NAG Library Routine Document

F07FNF (ZPOSV)

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

F07FNF (ZPOSV) computes the solution to a complex system of linear equations

$$AX = B,$$

where A is an n by n Hermitian positive definite matrix and X and B are n by r matrices.

2 Specification

```
SUBROUTINE F07FNF (UPLO, N, NRHS, A, LDA, B, LDB, INFO)
INTEGER                 N, NRHS, LDA, LDB, INFO
COMPLEX (KIND=nag_wp) A(LDA,*), B(LDB,*)
CHARACTER(1)            UPLO
```

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

3 Description

F07FNF (ZPOSV) uses the Cholesky decomposition to factor A as $A = U^{\mathrm{H}}U$ if UPLO = 'U' or $A = LL^{\mathrm{H}}$ if UPLO = 'L', where U is an upper triangular matrix and L is a lower triangular matrix. The factored form of A is then used to solve the system of equations $AX = B$.

4 References

Anderson E, Bai Z, Bischof C, Blackford S, Demmel J, Dongarra J J, Du Croz J J, Greenbaum A, Hammarling S, McKenney A and Sorensen D (1999) *LAPACK Users' Guide* (3rd Edition) SIAM, Philadelphia http://www.netlib.org/lapack/lug

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: if UPLO = 'U', the upper triangle of A is stored.

If UPLO = 'L', the lower triangle of A is stored.

Constraint: UPLO = 'U' or 'L'.

2: N – INTEGER *Input*

On entry: n, the number of linear equations, i.e., the order of the matrix A.

Constraint: N $\geq$ 0.

3: NRHS – INTEGER *Input*

On entry: r, the number of right-hand sides, i.e., the number of columns of the matrix B.

Constraint: NRHS $\geq$ 0.

4: 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 Hermitian 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: if INFO = 0, the factor U or L from the Cholesky factorization $A = U^{\mathrm{H}}U$ or $A = LL^{\mathrm{H}}$.

5: LDA – INTEGER *Input*

On entry: the first dimension of the array A as declared in the (sub)program from which F07FNF (ZPOSV) is called.

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

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

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

Note: to solve the equations $Ax = b$, where b is a single right-hand side, B may be supplied as a one-dimensional array with length LDB = max(1, N).

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

On exit: if INFO = 0, the n by r solution matrix X.

7: LDB – INTEGER *Input*

On entry: the first dimension of the array B as declared in the (sub)program from which F07FNF (ZPOSV) is called.

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

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

The leading minor of order $\langle$*value*$\rangle$ of A is not positive definite, so the factorization could not be completed, and the solution has not been computed.

7 Accuracy

The computed solution for a single right-hand side, $\hat{x}$, satisfies an equation of the form

$$(A + E)\hat{x} = b,$$

where

$$\|E\|_1 = O(\epsilon)\|A\|_1$$

and ϵ is the ***machine precision***. An approximate error bound for the computed solution is given by

$$\frac{\|\hat{x}-x\|_1}{\|x\|_1} \leq \kappa(A)\frac{\|E\|_1}{\|A\|_1},$$

where $\kappa(A) = \|A^{-1}\|_1 \|A\|_1$, the condition number of A with respect to the solution of the linear equations. See Section 4.4 of Anderson *et al.* (1999) for further details.

F07FPF (ZPOSVX) is a comprehensive LAPACK driver that returns forward and backward error bounds and an estimate of the condition number. Alternatively, F04CDF solves $Ax = b$ and returns a forward error bound and condition estimate. F04CDF calls F07FNF (ZPOSV) to solve the equations.

8 Parallelism and Performance

F07FNF (ZPOSV) is threaded by NAG for parallel execution in multithreaded implementations of the NAG Library.

F07FNF (ZPOSV) 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 floating-point operations is approximately $\frac{4}{3}n^3 + 8n^2r$, where r is the number of right-hand sides.

The real analogue of this routine is F07FAF (DPOSV).

