NAG Toolbox

nag_lapack_dsptri (f07pj)

1 Purpose

nag_lapack_dsptri (f07pj) computes the inverse of a real symmetric indefinite matrix A, where A has been factorized by nag_lapack_dsptrf (f07pd), using packed storage.

2 Syntax

```
[ap, info] = nag_lapack_dsptri(uplo, ap, ipiv, 'n', n)
[ap, info] = f07pj(uplo, ap, ipiv, 'n', n)
```

3 Description

nag_lapack_dsptri (f07pj) is used to compute the inverse of a real symmetric indefinite matrix A, the function must be preceded by a call to nag_lapack_dsptrf (f07pd), which computes the Bunch–Kaufman factorization of A, using packed storage.

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

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

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: **ap**(:) – REAL (KIND=nag_wp) array

The dimension of the array **ap** must be at least $\max(1, \mathbf{n} \times (\mathbf{n}+1)/2)$

The factorization of A stored in packed form, as returned by nag_lapack_dsptrf (f07pd).

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_dsptrf (f07pd).

5.2 Optional Input Parameters

1: **n** – INTEGER

Default: the dimension of the array **ipiv**.

n, the order of the matrix A.

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

5.3 Output Parameters

1: **ap**(:) – REAL (KIND=nag_wp) array

The dimension of the array **ap** will be $\max(1, \mathbf{n} \times (\mathbf{n}+1)/2)$

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

More precisely,

if **uplo** = 'U', the upper triangle of A^{-1} must be stored with element A_{ij} in $\mathbf{ap}(i + j(j-1)/2)$ for $i \leq j$;

if **uplo** = 'L', the lower triangle of A^{-1} must be stored with element A_{ij} in $\mathbf{ap}(i + (2n-j)(j-1)/2)$ for $i \geq j$.

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 ⟨*value*⟩ 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', $|DU^{\mathrm{T}}P^{\mathrm{T}}XPU - I| \leq c(n)\epsilon\big(|D||U^{\mathrm{T}}|P^{\mathrm{T}}|X|P|U| + |D|\big|D^{-1}\big|\big)$;

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

$c(n)$ 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_zhptri (f07pw) for Hermitian matrices and nag_lapack_zsptri (f07qw) 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, stored in packed form, and must first be factorized by nag_lapack_dsptrf (f07pd).

9.1 Program Text

```
     function f07pj_example

fprintf('f07pj example results\n\n');

% Indefinite matrix A (lower triangular part stored in packed format)
uplo = 'L';
n = nag_int(4);
ap = [2.07;   3.87;   4.20;   -1.15;
             -0.21;   1.87;    0.63;
                      1.15;    2.06;
                              -1.81];

% Factorize
[apf, ipiv, info] = f07pd( ...
                           uplo, n, ap);

% Invert
[ainv, info] = f07pj( ...
                      uplo, apf, ipiv);

[ifail] = x04cc( ...
                 uplo, 'Non-unit', n, ainv, 'Inverse');
```

9.2 Program Results

```
     f07pj 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
```