# NAG Toolbox

# nag_lapack_zhetrf (f07mr)

## 1 Purpose

nag_lapack_zhetrf (f07mr) computes the Bunch–Kaufman factorization of a complex Hermitian indefinite matrix.

## 2 Syntax

```
[a, ipiv, info] = nag_lapack_zhetrf(uplo, a, 'n', n)
[a, ipiv, info] = f07mr(uplo, a, 'n', n)
```

## 3 Description

nag_lapack_zhetrf (f07mr) factorizes a complex Hermitian matrix $A$, using the Bunch–Kaufman diagonal pivoting method. $A$ is factorized either as $A = PUDU^{\mathrm{H}}P^{\mathrm{T}}$ if **uplo** = 'U' or $A = PLDL^{\mathrm{H}}P^{\mathrm{T}}$ if **uplo** = 'L', where $P$ is a permutation matrix, $U$ (or $L$) is a unit upper (or lower) triangular matrix and $D$ is an Hermitian 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 keeping the matrix Hermitian.

This method is suitable for Hermitian matrices which are not known to be positive definite. If $A$ is in fact positive definite, no interchanges are performed and no 2 by 2 blocks occur in $D$.

## 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 $PUDU^{\mathrm{H}}P^{\mathrm{T}}$, where $U$ is upper triangular.

**uplo** = 'L'
: The lower triangular part of $A$ is stored and $A$ is factorized as $PLDL^{\mathrm{H}}P^{\mathrm{T}}$, 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 $\max(1, \mathbf{n})$.

The $n$ by $n$ Hermitian 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: **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: **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 $i$th row and column of $A$ were interchanged with the $k$th row and column;

if **uplo** = '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 $l$th row and column;

if **uplo** = '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 $m$th row and column.

3: **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. 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 **uplo** = 'U', the computed factors $U$ and $D$ are the exact factors of a perturbed matrix $A + E$, where

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

$c(n)$ is a modest linear function of $n$, and $\epsilon$ 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 $i = 1, 2, \ldots, n$ (as is the case when $A$ is positive definite), 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_zhetrf (f07mr) may be followed by calls to the functions:

nag_lapack_zhetrs (f07ms) to solve $AX = B$;

nag_lapack_zhecon (f07mu) to estimate the condition number of $A$;

nag_lapack_zhetri (f07mw) 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

$$A = \begin{pmatrix} -1.36 + 0.00i & 1.58 + 0.90i & 2.21 - 0.21i & 3.91 + 1.50i \\ 1.58 - 0.90i & -8.87 + 0.00i & -1.84 - 0.03i & -1.78 + 1.18i \\ 2.21 + 0.21i & -1.84 + 0.03i & -4.63 + 0.00i & 0.11 + 0.11i \\ 3.91 - 1.50i & -1.78 - 1.18i & 0.11 - 0.11i & -1.84 + 0.00i \end{pmatrix}.$$

### 9.1 Program Text

```
function f07mr_example

fprintf('f07mr example results\n\n');

% Hermitian indefinite matrix A (Lower triangular part stored)
uplo = 'L';
a = [-1.36 + 0i,     0    + 0i,     0    + 0i,      0    + 0i;
      1.58 - 0.90i, -8.87 + 0i,     0    + 0i,      0    + 0i;
      2.21 + 0.21i, -1.84 + 0.03i, -4.63 + 0i,      0    + 0i;
      3.91 - 1.50i, -1.78 - 1.18i,  0.11 - 0.11i,  -1.84 + 0i];

% Factorize
[af, ipiv, info] = f07mr( ...
                          uplo, a);

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

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

## 9.2 Program Results

```
     f07mr example results

 Details of factorization
            1          2          3          4
 1     -1.3600
        0.0000

 2      3.9100    -1.8400
       -1.5000     0.0000

 3      0.3100     0.5637    -5.4176
        0.0433     0.2850     0.0000

 4     -0.1518     0.3397     0.2997    -7.1028
        0.3743     0.0303     0.1578     0.0000

Pivot indices
           -4         -4          3          4
```