NAG Library Routine Document

F07PBF (DSPSVX)

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

F07PBF (DSPSVX) uses the diagonal pivoting factorization

$$A = UDU^{\mathrm{T}} \quad \text{or} \quad A = LDL^{\mathrm{T}}$$

to compute 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. Error bounds on the solution and a condition estimate are also provided.

2 Specification

```
SUBROUTINE F07PBF (FACT, UPLO, N, NRHS, AP, AFP, IPIV, B, LDB, X, LDX,          &
                   RCOND, FERR, BERR, WORK, IWORK, INFO)

INTEGER            N, NRHS, IPIV(N), LDB, LDX, IWORK(N), INFO
REAL (KIND=nag_wp) AP(*), AFP(*), B(LDB,*), X(LDX,*), RCOND, FERR(NRHS),        &
                   BERR(NRHS), WORK(3*N)
CHARACTER(1)       FACT, UPLO
```

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

3 Description

F07PBF (DSPSVX) performs the following steps:

1. If FACT = 'N', the diagonal pivoting method is used to factor A as $A = UDU^{\mathrm{T}}$ if UPLO = 'U' or $A = LDL^{\mathrm{T}}$ if UPLO = 'L', where U (or L) is a product of permutation and unit upper (lower) triangular matrices and D is symmetric and block diagonal with 1 by 1 and 2 by 2 diagonal blocks.
2. If some $d_{ii} = 0$, so that D is exactly singular, then the routine returns with INFO $= i$. Otherwise, the factored form of A is used to estimate the condition number of the matrix A. If the reciprocal of the condition number is less than ***machine precision***, INFO $=$ N $+ 1$ is returned as a warning, but the routine still goes on to solve for X and compute error bounds as described below.
3. The system of equations is solved for X using the factored form of A.
4. Iterative refinement is applied to improve the computed solution matrix and to calculate error bounds and backward error estimates for it.

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: FACT – CHARACTER(1) *Input*

On entry: specifies whether or not the factorized form of the matrix A has been supplied.

FACT = 'F'
: AFP and IPIV contain the factorized form of the matrix A. AFP and IPIV will not be modified.

FACT = 'N'
: The matrix A will be copied to AFP and factorized.

Constraint: FACT = 'F' or 'N'.

2: 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'.

3: N – INTEGER *Input*

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

Constraint: $\mathrm{N} \geq 0$.

4: NRHS – INTEGER *Input*

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

Constraint: $\mathrm{NRHS} \geq 0$.

5: AP($*$) – REAL (KIND=nag_wp) array *Input*

Note: the dimension of the array AP must be at least $\max(1, \mathrm{N} \times (\mathrm{N}+1)/2)$.

On entry: the n by n symmetric matrix A, packed by columns.

More precisely,

if UPLO = 'U', the upper triangle of A must be stored with element A_{ij} in $\mathrm{AP}(i + j(j-1)/2)$ for $i \leq j$;

if UPLO = 'L', the lower triangle of A must be stored with element A_{ij} in $\mathrm{AP}(i + (2n-j)(j-1)/2)$ for $i \geq j$.

6: AFP($*$) – REAL (KIND=nag_wp) array *Input/Output*

Note: the dimension of the array AFP must be at least $\max(1, \mathrm{N} \times (\mathrm{N}+1)/2)$.

On entry: if FACT = 'F', AFP contains the block diagonal matrix D and the multipliers used to obtain the factor U or L from the factorization $A = UDU^{\mathrm{T}}$ or $A = LDL^{\mathrm{T}}$ as computed by F07PDF (DSPTRF), stored as a packed triangular matrix in the same storage format as A.

On exit: if FACT = 'N', AFP contains the block diagonal matrix D and the multipliers used to obtain the factor U or L from the factorization $A = UDU^{\mathrm{T}}$ or $A = LDL^{\mathrm{T}}$ as computed by F07PDF (DSPTRF), stored as a packed triangular matrix in the same storage format as A.

7: IPIV(N) – INTEGER array *Input/Output*

On entry: if FACT = 'F', IPIV contains details of the interchanges and the block structure of D, as determined by F07PDF (DSPTRF).

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

On exit: if FACT = 'N', IPIV contains details of the interchanges and the block structure of D, as determined by F07PDF (DSPTRF), as described above.

