NAG Toolbox

nag_lapack_zsytri (f07nw)

1 Purpose

nag_lapack_zsytri (f07nw) computes the inverse of a complex symmetric matrix A, where A has been factorized by nag_lapack_zsytrf (f07nr).

2 Syntax

```
[a, info] = nag_lapack_zsytri(uplo, a, ipiv, 'n', n)
[a, info] = f07nw(uplo, a, ipiv, 'n', n)
```

3 Description

nag_lapack_zsytri (f07nw) is used to compute the inverse of a complex symmetric matrix A, the function must be preceded by a call to nag_lapack_zsytrf (f07nr), 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**$(lda, :)$ – COMPLEX (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_zsytrf (f07nr).

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_zsytrf (f07nr).

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: **a**$(lda,:)$ – COMPLEX (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', $|DU^{\mathrm{T}}P^{\mathrm{T}}XPU - I| \leq c(n)\epsilon\big(|D||U^{\mathrm{T}}|P^{\mathrm{T}}|X|P|U| + |D|\big|D^{-1}\big|\big)$;

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 real floating-point operations is approximately $\frac{8}{3}n^3$.

The real analogue of this function is nag_lapack_dsytri (f07mj).

9 Example

This example computes the inverse of the matrix A, where

$$A = \begin{pmatrix} -0.39 - 0.71i & 5.14 - 0.64i & -7.86 - 2.96i & 3.80 + 0.92i \\ 5.14 - 0.64i & 8.86 + 1.81i & -3.52 + 0.58i & 5.32 - 1.59i \\ -7.86 - 2.96i & -3.52 + 0.58i & -2.83 - 0.03i & -1.54 - 2.86i \\ 3.80 + 0.92i & 5.32 - 1.59i & -1.54 - 2.86i & -0.56 + 0.12i \end{pmatrix}.$$

Here A is symmetric and must first be factorized by nag_lapack_zsytrf (f07nr).

9.1 Program Text

```
     function f07nw_example

fprintf('f07nw example results\n\n');

% Symmetric indefinite matrix A (Upper triangular part stored)
uplo = 'L';
a = [-0.39 - 0.71i,  0    + 0i,     0    + 0i,     0    + 0i;
      5.14 - 0.64i,  8.86 + 1.81i,  0    + 0i,     0    + 0i;
     -7.86 - 2.96i, -3.52 + 0.58i, -2.83 - 0.03i,  0    + 0i;
      3.80 + 0.92i,  5.32 - 1.59i, -1.54 - 2.86i, -0.56 + 0.12i];

% Factorize
[af, ipiv, info] = f07nr( ...
                          uplo, a);

% Invert
[ainv, info] = f07nw( ...
                      uplo, af, ipiv);

[ifail] = x04da( ...
                 uplo, 'Non-unit', ainv, 'Inverse');
```

9.2 Program Results

```
     f07nw example results

 Inverse
          1       2       3       4
 1  -0.1562
    -0.1014

 2   0.0400  0.0946
     0.1527 -0.1475

 3   0.0550 -0.0326 -0.1320
     0.0845 -0.1370 -0.0102

 4   0.2162 -0.0995 -0.1793 -0.2269
    -0.0742 -0.0461  0.1183  0.2383
```