NAG Library Routine Document

F07NRF (ZSYTRF)

Note: before using this routine, please read the Users' Note for your implementation to check the interpretation of **bold italicised** terms and other implementation-dependent details.

1 Purpose

F07NRF (ZSYTRF) computes the Bunch–Kaufman factorization of a complex symmetric matrix.

2 Specification

```
SUBROUTINE F07NRF (UPLO, N, A, LDA, IPIV, WORK, LWORK, INFO)
INTEGER              N, LDA, IPIV(*), LWORK, INFO
COMPLEX (KIND=nag_wp) A(LDA,*), WORK(max(1,LWORK))
CHARACTER(1)         UPLO
```

The routine may be called by its LAPACK name *zsytrf*.

3 Description

F07NRF (ZSYTRF) 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 Arguments

1:   UPLO – CHARACTER(1)   *Input*

On entry: 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:   N – INTEGER   *Input*

On entry: n, the order of the matrix A.

Constraint: N $\geq$ 0.

3:     A(LDA, ∗) – COMPLEX (KIND=nag_wp) array   *Input/Output*

   Note: the second dimension of the array A must be at least max(1, N).

   On entry: 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.

   On exit: the upper or lower triangle of A is overwritten by details of the block diagonal matrix D
   and the multipliers used to obtain the factor U or L as specified by UPLO.

4:     LDA – INTEGER   *Input*

   On entry: the first dimension of the array A as declared in the (sub)program from which F07NRF
   (ZSYTRF) is called.

   Constraint: $\text{LDA} \geq \max(1, N)$.

5:     IPIV(∗) – INTEGER array   *Output*

   Note: the dimension of the array IPIV must be at least max(1, N).

   On exit: details of the interchanges and the block structure of D. More precisely,

      if $\text{IPIV}(i) = k > 0$, 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 UPLO = 'U' and $\text{IPIV}(i-1) = \text{IPIV}(i) = -l < 0$, $\begin{pmatrix} d_{i-1,i-1} & \bar{d}_{i,i-1} \\ \bar{d}_{i,i-1} & d_{ii} \end{pmatrix}$ 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 UPLO = 'L' and $\text{IPIV}(i) = \text{IPIV}(i+1) = -m < 0$, $\begin{pmatrix} d_{ii} & d_{i+1,i} \\ d_{i+1,i} & d_{i+1,i+1} \end{pmatrix}$ 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.

6:     WORK(max(1, LWORK)) – COMPLEX (KIND=nag_wp) array   *Workspace*

   On exit: if INFO = 0, WORK(1) contains the minimum value of LWORK required for optimum
   performance.

7:     LWORK – INTEGER   *Input*

   On entry: the dimension of the array WORK as declared in the (sub)program from which
   F07NRF (ZSYTRF) is called, unless LWORK = −1, in which case a workspace query is
   assumed and the routine only calculates the optimal dimension of WORK (using the formula
   given below).

   Suggested value: for optimum performance LWORK should be at least $N \times nb$, where nb is the
   block size.

   Constraint: $\text{LWORK} \geq 1$ or $\text{LWORK} = -1$.

8:     INFO – INTEGER   *Output*

   On exit: INFO = 0 unless the routine 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

> 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

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

$c(n)$ 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 Parallelism and Performance

F07NRF (ZSYTRF) makes calls to BLAS and/or LAPACK routines, which may be threaded within the vendor library used by this implementation. Consult the documentation for the vendor library for further information.

Please consult the X06 Chapter Introduction for information on how to control and interrogate the OpenMP environment used within this routine. Please also consult the Users' Note for your implementation for any additional implementation-specific information.

9 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 F07NRF (ZSYTRF) may be followed by calls to the routines:

> F07NSF (ZSYTRS) to solve $AX = B$;

> F07NUF (ZSYCON) to estimate the condition number of A;

> F07NWF (ZSYTRI) to compute the inverse of A.

The real analogue of this routine is F07MDF (DSYTRF).

10 Example

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

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

10.1 Program Text

```
      Program f07nrfe

!     F07NRF Example Program Text

!     Mark 26 Release. NAG Copyright 2016.

!     .. Use Statements ..
      Use nag_library, Only: nag_wp, x04dbf, zsytrf
!     .. Implicit None Statement ..
      Implicit None
!     .. Parameters ..
      Integer, Parameter                 :: nin = 5, nout = 6
!     .. Local Scalars ..
      Integer                            :: i, ifail, info, lda, lwork, n
      Character (1)                      :: uplo
!     .. Local Arrays ..
      Complex (Kind=nag_wp), Allocatable :: a(:,:), work(:)
      Integer, Allocatable               :: ipiv(:)
      Character (1)                      :: clabs(1), rlabs(1)
!     .. Executable Statements ..
      Write (nout,*) 'F07NRF Example Program Results'
!     Skip heading in data file
      Read (nin,*)
      Read (nin,*) n
      lda = n
      lwork = 64*n
      Allocate (a(lda,n),work(lwork),ipiv(n))

!     Read A from data file

      Read (nin,*) uplo
      If (uplo=='U') Then
        Read (nin,*)(a(i,i:n),i=1,n)
      Else If (uplo=='L') Then
        Read (nin,*)(a(i,1:i),i=1,n)
      End If

!     Factorize A
!     The NAG name equivalent of zsytrf is f07nrf
      Call zsytrf(uplo,n,a,lda,ipiv,work,lwork,info)

      Write (nout,*)
      Flush (nout)

!     Print details of factorization

!     ifail: behaviour on error exit
!            =0 for hard exit, =1 for quiet-soft, =-1 for noisy-soft
      ifail = 0
      Call x04dbf(uplo,'Nonunit',n,n,a,lda,'Bracketed','F7.4',          &
        'Details of factorization','Integer',rlabs,'Integer',clabs,80,0,ifail)

!     Print pivot indices

      Write (nout,*)
      Write (nout,*) 'IPIV'
      Write (nout,99999) ipiv(1:n)

      If (info/=0) Then
        Write (nout,*) 'The factor D is singular'
      End If

99999 Format ((1X,I12,3I18))
      End Program f07nrfe
```

10.2 Program Data

```
F07NRF Example Program Data
  4                                                       :Value of N
  'L'                                                     :Value of UPLO
 (-0.39,-0.71)
 ( 5.14,-0.64) ( 8.86, 1.81)
 (-7.86,-2.96) (-3.52, 0.58) (-2.83,-0.03)
 ( 3.80, 0.92) ( 5.32,-1.59) (-1.54,-2.86) (-0.56, 0.12)  :End of matrix A
```

10.3 Program Results

```
F07NRF Example Program Results

Details of factorization
                    1                   2                   3                   4
1 (-0.3900,-0.7100)
2 (-7.8600,-2.9600) (-2.8300,-0.0300)
3 ( 0.5279,-0.3715) (-0.6078, 0.2811) ( 4.4079, 5.3991)
4 ( 0.4426, 0.1936) (-0.4823, 0.0150) (-0.1071,-0.3157) (-2.0954,-2.2011)

IPIV
            -3                  -3                   3                   4
```