

NAG Toolbox

nag_lapack_dtftri (f07wk)

1 Purpose

nag_lapack_dtftri (f07wk) computes the inverse of a real triangular matrix stored in Rectangular Full Packed (RFP) format.

2 Syntax

```
[ar, info] = nag_lapack_dtftri(transr, uplo, diag, n, ar)
[ar, info] = f07wk(transr, uplo, diag, n, ar)
```

3 Description

nag_lapack_dtftri (f07wk) forms the inverse of a real triangular matrix A , stored using RFP format. The RFP storage format is described in Section 3.2.3 in the F07 Chapter Introduction. Note that the inverse of an upper (lower) triangular matrix is also upper (lower) triangular.

4 References

Du Croz J J and Higham N J (1992) Stability of methods for matrix inversion *IMA J. Numer. Anal.* **12** 1–19

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 RFP representation of A is normal or transposed.

transr = 'N'

The matrix A is stored in normal RFP format.

transr = 'T'

The matrix A is stored in transposed RFP format.

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: **diag** – CHARACTER(1)

Indicates whether A is a nonunit or unit triangular matrix.

diag = 'N'

A is a nonunit triangular matrix.

diag = 'U'

A is a unit triangular matrix; the diagonal elements are not referenced and are assumed to be 1.

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

4: **n** – INTEGER

n , the order of the matrix A .

Constraint: **n** ≥ 0 .

5: **ar**($\mathbf{n} \times (\mathbf{n} + 1)/2$) – REAL (KIND=nag_wp) array

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

5.2 Optional Input Parameters

None.

5.3 Output Parameters

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

A stores A^{-1} , in the same storage format as A .

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.

info > 0 (*warning*)

Diagonal element $\langle value \rangle$ of A is exactly zero. A is singular its inverse cannot be computed.

7 Accuracy

The computed inverse X satisfies

$$|XA - I| \leq c(n)\epsilon|X||A|,$$

where $c(n)$ is a modest linear function of n , and ϵ is the **machine precision**.

Note that a similar bound for $|AX - I|$ cannot be guaranteed, although it is almost always satisfied.

The computed inverse satisfies the forward error bound

$$|X - A^{-1}| \leq c(n)\epsilon|A^{-1}||A||X|.$$

See Du Croz and Higham (1992).

8 Further Comments

The total number of floating-point operations is approximately $\frac{1}{3}n^3$.

The complex analogue of this function is nag_lapack_ztftri (f07wx).

9 Example

This example computes the inverse of the matrix A , where

$$A = \begin{pmatrix} 4.30 & 0.00 & 0.00 & 0.00 \\ -3.96 & -4.87 & 0.00 & 0.00 \\ 0.40 & 0.31 & -8.02 & 0.00 \\ -0.27 & 0.07 & -5.95 & 0.12 \end{pmatrix}$$

and is stored using RFP format.

9.1 Program Text

```
function f07wk_example

fprintf('f07wk example results\n\n');

% Symmetric matrix in RFP format
transr = 'n';
uplo   = 'l';
diag   = 'n';
ar = [-8.02 -5.95;
       4.30  0.12;
      -3.96 -4.87;
       0.40  0.31;
      -0.27  0.07];
n = nag_int(4);
n2 = (n*(n+1))/2;
ar = reshape(ar,[n2,1]);

% Compute inverse of a
[ar, info] = f07wk( ...
    transr, uplo, diag, n, ar);

if info == 0
    % Convert inverse to full array form, and print it
    [a, info] = f01vg(transr, uplo, n, ar);
    fprintf('\n');
    [ifail] = x04ca(uplo, 'n', a, 'Inverse');
else
    fprintf('\na is singular.\n');
end
```

9.2 Program Results

f07wk example results

Inverse	1	2	3	4
1	0.2326			
2	-0.1891	-0.2053		
3	0.0043	-0.0079	-0.1247	
4	0.8463	-0.2738	-6.1825	8.3333
