

NAG Library Routine Document

F07HBF (DPBSVX)

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

F07HBF (DPBSVX) uses the Cholesky factorization

$$A = U^T U \quad \text{or} \quad A = LL^T$$

to compute the solution to a real system of linear equations

$$AX = B,$$

where A is an n by n symmetric positive definite band matrix of bandwidth $(2k_d + 1)$ and X and B are n by r matrices. Error bounds on the solution and a condition estimate are also provided.

2 Specification

```
SUBROUTINE F07HBF (FACT, UPLO, N, KD, NRHS, AB, LDAB, AFB, LDAFB, EQUED, S,      &
                  B, LDB, X, LDX, RCOND, FERR, BERR, WORK, IWORK, INFO)
INTEGER          N, KD, NRHS, LDAB, LDAFB, LDB, LDX, IWORK(N), INFO
REAL (KIND=nag_wp) AB(LDAB,*), AFB(LDAFB,*), S(*), B(LDB,*), X(LDX,*),      &
                  RCOND, FERR(NRHS), BERR(NRHS), WORK(3*N)
CHARACTER(1)     FACT, UPLO, EQUED
```

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

3 Description

F07HBF (DPBSVX) performs the following steps:

1. If FACT = 'E', real diagonal scaling factors, D_S , are computed to equilibrate the system:

$$(D_S A D_S)(D_S^{-1} X) = D_S B.$$

Whether or not the system will be equilibrated depends on the scaling of the matrix A , but if equilibration is used, A is overwritten by $D_S A D_S$ and B by $D_S B$.

2. If FACT = 'N' or 'E', the Cholesky decomposition is used to factor the matrix A (after equilibration if FACT = 'E') as $A = U^T U$ if UPLO = 'U' or $A = LL^T$ if UPLO = 'L', where U is an upper triangular matrix and L is a lower triangular matrix.
3. If the leading i by i principal minor of A is not positive definite, 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.
4. The system of equations is solved for X using the factored form of A .
5. Iterative refinement is applied to improve the computed solution matrix and to calculate error bounds and backward error estimates for it.
6. If equilibration was used, the matrix X is premultiplied by D_S so that it solves the original system before equilibration.

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 is supplied on entry, and if not, whether the matrix A should be equilibrated before it is factorized.
 FACT = 'F'
 AFB contains the factorized form of A . If EQUED = 'Y', the matrix A has been equilibrated with scaling factors given by S . AB and AFB will not be modified.
 FACT = 'N'
 The matrix A will be copied to AFB and factorized.
 FACT = 'E'
 The matrix A will be equilibrated if necessary, then copied to AFB and factorized.
Constraint: FACT = 'F', 'N' or 'E'.
- 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: $N \geq 0$.
- 4: KD – INTEGER *Input*
On entry: k_d , the number of superdiagonals of the matrix A if UPLO = 'U', or the number of subdiagonals if UPLO = 'L'.
Constraint: $KD \geq 0$.
- 5: NRHS – INTEGER *Input*
On entry: r , the number of right-hand sides, i.e., the number of columns of the matrix B .
Constraint: $NRHS \geq 0$.
- 6: AB(LDAB,*) – REAL (KIND=nag_wp) array *Input/Output*
Note: the second dimension of the array AB must be at least $\max(1, N)$.
On entry: the upper or lower triangle of the symmetric band matrix A , except if FACT = 'F' and EQUED = 'Y', in which case AB must contain the equilibrated matrix $D_S A D_S$.

The matrix is stored in rows 1 to $k_d + 1$, more precisely,

if UPLO = 'U', the elements of the upper triangle of A within the band must be stored with element A_{ij} in $AB(k_d + 1 + i - j, j)$ for $\max(1, j - k_d) \leq i \leq j$;

if UPLO = 'L', the elements of the lower triangle of A within the band must be stored with element A_{ij} in $AB(1 + i - j, j)$ for $j \leq i \leq \min(n, j + k_d)$.

On exit: if FACT = 'E' and EQUED = 'Y', AB is overwritten by $D_S A D_S$.

7: LDAB – INTEGER *Input*

On entry: the first dimension of the array AB as declared in the (sub)program from which F07HBF (DPBSVX) is called.

Constraint: LDAB \geq KD + 1.

8: AFB(LDAFB,*) – REAL (KIND=nag_wp) array *Input/Output*

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

On entry: if FACT = 'F', AFB contains the triangular factor U or L from the Cholesky factorization $A = U^T U$ or $A = LL^T$ of the band matrix A , in the same storage format as A . If EQUED = 'Y', AFB is the factorized form of the equilibrated matrix A .

