

NAG Library Routine Document

F07PNF (ZHPSV)

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

F07PNF (ZHPSV) computes the solution to a complex system of linear equations

$$AX = B,$$

where A is an n by n Hermitian matrix stored in packed format and X and B are n by r matrices.

2 Specification

```
SUBROUTINE F07PNF(UPLO, N, NRHS, AP, IPIV, B, LDB, INFO)
```

```
INTEGER N, NRHS, IPIV(N), LDB, INFO
```

```
complex*16 AP(*), B(LDB,*)
```

```
CHARACTER*1 UPLO
```

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

3 Description

F07PNF (ZHPSV) uses the diagonal pivoting method to factor A as $A = UDU^H$ if UPLO = 'U' or $A = LDL^H$ if UPLO = 'L', where U (or L) is a product of permutation and unit upper (lower) triangular matrices, D is Hermitian and block diagonal with 1 by 1 and 2 by 2 diagonal blocks. 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

Higham N J (2002) *Accuracy and Stability of Numerical Algorithms* (2nd Edition) SIAM, Philadelphia

5 Parameters

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: $\text{NRHS} \geq 0$.
- 4: AP(*) – **complex*16** array *Input/Output*
Note: the dimension of the array AP must be at least $\max(1, N \times (N + 1)/2)$.
On entry: the n by n Hermitian matrix A , packed by columns.
 More precisely,
 if UPLO = 'U', the upper triangle of A must be stored with element A_{ij} in
 AP($i + j(j - 1)/2$) for $i \leq j$;
 if UPLO = 'L', the lower triangle of A must be stored with element A_{ij} in
 AP($i + (2n - j)(j - 1)/2$) for $i \geq j$.
On exit: the block diagonal matrix D and the multipliers used to obtain the factor U or L from the
 factorization $A = UDU^H$ or $A = LDL^H$ as computed by F07PRF (ZHPTRF), stored as a packed
 triangular matrix in the same storage format as A .
- 5: IPIV(N) – INTEGER array *Output*
On exit: details of the interchanges and the block structure of D , as determined by F07PRF
 (ZHPTRF).
 IPIV(k) > 0
 Rows and columns k and IPIV(k) were interchanged, and $D(k, k)$ is a 1 by 1 diagonal block.
 UPLO = 'U' and IPIV(k) = IPIV($k - 1$) < 0
 Rows and columns $k - 1$ and $-IPIV(k)$ were interchanged and $D(k - 1 : k, k - 1 : k)$ is a 2
 by 2 diagonal block.
 UPLO = 'L' and IPIV(k) = IPIV($k + 1$) < 0
 Rows and columns $k + 1$ and $-IPIV(k)$ were interchanged and $D(k : k + 1, k : k + 1)$ is a 2
 by 2 diagonal block.
- 6: B(LDB,*) – **complex*16** array *Input/Output*
Note: the second dimension of the array B must be at least $\max(1, \text{NRHS})$.
 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 $\text{LDB} = \max(1, N)$.
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 F07PNF
 (ZHPSV) is called.
Constraint: $\text{LDB} \geq \max(1, N)$.
- 8: INFO – INTEGER *Output*
On exit: INFO = 0 unless the routine detects an error (see Section 6).

6 Error Indicators and Warnings

Errors or warnings detected by the routine:

INFO < 0

If INFO = $-i$, the i th argument had an illegal value. An explanatory message is output, and execution of the program is terminated.

INFO > 0

If INFO = i , d_{ii} is exactly zero. The factorization has been completed, but the block diagonal matrix D is exactly singular, so the solution could not be 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) and Chapter 11 of Higham (2002) for further details.

F07PPF (ZHPSVX) is a comprehensive LAPACK driver that returns forward and backward error bounds and an estimate of the condition number. Alternatively, F04CJF solves $Ax = b$ and returns a forward error bound and condition estimate. F04CJF calls F07PNF (ZHPSV) to solve the equations.

8 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 F07PAF (DSPSV).

9 Example

This example solves the equations

$$Ax = b,$$

where A is the Hermitian matrix

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

and

$$b = \begin{pmatrix} 2.98 - 10.18i \\ -9.58 + 3.88i \\ -0.77 - 16.05i \\ 7.79 + 5.48i \end{pmatrix}.$$

Details of the factorization of A are also output.

