NAG Library Routine Document

F07NSF (ZSYTRS)

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

F07NSF (ZSYTRS) solves a complex symmetric system of linear equations with multiple right-hand sides,

$$AX = B,$$

where A has been factorized by F07NRF (ZSYTRF).

2 Specification

```
SUBROUTINE F07NSF (UPLO, N, NRHS, A, LDA, IPIV, B, LDB, INFO)
INTEGER              N, NRHS, LDA, IPIV(*), LDB, INFO
COMPLEX (KIND=nag_wp) A(LDA,*), B(LDB,*)
CHARACTER(1)         UPLO
```

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

3 Description

F07NSF (ZSYTRS) is used to solve a complex symmetric system of linear equations $AX = B$, this routine must be preceded by a call to F07NRF (ZSYTRF) which computes the Bunch–Kaufman factorization of A.

If UPLO = 'U', $A = PUDU^{\mathrm{T}}P^{\mathrm{T}}$, where P is a permutation matrix, U is an upper triangular matrix and D is a symmetric block diagonal matrix with 1 by 1 and 2 by 2 blocks; the solution X is computed by solving $PUDY = B$ and then $U^{\mathrm{T}}P^{\mathrm{T}}X = Y$.

If UPLO = 'L', $A = PLDL^{\mathrm{T}}P^{\mathrm{T}}$, where L is a lower triangular matrix; the solution X is computed by solving $PLDY = B$ and then $L^{\mathrm{T}}P^{\mathrm{T}}X = Y$.

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 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: N – INTEGER *Input*

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

Constraint: $N \geq 0$.

3: NRHS – INTEGER *Input*

On entry: r, the number of right-hand sides.

Constraint: $NRHS \geq 0$.

4: $A(LDA, *)$ – COMPLEX (KIND=nag_wp) array *Input*

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

On entry: details of the factorization of A, as returned by F07NRF (ZSYTRF).

5: LDA – INTEGER *Input*

On entry: the first dimension of the array A as declared in the (sub)program from which F07NSF (ZSYTRS) is called.

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

6: $IPIV(*)$ – INTEGER array *Input*

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

On entry: details of the interchanges and the block structure of D, as returned by F07NRF (ZSYTRF).

7: $B(LDB, *)$ – COMPLEX (KIND=nag_wp) array *Input/Output*

Note: the second dimension of the array B must be at least $\max(1, NRHS)$.

On entry: the n by r right-hand side matrix B.

On exit: the n by r solution matrix X.

8: LDB – INTEGER *Input*

On entry: the first dimension of the array B as declared in the (sub)program from which F07NSF (ZSYTRS) is called.

Constraint: $LDB \geq \max(1, N)$.

9: 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.

7 Accuracy

For each right-hand side vector b, the computed solution x is the exact solution of a perturbed system of equations $(A + E)x = b$, where

if UPLO = 'U', $|E| \leq c(n)\epsilon P|U||D||U^{\mathrm{T}}|P^{\mathrm{T}}$;

if UPLO = 'L', $|E| \leq c(n)\epsilon P|L||D||L^{\mathrm{T}}|P^{\mathrm{T}}$,

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

If $\hat{x}$ is the true solution, then the computed solution x satisfies a forward error bound of the form

$$\frac{\|x-\hat{x}\|_\infty}{\|x\|_\infty} \leq c(n)\,\mathrm{cond}(A,x)\epsilon$$

where $\mathrm{cond}(A,x) = \left\||A^{-1}||A||x|\right\|_\infty / \|x\|_\infty \leq \mathrm{cond}(A) = \left\||A^{-1}||A|\right\|_\infty \leq \kappa_\infty(A)$.

Note that $\mathrm{cond}(A,x)$ can be much smaller than $\mathrm{cond}(A)$.

Forward and backward error bounds can be computed by calling F07NVF (ZSYRFS), and an estimate for $\kappa_\infty(A)$ ($=\kappa_1(A)$) can be obtained by calling F07NUF (ZSYCON).

8 Parallelism and Performance

F07NSF (ZSYTRS) 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 total number of real floating-point operations is approximately $8n^2r$.

This routine may be followed by a call to F07NVF (ZSYRFS) to refine the solution and return an error estimate.

The real analogue of this routine is F07MEF (DSYTRS).

10 Example

This example solves the system of equations $AX = B$, where

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

and

$$B = \begin{pmatrix} -55.64+41.22i & -19.09-35.97i \\ -48.18+66.00i & -12.08-27.02i \\ -0.49-\ \ 1.47i & 6.95+20.49i \\ -6.43+19.24i & -4.59-35.53i \end{pmatrix}.$$

Here A is symmetric and must first be factorized by F07NRF (ZSYTRF).

10.1 Program Text

```
    Program f07nsfe

!     F07NSF Example Program Text

!     Mark 26 Release. NAG Copyright 2016.

!     .. Use Statements ..
      Use nag_library, Only: nag_wp, x04dbf, zsytrf, zsytrs
!     .. Implicit None Statement ..
      Implicit None
!     .. Parameters ..
      Integer, Parameter               :: nin = 5, nout = 6
```

```
!     .. Local Scalars ..
      Integer                          :: i, ifail, info, lda, ldb, lwork, n,  &
                                          nrhs
      Character (1)                    :: uplo
!     .. Local Arrays ..
      Complex (Kind=nag_wp), Allocatable :: a(:,:), b(:,:), work(:)
      Integer, Allocatable             :: ipiv(:)
      Character (1)                    :: clabs(1), rlabs(1)
!     .. Executable Statements ..
      Write (nout,*) 'F07NSF Example Program Results'
!     Skip heading in data file
      Read (nin,*)
      Read (nin,*) n, nrhs
      lda = n
      ldb = n
      lwork = 64*n
      Allocate (a(lda,n),b(ldb,nrhs),work(lwork),ipiv(n))

!     Read A and B 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
      Read (nin,*)(b(i,1:nrhs),i=1,n)

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

      Write (nout,*)
      Flush (nout)
      If (info==0) Then

!       Compute solution
!       The NAG name equivalent of zsytrs is f07nsf
        Call zsytrs(uplo,n,nrhs,a,lda,ipiv,b,ldb,info)

!       Print solution

!       ifail: behaviour on error exit
!              =0 for hard exit, =1 for quiet-soft, =-1 for noisy-soft
        ifail = 0
        Call x04dbf('General',' ',n,nrhs,b,ldb,'Bracketed','F7.4',             &
          'Solution(s)','Integer',rlabs,'Integer',clabs,80,0,ifail)

      Else
        Write (nout,*) 'The factor D is singular'
      End If

    End Program f07nsfe
```

10.2 Program Data

```
F07NSF Example Program Data
  4  2                                                   :Values of N and NRHS
  '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
 (-55.64, 41.22) (-19.09,-35.97)
 (-48.18, 66.00) (-12.08,-27.02)
 ( -0.49, -1.47) (  6.95, 20.49)
 ( -6.43, 19.24) ( -4.59,-35.53)                         :End of matrix B
```

10.3 Program Results

```
F07NSF Example Program Results

Solution(s)
                      1                   2
1  ( 1.0000,-1.0000) (-2.0000,-1.0000)
2  (-2.0000, 5.0000) ( 1.0000,-3.0000)
3  ( 3.0000,-2.0000) ( 3.0000, 2.0000)
4  (-4.0000, 3.0000) (-1.0000, 1.0000)
```