On exit: if FACT = 'N', AFB returns the triangular factor U or L from the Cholesky factorization $A = U^T U$ or $A = LL^T$.

If FACT = 'E', AFB returns the triangular factor U or L from the Cholesky factorization $A = U^T U$ or $A = LL^T$ of the equilibrated matrix A (see the description of AB for the form of the equilibrated matrix).

9: LDAFB – INTEGER *Input*

On entry: the first dimension of the array AFB as declared in the (sub)program from which F07HBF (DPBSVX) is called.

Constraint: LDAFB \geq KD + 1.

10: EQUED – CHARACTER(1) *Input/Output*

On entry: if FACT = 'N' or 'E', EQUED need not be set.

If FACT = 'F', EQUED must specify the form of the equilibration that was performed as follows:

if EQUED = 'N', no equilibration;

if EQUED = 'Y', equilibration was performed, i.e., A has been replaced by $D_S A D_S$.

On exit: if FACT = 'F', EQUED is unchanged from entry.

Otherwise, if no constraints are violated, EQUED specifies the form of the equilibration that was performed as specified above.

Constraint: if FACT = 'F', EQUED = 'N' or 'Y'.

11: S(*) – REAL (KIND=nag_wp) array *Input/Output*

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

On entry: if FACT = 'N' or 'E', S need not be set.

If FACT = 'F' and EQUED = 'Y', S must contain the scale factors, D_S , for A ; each element of S must be positive.

On exit: if FACT = 'F', S is unchanged from entry.

Otherwise, if no constraints are violated and EQUED = 'Y', S contains the scale factors, D_S , for A ; each element of S is positive.

- 12: B(LDB,*) – REAL (KIND=nag_wp) 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 EQUED = 'N', B is not modified.
 If EQUED = 'Y', B is overwritten by $D_S B$.
- 13: LDB – INTEGER Input
On entry: the first dimension of the array B as declared in the (sub)program from which F07HBF (DPBSVX) is called.
Constraint: $\text{LDB} \geq \max(1, N)$.
- 14: X(LDX,*) – REAL (KIND=nag_wp) array Output
Note: the second dimension of the array X must be at least $\max(1, \text{NRHS})$.
On exit: if INFO = 0 or $N + 1$, the n by r solution matrix X to the original system of equations. Note that the arrays A and B are modified on exit if EQUED = 'Y', and the solution to the equilibrated system is $D_S^{-1} X$.
- 15: LDX – INTEGER Input
On entry: the first dimension of the array X as declared in the (sub)program from which F07HBF (DPBSVX) is called.
Constraint: $\text{LDX} \geq \max(1, N)$.
- 16: RCOND – REAL (KIND=nag_wp) Output
On exit: if no constraints are violated, an estimate of the reciprocal condition number of the matrix A (after equilibration if that is performed), computed as $\text{RCOND} = 1.0 / (\|A\|_1 \|A^{-1}\|_1)$.
- 17: FERR(NRHS) – REAL (KIND=nag_wp) array Output
On exit: if INFO = 0 or $N + 1$, an estimate of the forward error bound for each computed solution vector, such that $\|\hat{x}_j - x_j\|_\infty / \|x_j\|_\infty \leq \text{FERR}(j)$ where \hat{x}_j is the j th 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.
- 18: BERR(NRHS) – REAL (KIND=nag_wp) array Output
On exit: if INFO = 0 or $N + 1$, 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).
- 19: WORK(3 × N) – REAL (KIND=nag_wp) array Workspace
- 20: IWORK(N) – INTEGER array Workspace
- 21: 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 and INFO ≤ N

If INFO = i and $i \leq N$, the leading minor of order i of A is not positive definite, so the factorization could not be completed, and the solution has not been computed. RCOND = 0.0 is returned.

INFO = N + 1

The triangular matrix U (or L) 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 x is the exact solution of a perturbed system of equations $(A + E)x = b$, where

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

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

$c(n)$ is a modest linear function of n , and ϵ is the *machine precision*. See Section 10.1 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

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

where $\text{cond}(A, \hat{x}, b) = \frac{\| |A^{-1}|(|A||\hat{x}| + |b|) \|_\infty}{\|\hat{x}\|_\infty} \leq \text{cond}(A) = \| |A^{-1}| |A| \|_\infty \leq \kappa_\infty(A)$. If \hat{x} is the j th 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

When $n \gg k$, the factorization of A requires approximately $n(k + 1)^2$ floating point operations, where k is the number of superdiagonals.

