NAG Fortran Library Routine Document

F07BPF (ZGBSVX)

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

F07BPF (ZGBSVX) uses the LU factorization to compute the solution to a complex system of linear equations

$$AX = B, \quad A^T X = B \quad \text{or} \quad A^H X = B,$$

where A is an n by n band matrix, with k_l sub-diagonals and k_u super-diagonals, and X and B are n by r matrices. Error bounds on the solution and a condition estimate are also provided.

2 Specification

```
 SUBROUTINE F07BPF (FACT, TRANS, N, KL, KU, NRHS, AB, LDAB, AFB, LDAFB,
1                   IPIV, EQUED, R, C, B, LDB, X, LDX, RCOND, FERR, BERR,
2                   WORK, RWORK, INFO)
 INTEGER            N, KL, KU, NRHS, LDAB, LDAFB, IPIV(*), LDB, LDX, INFO
 double precision   R(*), C(*), RCOND, FERR(*), BERR(*), RWORK(*)
 complex*16         AB(LDAB,*), AFB(LDAFB,*), B(LDB,*), X(LDX,*), WORK(*)
 CHARACTER*1        FACT, TRANS, EQUED
```

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

3 Description

The following steps are performed:

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

   if TRANS = 'N', $(D_R A D_C)(D_C^{-1}) = D_R B$;
   if TRANS = 'T', $(D_R A D_C)^T (D_R^{-1} X) = D_C B$;
   if TRANS = 'C', $(D_R A D_C)^H (D_R^{-1} X) = D_C B$;

   where D_R and D_C are diagonal matrices with positive diagonal elements.

   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 $D_R A D_C$ and B by $D_R B$ (if TRANS = 'N') or $D_C B$ (if TRANS = 'T' or 'C').

2. If FACT = 'N' or 'E', the LU decomposition is used to factor the matrix A (after equilibration if FACT = 'E') as

   $$A = PLU,$$

   where P is a permutation matrix, L is a unit lower triangular matrix, and U is upper triangular.

3. If some $u_{ii} = 0$, so that U is exactly singular, then the routine returns with INFO $= i$. Otherwise, the factorized 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 $= \mathrm{N} + 1$ 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 factorized form of A.

5. Iterative refinement is applied to improve the computed solution matrix and to calculate error bounds and backward error estimates for the computed solution.
6. If equilibration was used, the matrix X is premultiplied by D_C (if TRANS = 'N') or D_R (if TRANS = 'T' or 'C') 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 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: FACT – CHARACTER*1 *Input*

On entry: specifies whether or not the factored form of the matrix A is supplied on entry, and if not, whether the matrix A should be equilibrated before it is factorized as follows:

if FACT = 'F', on entry, AFB and IPIV contain the factored form of A. If EQUED $\neq$ 'N', the matrix A has been equilibrated with scaling factors given by R and C. AB, AFB and IPIV are not modified;
if FACT = 'N', the matrix A will be copied to AFB and factorized;
if FACT = 'E', the matrix A will be equilibrated if necessary, then copied to AFB and factorized.

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

2: TRANS – CHARACTER*1 *Input*

On entry: specifies the form of the system of equations as follows:

if TRANS = 'N', $AX = B$ (No transpose);
if TRANS = 'T', $A^TX = B$ (Transpose);
if TRANS = 'C', $A^HX = B$ (Conjugate transpose).

Constraint: TRANS = 'N', 'T' or 'C'.

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: KL – INTEGER *Input*

On entry: k_l, the number of sub-diagonals within the band of the matrix A.

Constraint: KL $\geq$ 0.

5: KU – INTEGER *Input*

On entry: k_u, the number of super-diagonals within the band of the matrix A.

Constraint: KU $\geq$ 0.

6: 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.

7: AB(LDAB,$*$) – ***complex*16*** array *Input/Output*

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

On entry: the n by n coefficient matrix A in band storage, in rows 1 to $\mathrm{KL} + \mathrm{KU} + 1$. The jth column of A is stored in the jth column of the array AB as follows:

$$\mathrm{AB}(k_u + 1 + i - j, j) = a_{ij} \quad \text{for } \max(1, j - k_u) \leq i \leq \min(m, j + k_l).$$

See Section 8 for further details. If FACT = 'F' and EQUED $\neq$ 'N', A must have been equilibrated by the scaling factors in R and/or C.