9.1 Program Text

```

*      F07PNF Example Program Text
*      Mark 21 Release. NAG Copyright 2004.
*      .. Parameters ..
      INTEGER          NIN, NOUT
      PARAMETER        (NIN=5,NOUT=6)
      INTEGER          NMAX
      PARAMETER        (NMAX=8)
      CHARACTER        UPLO
      PARAMETER        (UPLO='U')
*      .. Local Scalars ..
      INTEGER          I, IFAIL, INFO, J, N
*      .. Local Arrays ..
      COMPLEX *16      AP((NMAX*(NMAX+1))/2), B(NMAX)
      INTEGER          IPIV(NMAX)
      CHARACTER        CLABS(1), RLABS(1)
*      .. External Subroutines ..
      EXTERNAL          X04DDF, ZHPSV
*      .. Executable Statements ..
      WRITE (NOUT,*) 'F07PNF Example Program Results'
      WRITE (NOUT,*)
*      Skip heading in data file
      READ (NIN,*)
      READ (NIN,*) N
      IF (N.LE.NMAX) THEN

*
*          Read the upper or lower triangular part of the matrix A from
*          data file
*
          IF (UPLO.EQ.'U') THEN
              READ (NIN,*) ((AP(I+(J*(J-1))/2),J=I,N),I=1,N)
          ELSE IF (UPLO.EQ.'L') THEN
              READ (NIN,*) ((AP(I+((2*N-J)*(J-1))/2),J=1,I),I=1,N)
          END IF

*
*          Read b from data file
*
          READ (NIN,*) (B(I),I=1,N)

*
*          Solve the equations Ax = b for x
*
          CALL ZHPSV(UPLO,N,1,AP,IPIV,B,N,INFO)

*
          IF (INFO.EQ.0) THEN

*
*              Print solution
*
              WRITE (NOUT,*) 'Solution'
              WRITE (NOUT,99999) (B(I),I=1,N)

*
*              Print details of factorization
*
              WRITE (NOUT,*)
              IFAIL = 0
              CALL X04DDF(UPLO,'Non-unit diagonal',N,AP,'Bracketed',
+                  'F7.4','Details of factorization','Integer',
+                  RLABS,'Integer',CLABS,80,0,IFAIL)

*
*              Print pivot indices
*
              WRITE (NOUT,*)
              WRITE (NOUT,*) 'Pivot indices'
              WRITE (NOUT,99998) (IPIV(I),I=1,N)

*
          ELSE
              WRITE (NOUT,99997) 'The diagonal block ', INFO,
+                  ' of D is zero'
          END IF
          ELSE
              WRITE (NOUT,*) 'NMAX too small'
          
```

```

      END IF
*
99999 FORMAT ((3X,4(' (',F7.4,',',F7.4,')',:)))
99998 FORMAT (1X,7I11)
99997 FORMAT (1X,A,I3,A)
      END

```

9.2 Program Data

F07PNF Example Program Data

```

      4                                     :Value of N
( -1.84,  0.00) (  0.11, -0.11) ( -1.78, -1.18) (  3.91, -1.50)
              ( -4.63,  0.00) ( -1.84,  0.03) (  2.21,  0.21)
                      ( -8.87,  0.00) (  1.58, -0.90)
                                ( -1.36,  0.00) :End matrix A
(  2.98,-10.18) ( -9.58,  3.88) ( -0.77,-16.05) (  7.79,  5.48) :End vector b

```

9.3 Program Results

F07PNF Example Program Results

Solution

```

( 2.0000, 1.0000) ( 3.0000,-2.0000) (-1.0000, 2.0000) ( 1.0000,-1.0000)

```

Details of factorization

```

      1          2          3          4
1 (-7.1028, 0.0000) ( 0.2997, 0.1578) ( 0.3397, 0.0303) (-0.1518, 0.3743)
2              (-5.4176, 0.0000) ( 0.5637, 0.2850) ( 0.3100, 0.0433)
3                      (-1.8400, 0.0000) ( 3.9100,-1.5000)
4                                  (-1.3600, 0.0000)

```

Pivot indices

```

      1          2          -1          -1

```