10 Example

This example solves the equations

$$Ax = b,$$

where A is the symmetric positive definite matrix

$$A = \begin{pmatrix} 3.23 & 1.51-1.92i & 1.90+0.84i & 0.42+2.50i \\ 1.51+1.92i & 3.58 & -0.23+1.11i & -1.18+1.37i \\ 1.90-0.84i & -0.23-1.11i & 4.09 & 2.33-0.14i \\ 0.42-2.50i & -1.18-1.37i & 2.33+0.14i & 4.29 \end{pmatrix}$$

and

$$b = \begin{pmatrix} 3.93-6.14i \\ 6.17+9.42i \\ -7.17-21.83i \\ 1.99-14.38i \end{pmatrix}.$$

Details of the Cholesky factorization of A are also output.

10.1 Program Text

```
    Program f07fnfe

!     F07FNF Example Program Text

!     Mark 26 Release. NAG Copyright 2016.

!     .. Use Statements ..
      Use nag_library, Only: nag_wp, x04dbf, zposv
!     .. Implicit None Statement ..
```

```
      Implicit None
!     .. Parameters ..
      Integer, Parameter               :: nin = 5, nout = 6
!     .. Local Scalars ..
      Integer                          :: i, ifail, info, lda, n
!     .. Local Arrays ..
      Complex (Kind=nag_wp), Allocatable :: a(:,:), b(:)
      Character (1)                    :: clabs(1), rlabs(1)
!     .. Executable Statements ..
      Write (nout,*) 'F07FNF Example Program Results'
      Write (nout,*)
      Flush (nout)
!     Skip heading in data file
      Read (nin,*)
      Read (nin,*) n
      lda = n
      Allocate (a(lda,n),b(n))

!     Read the upper triangular part of A from data file

      Read (nin,*)(a(i,i:n),i=1,n)

!     Read b from data file

      Read (nin,*) b(1:n)

!     Solve the equations Ax = b for x
!     The NAG name equivalent of zposv is f07fnf
      Call zposv('Upper',n,1,a,lda,b,n,info)

      If (info==0) Then

!       Print solution

        Write (nout,*) 'Solution'
        Write (nout,99999) b(1:n)

!       Print details of factorization

        Write (nout,*)
        Flush (nout)

!       ifail: behaviour on error exit
!              =0 for hard exit, =1 for quiet-soft, =-1 for noisy-soft
        ifail = 0
        Call x04dbf('Upper','Non-unit diagonal',n,n,a,lda,'Bracketed','F7.4',  &
          'Cholesky factor U','Integer',rlabs,'Integer',clabs,80,0,ifail)

      Else
        Write (nout,99998) 'The leading minor of order ', info,                &
          ' is not positive definite'
      End If

99999 Format ((3X,4(' (',F7.4,',',F7.4,')',:)))
99998 Format (1X,A,I3,A)
    End Program f07fnfe
```

10.2 Program Data

```
F07FNF Example Program Data
   4                                                       :Value of N
 ( 3.23,  0.00) ( 1.51, -1.92) ( 1.90,  0.84) ( 0.42,  2.50)
                ( 3.58,  0.00) (-0.23,  1.11) (-1.18,  1.37)
                               ( 4.09,  0.00) ( 2.33, -0.14)
                                              ( 4.29,  0.00) :End of matrix A
 ( 3.93, -6.14) ( 6.17,  9.42) (-7.17,-21.83) ( 1.99,-14.38) :End of vector b
```

10.3 Program Results

```
F07FNF Example Program Results

Solution
   ( 1.0000,-1.0000) (-0.0000, 3.0000) (-4.0000,-5.0000) ( 2.0000, 1.0000)

Cholesky factor U
                   1                 2                 3                 4
1  ( 1.7972, 0.0000) ( 0.8402,-1.0683) ( 1.0572, 0.4674) ( 0.2337, 1.3910)
2                    ( 1.3164, 0.0000) (-0.4702,-0.3131) ( 0.0834,-0.0368)
3                                      ( 1.5604, 0.0000) ( 0.9360,-0.9900)
4                                                        ( 0.6603, 0.0000)
```