On exit: AB is not modified if FACT = 'F' or 'N', or if FACT = 'E' and EQUED = 'N'.

If EQUED $\neq$ 'N' then, if INFO ≥ 0, A is scaled as follows:

if EQUED = 'R', $A = D_r A$;
if EQUED = 'C', $A = A D_c$;
if EQUED = 'B', $A = D_r A D_c$.

8: LDAB – INTEGER *Input*

On entry: the first dimension of the array AB as declared in the (sub)program from which F07BPF (ZGBSVX) is called.

Constraint: $\mathrm{LDAB} \geq \mathrm{KL} + \mathrm{KU} + 1$.

9: AFB(LDAFB,$*$) – ***complex*16*** array *Input/Output*

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

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

If FACT = 'F', details of the LU factorization of the n by n band matrix A, as computed by F07BDF (DGBTRF). U is stored as an upper triangular band matrix with $\mathrm{KL} + \mathrm{KU}$ super-diagonals in rows 1 to $\mathrm{KL} + \mathrm{KU} + 1$, and the multipliers used during the factorization are stored in rows $\mathrm{KL} + \mathrm{KU} + 2$ to $2 \times \mathrm{KL} + \mathrm{KU} + 1$. If EQUED $\neq$ 'N', AFB is the factored form of the equilibrated matrix A.

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

Otherwise, if INFO ≥ 0, then if FACT = 'N', AFB returns details of the LU factorization of the band matrix A, and if FACT = 'E', AFB returns details of the LU factorization of the equilibrated band matrix A (see the description of AB for the form of the equilibrated matrix).

10: LDAFB – INTEGER *Input*

On entry: the first dimension of the array AFB as declared in the (sub)program from which F07BPF (ZGBSVX) is called.

Constraint: $\mathrm{LDAFB} \geq 2 \times \mathrm{KL} + \mathrm{KU} + 1$.

11: IPIV($*$) – INTEGER array *Input/Output*

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

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

If FACT = 'F', IPIV contains the pivot indices from the factorization $A = LU$, as computed by F07BDF (DGBTRF); row i of the matrix was interchanged with row IPIV(i).

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

Otherwise, if INFO ≥ 0, IPIV contains the pivot indices that define the permutation matrix P; at the ith step row i of the matrix was interchanged with row IPIV(i). IPIV(i) $= i$ indicates a row interchange was not required.

If FACT = 'N', the pivot indices are those corresponding to the factorization $A = LU$ of the original matrix A.

If FACT = 'E', the pivot indices are those corresponding to the factorization of $A = LU$ of the equilibrated matrix A.

12: 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 = 'R', row equilibration, i.e., A has been premultiplied by D_R;
if EQUED = 'C', column equilibration, i.e., A has been postmultiplied by D_C;
if EQUED = 'B', both row and column equilibration, i.e., A has been replaced by $D_R A D_C$.

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

Otherwise, if INFO ≥ 0, EQUED specifies the form of equilibration that was performed as specified above.

Constraint: if FACT = 'F' , EQUED = 'N', 'R', 'C' or 'B'.

13: R($*$) – ***double precision*** array *Input/Output*

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

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

If FACT = 'F' and EQUED = 'R' or 'B', R must contain the row scale factors for A, D_R; each element of R must be positive.

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

Otherwise, if INFO ≥ 0 and EQUED = 'R' or 'B', R contains the row scale factors for A, D_R, such that A is multiplied on the left by D_R; each element of R is positive.

14: C($*$) – ***double precision*** array *Input/Output*

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

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

If FACT = 'F' or EQUED = 'C' or 'B', C must contain the column scale factors for A, D_C; each element of C must be positive.

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

Otherwise, if INFO ≥ 0 and EQUED = 'C' or 'B', C contains the row scale factors for A, D_C; each element of C is positive.

15: B(LDB,$*$) – ***complex*16*** array *Input/Output*

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.

On exit: if EQUED = 'N', B is not modified.

If TRANS = 'N' and EQUED = 'R' or 'B', B is overwritten by $D_R B$.

If TRANS = 'T' or 'C' and EQUED = 'C' or 'B', B is overwritten by $D_C B$.