8: B(LDB,$*$) – REAL (KIND=nag_wp) array *Input*

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

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

9: LDB – INTEGER *Input*

On entry: the first dimension of the array B as declared in the (sub)program from which F07PBF (DSPSVX) is called.

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

10: X(LDX,$*$) – REAL (KIND=nag_wp) array *Output*

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

On exit: if $\mathrm{INFO} = 0$ or $\mathrm{N} + 1$, the n by r solution matrix X.

11: LDX – INTEGER *Input*

On entry: the first dimension of the array X as declared in the (sub)program from which F07PBF (DSPSVX) is called.

Constraint: $\mathrm{LDX} \geq \max(1, \mathrm{N})$.

12: RCOND – REAL (KIND=nag_wp) *Output*

On exit: the estimate of the reciprocal condition number of the matrix A. If $\mathrm{RCOND} = 0.0$, the matrix may be exactly singular. This condition is indicated by $\mathrm{INFO} > 0$ and $\mathrm{INFO} \leq \mathrm{N}$. Otherwise, if RCOND is less than the ***machine precision***, the matrix is singular to working precision. This condition is indicated by $\mathrm{INFO} = \mathrm{N} + 1$.

13: FERR(NRHS) – REAL (KIND=nag_wp) array *Output*

On exit: if $\mathrm{INFO} = 0$ or $\mathrm{N} + 1$, an estimate of the forward error bound for each computed solution vector, such that $\|\hat{x}_j - x_j\|_\infty / \|x_j\|_\infty \leq \mathrm{FERR}(j)$ where $\hat{x}_j$ is the jth column of the computed solution returned in the array X and x_j is the corresponding column of the exact solution X. The estimate is as reliable as the estimate for RCOND, and is almost always a slight overestimate of the true error.

14: BERR(NRHS) – REAL (KIND=nag_wp) array *Output*

On exit: if INFO $= 0$ or $\mathrm{N} + 1$, an estimate of the component-wise relative backward error of each computed solution vector $\hat{x}_j$ (i.e., the smallest relative change in any element of A or B that makes $\hat{x}_j$ an exact solution).

15: WORK($3 \times \mathrm{N}$) – REAL (KIND=nag_wp) array *Workspace*

16: IWORK(N) – INTEGER array *Workspace*

17: 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 ith argument had an illegal value. An explanatory message is output, and execution of the program is terminated.

INFO > 0 and INFO $\leq$ N

If INFO $\leq$ N, $d(i,i)$ is exactly zero. The factorization has been completed, but the factor D is exactly singular, so the solution and error bounds could not be computed. RCOND $= 0.0$ is returned.

INFO $= \mathrm{N} + 1$

D is nonsingular, but RCOND is less than ***machine precision***, meaning that the matrix is singular to working precision. Nevertheless, the solution and error bounds are computed because there are a number of situations where the computed solution can be more accurate than the value of RCOND would suggest.

7 Accuracy

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

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

where ϵ is the ***machine precision***. See Chapter 11 of Higham (2002) for further details.

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

$$\frac{\|x-\hat{x}\|_\infty}{\|\hat{x}\|_\infty} \leq w_c \operatorname{cond}(A,\hat{x},b)$$

where $\operatorname{cond}(A,\hat{x},b) = \left\| |A^{-1}|(|A||\hat{x}|+|b|) \right\|_\infty / \|\hat{x}\|_\infty \leq \operatorname{cond}(A) = \left\| |A^{-1}||A| \right\|_\infty \leq \kappa_\infty(A)$. If $\hat{x}$ is the jth column of X, then w_c is returned in BERR(j) and a bound on $\|x-\hat{x}\|_\infty / \|\hat{x}\|_\infty$ is returned in FERR(j). See Section 4.4 of Anderson *et al.* (1999) for further details.

8 Further Comments

The factorization of A requires approximately $\frac{1}{3}n^3$ floating point operations.

For each right-hand side, computation of the backward error involves a minimum of $4n^2$ floating point operations. Each step of iterative refinement involves an additional $6n^2$ operations. At most five steps of iterative refinement are performed, but usually only one or two steps are required. Estimating the forward error involves solving a number of systems of equations of the form $Ax = b$; the number is usually 4 or 5 and never more than 11. Each solution involves approximately $2n^2$ operations.

The complex analogues of this routine are F07PPF (ZHPSVX) for Hermitian matrices, and F07QPF (ZSPSVX) for symmetric matrices.

