

NAG Fortran Library Routine Document

F07PAF (DSPSV)

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

F07PAF (DSPSV) computes the solution to a real system of linear equations

$$AX = B,$$

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

2 Specification

SUBROUTINE F07PAF (UPLO, N, NRHS, AP, IPIV, B, LDB, INFO)

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

double precision AP(*), B(LDB,*)

CHARACTER*1 UPLO

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

3 Description

The diagonal pivoting method is used to factor A as $A = UDU^T$, if UPLO = 'U' or $A = LDL^T$, if UPLO = 'L', where U (or L) is a product of permutation and unit upper (lower) triangular matrices, D is symmetric 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 URL: <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(*) – **double precision** array *Input/Output*
Note: the dimension of the array AP must be at least $\max(N \times (N + 1)/2)$.
On entry: the upper or lower triangle of the symmetric matrix A , packed columnwise in a linear array. The j th column of A is stored in the array AP as follows:
 if UPLO = 'U', $\text{AP}(i + (j - 1) \times j/2) = a_{ij}$ for $1 \leq i \leq j$;
 if UPLO = 'L', $\text{AP}(i + (j - 1) \times (2n - j)/2) = a_{ij}$ for $j \leq i \leq n$.
On exit: the block diagonal matrix D and the multipliers used to obtain the factor U or L from the factorization $A = UDU^T$ or $A = LDL^T$ as computed by F07PDF (DSPTRF), stored as a packed triangular matrix in the same storage format as A .
- 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 , as determined by F07PDF (DSPTRF). If $\text{IPIV}(k) > 0$, then rows and columns k and $\text{IPIV}(k)$ were interchanged, and $D(k, k)$ is a 1 by 1 diagonal block. If UPLO = 'U' and $\text{IPIV}(k) = \text{IPIV}(k - 1) < 0$, then rows and columns $k - 1$ and $-\text{IPIV}(k)$ were interchanged and $D(k - 1 : k, k - 1 : k)$ is a 2 by 2 diagonal block. If UPLO = 'L' and $\text{IPIV}(k) = \text{IPIV}(k + 1) < 0$, then rows and columns $k + 1$ and $-\text{IPIV}(k)$ were interchanged and $D(k : k + 1, k : k + 1)$ is a 2 by 2 diagonal block.
- 6: B(LDB,*) – **double precision** array *Input/Output*
Note: the second dimension of the array B must be at least $\max(1, \text{NRHS})$.
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 F07PAF (DSPSV) 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.

F07PBF (DSPSVX) is a comprehensive LAPACK driver that returns forward and backward error bounds and an estimate of the condition number. Alternatively, F04BJF solves $Ax = b$ and returns a forward error bound and condition estimate. F04BJF calls F07PAF (DSPSV) to solve the equations.

8 Further Comments

The total number of floating point operations is approximately $\frac{1}{3}n^3 + 2n^2r$, where r is the number of right-hand sides.

The complex analogues of F07PAF (DSPSV) are F07PNF (ZHPSV) for Hermitian matrices, and F07QNF (ZSPSV) for symmetric matrices.

9 Example

To solve the equations

$$Ax = b,$$

where A is the symmetric matrix

$$A = \begin{pmatrix} -1.81 & 2.06 & 0.63 & -1.15 \\ 2.06 & 1.15 & 1.87 & 4.20 \\ 0.63 & 1.87 & -0.21 & 3.87 \\ -1.15 & 4.20 & 3.87 & 2.07 \end{pmatrix}$$

and

$$b = \begin{pmatrix} 0.96 \\ 6.07 \\ 8.38 \\ 9.50 \end{pmatrix}.$$

Details of the factorization of A are also output.

9.1 Program Text

Note: the listing of the example program presented below uses *bold italicised* terms to denote precision-dependent details. Please read the Users' Note for your implementation to check the interpretation of these terms. As explained in the Essential Introduction to this manual, the results produced may not be identical for all implementations.

```
*      F07PAF 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 ..
DOUBLE PRECISION AP((NMAX*(NMAX+1))/2), B(NMAX)
INTEGER          IPIV(NMAX)
*   .. External Subroutines ..
EXTERNAL        DSPSV, X04CCF
*   .. Executable Statements ..
WRITE (NOUT,*) 'F07PAF 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 DSPSV(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 X04CCF(UPLO,'Non-unit diagonal',N,AP,
+             'Details of the factorization',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
  STOP
*
99999 FORMAT ((3X,7F11.4))
99998 FORMAT (1X,7I11)
99997 FORMAT (1X,A,I3,A)
END

```

9.2 Program Data

```

F07PAF Example Program Data
4                               :Value of N
-1.81   2.06   0.63  -1.15
         1.15   1.87   4.20
                -0.21  3.87
                        2.07 :End of matrix A
0.96   6.07   8.38   9.50 :End of vector b

```

9.3 Program Results

F07PAF Example Program Results

```

Solution
-5.0000   -2.0000   1.0000   4.0000

```

Details of the factorization

	1	2	3	4
1	0.4074	0.3031	-0.5960	0.6537
2		-2.5907	0.8115	0.2230
3			1.1500	4.2000
4				2.0700

Pivot indices

1	2	-2	-2
---	---	----	----
