NAG Toolbox

nag_lapack_zsytrf (f07nr)

1 Purpose

nag_lapack_zsytrf (f07nr) computes the Bunch–Kaufman factorization of a complex symmetric matrix.

2 Syntax

```
[a, ipiv, info] = nag_lapack_zsytrf(uplo, a, 'n', n)
[a, ipiv, info] = f07nr(uplo, a, 'n', n)
```

3 Description

nag_lapack_zsytrf (f07nr) factorizes a complex symmetric matrix A, using the Bunch–Kaufman diagonal pivoting method. A is factorized as either $A = PUDU^{\mathrm{T}}P^{\mathrm{T}}$ if **uplo** = 'U' or $A = PLDL^{\mathrm{T}}P^{\mathrm{T}}$ if **uplo** = 'L', where P is a permutation matrix, U (or L) is a unit upper (or lower) triangular matrix and D is a symmetric block diagonal matrix with 1 by 1 and 2 by 2 diagonal blocks; U (or L) has 2 by 2 unit diagonal blocks corresponding to the 2 by 2 blocks of D. Row and column interchanges are performed to ensure numerical stability while preserving symmetry.

4 References

Golub G H and Van Loan C F (1996) *Matrix Computations* (3rd Edition) Johns Hopkins University Press, Baltimore

5 Parameters

5.1 Compulsory Input Parameters

1: **uplo** – CHARACTER(1)

Specifies whether the upper or lower triangular part of A is stored and how A is to be factorized.

uplo = 'U'
: The upper triangular part of A is stored and A is factorized as $PUDU^{\mathrm{T}}P^{\mathrm{T}}$, where U is upper triangular.

uplo = 'L'
: The lower triangular part of A is stored and A is factorized as $PLDL^{\mathrm{T}}P^{\mathrm{T}}$, 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, **n**).

The second dimension of the array **a** must be at least max(1, **n**).

The n by n symmetric indefinite matrix A.

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

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

5.2 Optional Input Parameters

1: **n** – INTEGER

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

n, the order of the matrix A.

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

5.3 Output Parameters

1: $\mathbf{a}(lda,:)$ – 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 upper or lower triangle of A stores details of the block diagonal matrix D and the multipliers used to obtain the factor U or L as specified by **uplo**.

2: $\mathbf{ipiv}(:)$ – INTEGER array

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

Details of the interchanges and the block structure of D. More precisely,

if $\mathbf{ipiv}(i) = k > 0$, d_{ii} is a 1 by 1 pivot block and the ith row and column of A were interchanged with the kth row and column;

if $\mathbf{uplo} = \text{'U'}$ and $\mathbf{ipiv}(i-1) = \mathbf{ipiv}(i) = -l < 0$, $\begin{pmatrix} d_{i-1,i-1} & \bar{d}_{i,i-1} \\ \bar{d}_{i,i-1} & d_{ii} \end{pmatrix}$ is a 2 by 2 pivot block and the $(i-1)$th row and column of A were interchanged with the lth row and column;

if $\mathbf{uplo} = \text{'L'}$ and $\mathbf{ipiv}(i) = \mathbf{ipiv}(i+1) = -m < 0$, $\begin{pmatrix} d_{ii} & d_{i+1,i} \\ d_{i+1,i} & d_{i+1,i+1} \end{pmatrix}$ is a 2 by 2 pivot block and the $(i+1)$th row and column of A were interchanged with the mth row and column.

3: **info** – INTEGER

$\mathbf{info} = 0$ unless the function detects an error (see Section 6).

6 Error Indicators and Warnings

$\mathbf{info} < 0$

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

$\mathbf{info} > 0$ (*warning*)

Element $\langle value \rangle$ of the diagonal is exactly zero. The factorization has been completed, but the block diagonal matrix D is exactly singular, and division by zero will occur if it is used to solve a system of equations.

7 Accuracy

If $\mathbf{uplo} = \text{'U'}$, the computed factors U and D are the exact factors of a perturbed matrix $A + E$, where

$$|E| \leq c(n)\epsilon P|U||D|\left|U^{\mathrm{T}}\right|P^{\mathrm{T}},$$

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

If **uplo** = 'L', a similar statement holds for the computed factors L and D.

8 Further Comments

The elements of D overwrite the corresponding elements of A; if D has 2 by 2 blocks, only the upper or lower triangle is stored, as specified by **uplo**.

The unit diagonal elements of U or L and the 2 by 2 unit diagonal blocks are not stored. The remaining elements of U or L are stored in the corresponding columns of the array **a**, but additional row interchanges must be applied to recover U or L explicitly (this is seldom necessary). If **ipiv**$(i) = i$, for $i = 1, 2, \ldots, n$, then U or L is stored explicitly (except for its unit diagonal elements which are equal to 1).

The total number of real floating-point operations is approximately $\frac{4}{3}n^3$.

A call to nag_lapack_zsytrf (f07nr) may be followed by calls to the functions:

nag_lapack_zsytrs (f07ns) to solve $AX = B$;

nag_lapack_zsycon (f07nu) to estimate the condition number of A;

nag_lapack_zsytri (f07nw) to compute the inverse of A.

The real analogue of this function is nag_lapack_dsytrf (f07md).

9 Example

This example computes the Bunch–Kaufman factorization of the matrix A, where

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

9.1 Program Text

```
function f07nr_example

fprintf('f07nr 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];

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

[ifail] = x04da( ...
                 uplo, 'Non-unit', af, 'Details of factorization');

fprintf('\nPivot indices\n   ');
fprintf('%11d', ipiv);
fprintf('\n');
```

9.2 Program Results

```
f07nr example results

 Details of factorization
             1          2          3          4
 1     -0.3900
       -0.7100

 2     -7.8600    -2.8300
```

```
       -2.9600    -0.0300

 3      0.5279    -0.6078     4.4079
       -0.3715     0.2811     5.3991

 4      0.4426    -0.4823    -0.1071    -2.0954
        0.1936     0.0150    -0.3157    -2.2011

Pivot indices
            -3          -3           3           4
```