Math. Software* **37**, 2

5 Parameters

5.1 Compulsory Input Parameters

1: **transr** – CHARACTER(1)

Specifies whether the normal RFP representation of A or its transpose is stored.

transr = 'N'

The RFP representation of the matrix A is stored.

transr = 'T'

The transpose of the RFP representation of the matrix A is stored.

Constraint: **transr** = 'N' or 'T'.

2: **uplo** – CHARACTER(1)

Specifies whether A is upper or lower triangular.

uplo = 'U'

A is upper triangular.

uplo = 'L'

A is lower triangular.

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

3: **n** – INTEGER

n , the order of the matrix A .

Constraint: **n** \geq 0.

4: **ar**($n \times (n + 1)/2$) – REAL (KIND=nag_wp) array

The upper or lower n by n triangular matrix A (as specified by **uplo**) in either normal or transposed RFP format (as specified by **transr**). The storage format is described in Section 3.2.3 in the F07 Chapter Introduction.

5.2 Optional Input Parameters

None.

5.3 Output Parameters

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

The first dimension of the array **a** will be $\max(1, n)$.

The second dimension of the array **a** will be **n**.

The triangular matrix A .

If **uplo** = 'U', a is upper triangular and the elements of the array below the diagonal are not referenced.

If **uplo** = 'L', a is lower triangular and the elements of the array above the diagonal are not referenced.

2: **info** – INTEGER

info = 0 unless the function detects an error (see Section 6).

6 Error Indicators and Warnings

info < 0

If **info** = $-i$, argument i had an illegal value. An explanatory message is output, and execution of the program is terminated.

7 Accuracy

Not applicable.

8 Further Comments

None.

9 Example

This example reads in a triangular matrix in RFP format and unpacks it to full format.

9.1 Program Text

```
function f01vg_example

fprintf('f01vg example results\n\n');

transr = 'n';
uplo   = 'u';
n      = nag_int(4);
ar = [1.30; 2.30; 3.30; 1.10; 1.20; 1.40; 2.40; 3.40; 4.40; 2.20];
% Print the Rectangular Full Packed array
fprintf('\n');
[ifail] = x04cb('g', 'x', ar, 'f5.2', 'RFP Packed Array ar:', 'i', ...
               {''}, 'n', {''}, nag_int(80), nag_int(0));
```

```
% Convert to triangular form
[a, info] = f01vg(transr, uplo, n, ar);
% Print the unpacked matrix
fprintf('\n');
[ifail] = x04cb(uplo, 'n', a, 'f5.2', 'Unpacked matrix a:', 'i', ...
               {''}, 'i', {''}, nag_int(80), nag_int(0));
```

9.2 Program Results

f01vg example results

RFP Packed Array ar:

```
1  1.30
2  2.30
3  3.30
4  1.10
5  1.20
6  1.40
7  2.40
8  3.40
9  4.40
10 2.20
```

Unpacked matrix a:

```
      1      2      3      4
1  1.10  1.20  1.30  1.40
2      2.20  2.30  2.40
3      3.30  3.40
4      4.40
```