For each right-hand side, computation of the backward error involves a minimum of $8nk$ floating point operations. Each step of iterative refinement involves an additional $12nk$ 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 $4nk$ operations.

The complex analogue of this routine is F07HPF (ZPBSVX).

9 Example

This example solves the equations

$$AX = B,$$

where A is the symmetric positive definite band matrix

$$A = \begin{pmatrix} 5.49 & 2.68 & 0 & 0 \\ 2.68 & 5.63 & -2.39 & 0 \\ 0 & -2.39 & 2.60 & -2.22 \\ 0 & 0 & -2.22 & 5.17 \end{pmatrix}$$

and

$$B = \begin{pmatrix} 22.09 & 5.10 \\ 9.31 & 30.81 \\ -5.24 & -25.82 \\ 11.83 & 22.90 \end{pmatrix}.$$

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

9.1 Program Text

Program f07hbfe

```

!      F07HBF Example Program Text
!
!      Mark 24 Release. NAG Copyright 2012.
!
!      .. Use Statements ..
!      Use nag_library, Only: dpbsvx, 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, kd, ldab, ldafb, &
!                                ldb, ldx, n, nrhs
!      Character (1)              :: equed
!      .. Local Arrays ..
!      Real (Kind=nag_wp), Allocatable :: ab(:,,:), afb(:,,:), b(:,,:), berr(:), &
!                                ferr(:), s(:), work(:), x(:,,:)
!      Integer, Allocatable        :: iwork(:)
!      .. Intrinsic Procedures ..
!      Intrinsic                   :: max, min
!      .. Executable Statements ..
!      Write (nout,*) 'F07HBF Example Program Results'
!      Write (nout,*)
!      Flush (nout)
!      Skip heading in data file
!      Read (nin,*)
!      Read (nin,*) n, kd, nrhs
!      ldb = n
!      ldx = n
!      ldab = kd + 1
!      ldafb = kd + 1
!      Allocate (ab(ldab,n),afb(ldafb,n),b(ldb,nrhs),berr(nrhs),ferr(nrhs), &
!              s(n),work(3*n),x(ldx,nrhs),iwork(n))
!
!      Read the upper or lower triangular part of the band matrix A
!      from data file
!
!      If (uplo=='U') Then
!          Read (nin,*)((ab(kd+1+i-j,j),j=i,min(n,i+kd)),i=1,n)
!      Else If (uplo=='L') Then
!          Read (nin,*)((ab(1+i-j,j),j=max(1,i-kd),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 dpbsvx is f07hbf
!       Call dpbsvx('Equilibration',uplo,n,kd,nrhs,ab,ldab,afb,ldafb,equed,s,b, &
!           ldb,x,ldx,rcond,ferr,berr,work,iwork,info)

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

!           Print solution, error bounds, condition number and the form
!           of equilibration

!           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 (equed=='N') Then
!               Write (nout,*) 'A has not been equilibrated'
!           Else If (equed=='Y') Then
!               Write (nout,*) &
!                   'A has been row and column scaled as diag(S)*A*diag(S)'
!           End If

!           If (info==n+1) Then
!               Write (nout,*)
!               Write (nout,*) 'The matrix A is singular to working precision'
!           End If
!       Else
!           Write (nout,99998) 'The leading minor of order ', info, &
!               ' is not positive definite'
!       End If

99999 Format ((3X,1P,7E11.1))
99998 Format (1X,A,I3,A)
!       End Program f07hbfe

```

9.2 Program Data

F07HBF Example Program Data

```

4           1           2           :Values of N, KD and NRHS

5.49       2.68
           5.63       -2.39
                   2.60       -2.22
                           5.17 :End of matrix A

22.09      5.10
           9.31      30.81
          -5.24     -25.82
          11.83     22.90           :End of matrix B

```

9.3 Program Results

F07HBF Example Program Results

```

Solution(s)
           1           2
1           5.0000      -2.0000
2          -2.0000       6.0000

```

3	-3.0000	-1.0000
4	1.0000	4.0000

Backward errors (machine-dependent)
1.1E-16 1.1E-16

Estimated forward error bounds (machine-dependent)
2.1E-14 3.0E-14

Estimate of reciprocal condition number
1.3E-02

A has not been equilibrated