9 Example

This example solves the equations

$$AX = B,$$

where A is the symmetric matrix

$$A = \begin{pmatrix} -1.81 & 2.06 & 0.63 & -1.15 \\ 2.06 & 1.15 & 1.87 & 4.20 \\ 0.63 & 1.87 & -0.21 & 3.87 \\ -1.15 & 4.20 & 3.87 & 2.07 \end{pmatrix} \quad \text{and} \quad B = \begin{pmatrix} 0.96 & 3.93 \\ 6.07 & 19.25 \\ 8.38 & 9.90 \\ 9.50 & 27.85 \end{pmatrix}.$$

Error estimates for the solutions, and an estimate of the reciprocal of the condition number of the matrix A are also output.

9.1 Program Text

```
    Program f07pbfe

!     F07PBF Example Program Text

!     Mark 24 Release. NAG Copyright 2012.

!     .. Use Statements ..
      Use nag_library, Only: dspsvx, nag_wp, x04caf
!     .. Implicit None Statement ..
      Implicit None
!     .. Parameters ..
      Integer, Parameter               :: nin = 5, nout = 6
      Character (1), Parameter         :: uplo = 'U'
!     .. Local Scalars ..
      Real (Kind=nag_wp)               :: rcond
      Integer                          :: i, ifail, info, j, ldb, ldx, n, nrhs
!     .. Local Arrays ..
      Real (Kind=nag_wp), Allocatable  :: afp(:), ap(:), b(:,:), berr(:),      &
                                          ferr(:), work(:), x(:,:)
      Integer, Allocatable             :: ipiv(:), iwork(:)
!     .. Executable Statements ..
      Write (nout,*) 'F07PBF Example Program Results'
      Write (nout,*)
      Flush (nout)
!     Skip heading in data file
      Read (nin,*)
      Read (nin,*) n, nrhs
      ldb = n
      ldx = n
      Allocate (afp((n*(n+1))/2),ap((n*(n+1))/2),b(ldb,nrhs),berr(nrhs),ferr( &
        nrhs),work(3*n),x(ldx,nrhs),ipiv(n),iwork(n))

!     Read the upper or lower triangular part of the matrix A from
!     data file

      If (uplo=='U') Then
        Read (nin,*)((ap(i+(j*(j-1))/2),j=i,n),i=1,n)
      Else If (uplo=='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,1:nrhs),i=1,n)
```

```
!     Solve the equations AX = B for X
!     The NAG name equivalent of dspsvx is f07pbf
      Call dspsvx('Not factored',uplo,n,nrhs,ap,afp,ipiv,b,ldb,x,ldx,rcond, &
        ferr,berr,work,iwork,info)

      If ((info==0) .Or. (info==n+1)) Then

!       Print solution, error bounds and condition number

!       ifail: behaviour on error exit
!              =0 for hard exit, =1 for quiet-soft, =-1 for noisy-soft
        ifail = 0
        Call x04caf('General',' ',n,nrhs,x,ldx,'Solution(s)',ifail)

        Write (nout,*)
        Write (nout,*) 'Backward errors (machine-dependent)'
        Write (nout,99999) berr(1:nrhs)
        Write (nout,*)
        Write (nout,*) 'Estimated forward error bounds (machine-dependent)'
        Write (nout,99999) ferr(1:nrhs)
        Write (nout,*)
        Write (nout,*) 'Estimate of reciprocal condition number'
        Write (nout,99999) rcond
        Write (nout,*)

        If (info==n+1) Then
          Write (nout,*)
          Write (nout,*) 'The matrix A is singular to working precision'
        End If
      Else
        Write (nout,99998) 'The diagonal block ', info, ' of D is zero'
      End If

99999 Format ((3X,1P,7E11.1))
99998 Format (1X,A,I3,A)
    End Program f07pbfe
```

9.2 Program Data

```
F07PBF Example Program Data
  4      2                      :Values of N and NRHS
 -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   3.93
  6.07  19.25
  8.38   9.90
  9.50  27.85                   :End of matrix B
```

9.3 Program Results

```
 F07PBF Example Program Results

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

 Backward errors (machine-dependent)
       1.4E-16    1.0E-16
```

```
 Estimated forward error bounds (machine-dependent)
       2.5E-14    3.2E-14

 Estimate of reciprocal condition number
       1.3E-02
```