NAG Toolbox

nag_lapack_dsytri (f07mj)

1 Purpose

nag_lapack_dsytri (f07mj) computes the inverse of a real symmetric indefinite matrix A, where A has been factorized by nag_lapack_dsytrf (f07md).

2 Syntax

```
[a, info] = nag_lapack_dsytri(uplo, a, ipiv, 'n', n)
[a, info] = f07mj(uplo, a, ipiv, 'n', n)
```

3 Description

nag_lapack_dsytri (f07mj) is used to compute the inverse of a real symmetric indefinite matrix A, the function must be preceded by a call to nag_lapack_dsytrf (f07md), which computes the Bunch–Kaufman factorization of A.

If **uplo** $=$ 'U', $A = PUDU^{\mathrm{T}}P^{\mathrm{T}}$ and A^{-1} is computed by solving $U^{\mathrm{T}}P^{\mathrm{T}}XPU = D^{-1}$ for X.

If **uplo** $=$ 'L', $A = PLDL^{\mathrm{T}}P^{\mathrm{T}}$ and A^{-1} is computed by solving $L^{\mathrm{T}}P^{\mathrm{T}}XPL = D^{-1}$ for X.

4 References

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

5 Parameters

5.1 Compulsory Input Parameters

1: **uplo** – CHARACTER(1)

Specifies how A has been factorized.

uplo $=$ 'U'
$A = PUDU^{\mathrm{T}}P^{\mathrm{T}}$, where U is upper triangular.

uplo $=$ 'L'
$A = PLDL^{\mathrm{T}}P^{\mathrm{T}}$, where L is lower triangular.

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

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

The first dimension of the array **a** must be at least $\max(1,\mathbf{n})$.

The second dimension of the array **a** must be at least $\max(1,\mathbf{n})$.

Details of the factorization of A, as returned by nag_lapack_dsytrf (f07md).

3: **ipiv**$(:)$ – INTEGER array

The dimension of the array **ipiv** must be at least $\max(1,\mathbf{n})$

Details of the interchanges and the block structure of D, as returned by nag_lapack_dsytrf (f07md).

5.2 Optional Input Parameters

1: **n** – INTEGER

Default: the first dimension of the array **a** and the second dimension of the arrays **a**, **ipiv**.

n, the order of the matrix A.

Constraint: $\mathbf{n} \geq 0$.

5.3 Output Parameters

1: $\mathbf{a}(lda,:)$ – REAL (KIND=nag_wp) array

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

The second dimension of the array **a** will be $\max(1,\mathbf{n})$.

The factorization stores the n by n symmetric matrix A^{-1}.

If **uplo** = 'U', the upper triangle of A^{-1} is stored in the upper triangular part of the array.

If **uplo** = 'L', the lower triangle of A^{-1} is stored in the lower triangular part of the array.

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*)

Element $\langle$*value*$\rangle$ of the diagonal is exactly zero. D is singular and the inverse of A cannot be computed.

7 Accuracy

The computed inverse X satisfies a bound of the form

if **uplo** = 'U', $|DU^{\mathrm{T}}P^{\mathrm{T}}XPU - I| \leq c(n)\epsilon\left(|D||U^{\mathrm{T}}|P^{\mathrm{T}}|X|P|U| + |D|\left|D^{-1}\right|\right)$;

if **uplo** = 'L', $|DL^{\mathrm{T}}P^{\mathrm{T}}XPL - I| \leq c(n)\epsilon\left(|D||L^{\mathrm{T}}|P^{\mathrm{T}}|X|P|L| + |D|\left|D^{-1}\right|\right)$,

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

8 Further Comments

The total number of floating-point operations is approximately $\frac{2}{3}n^3$.

The complex analogues of this function are nag_lapack_zhetri (f07mw) for Hermitian matrices and nag_lapack_zsytri (f07nw) for symmetric matrices.

9 Example

This example computes the inverse of the matrix A, where

$$A = \begin{pmatrix} 2.07 & 3.87 & 4.20 & -1.15 \\ 3.87 & -0.21 & 1.87 & 0.63 \\ 4.20 & 1.87 & 1.15 & 2.06 \\ -1.15 & 0.63 & 2.06 & -1.81 \end{pmatrix}.$$

Here A is symmetric indefinite and must first be factorized by nag_lapack_dsytrf (f07md).

9.1 Program Text

```
function f07mj_example

fprintf('f07mj example results\n\n');

% Indefinite matrix A (lower triangular part stored)
uplo = 'L';
a = [ 2.07,  0,    0,     0;
      3.87, -0.21, 0,     0;
      4.20,  1.87, 1.15,  0;
     -1.15,  0.63, 2.06, -1.81];

% Factorize
[af, ipiv, info] = f07md( ...
                          uplo, a);

% Invert
[ainv, info] = f07mj( ...
                      uplo, af, ipiv);

[ifail] = x04ca( ...
                 uplo, 'N', ainv, 'Inverse');
```

9.2 Program Results

```
f07mj example results

 Inverse
             1          2          3          4
 1      0.7485
 2      0.5221    -0.1605
 3     -1.0058    -0.3131     1.3501
 4     -1.4386    -0.7440     2.0667     2.4547
```