16: LDB – INTEGER *Input*

On entry: the first dimension of the array B as declared in the (sub)program from which F07BPF (ZGBSVX) is called.

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

17: X(LDX,$*$) – ***complex*16*** array *Output*

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

On exit: if INFO $= 0$ or INFO $= \mathrm{N} + 1$, 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 $\neq$ 'N', and the solution to the equilibrated system is $D_C^{-1}X$ if TRANS = 'N' and EQUED = 'C' or 'B', or $D_R^{-1}X$ if TRANS = 'T' or 'C' and EQUED = 'R' or 'B'.

18: LDX – INTEGER *Input*

On entry: the first dimension of the array X as declared in the (sub)program from which F07BPF (ZGBSVX) is called.

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

19: RCOND – ***double precision*** *Output*

On exit: if INFO ≥ 0, an estimate of the reciprocal condition number of the matrix A (after equilibration if that is performed), computed as $\mathrm{RCOND} = 1/\left(\|A\|_1\|A^{-1}\|_1\right)$.

20: FERR($*$) – ***double precision*** array *Output*

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

On exit: if INFO $= 0$ or INFO $= \mathrm{N} + 1$, 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.

21: BERR($*$) – ***double precision*** array *Output*

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

On exit: if INFO $= 0$ or INFO $= \mathrm{N} + 1$, an estimate of the componentwise 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).

22: WORK($*$) – ***complex*16*** array *Workspace*

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

23: RWORK($*$) – ***double precision*** array *Output*

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

On exit: if INFO $= 0$, $\mathrm{RWORK}(1)$ contains the reciprocal pivot growth factor $\max|a_{ij}|/\max|u_{ij}|$. If $\mathrm{RWORK}(1)$ is much less than 1, then the stability of the LU factorization of the (equilibrated) matrix A could be poor. This also means that the solution X, condition estimator RCOND, and forward error bound FERR could be unreliable. If the factorization fails with $0 < \mathrm{INFO} \leq \mathrm{N}$, $\mathrm{RWORK}(1)$ contains the reciprocal pivot growth factor for the leading INFO columns of A.

24: 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 $<$ N

If INFO $= i$, u_{ii} is exactly zero. The factorization has been completed, but the factor U is exactly singular, so the solution could not be computed.

INFO $=$ N $+ 1$

U is nonsingular, but RCOND is less than ***machine precision***, so that the matrix A is numerically singular. A solution to the equations $AX = B$, and corresponding error bounds, have nevertheless been computed because there are some situations where the computed solution can be more accurate that 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

$$|E| \leq c(n)\epsilon P|L||U|,$$

$c(n)$ is a modest linear function of n, and ϵ is the ***machine precision***. See Section 9.3 of Higham (2002) for further details.

If x is the true solution, then the computed solution $\hat{x}$ 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 band storage scheme for the array AB is illustrated by the following example, when $n = 6$, $k_l = 1$, and $k_u = 2$. Storage of the band matrix A in the array AB:

$$\begin{array}{cccccc}
* & * & a_{13} & a_{24} & a_{35} & a_{46} \\
* & a_{12} & a_{23} & a_{34} & a_{45} & a_{56} \\
a_{11} & a_{22} & a_{33} & a_{44} & a_{55} & a_{66} \\
a_{21} & a_{32} & a_{43} & a_{54} & a_{65} & *
\end{array}$$

The total number of floating-point operations required to solve the equations $AX = B$ depends upon the pivoting required, but if $n \gg k_l + k_u$ then it is approximately bounded by $O(nk_l(k_l + k_u))$ for the factorization and $O(n(2k_l + k_u)r)$ for the solution following the factorization. The condition number estimation typically requires between four and five solves and never more than eleven solves, following the factorization. The solution is then refined, and the errors estimated, using iterative refinement; see F07BVF (ZGBRFS) for information on the floating-point operations required.

In practice the condition number estimator is very reliable, but it can underestimate the true condition number; see Section 15.3 of Higham (2002) for further details.

The real analogue of this routine is F07BBF (DGBSVX).

9 Example

To solve the equations

$$AX = B,$$

where A is the band matrix

$$A = \begin{pmatrix} -1.65+2.26i & -2.05-0.85i & 0.97-2.84i & 0 \\ 6.30i & -1.48-1.75i & -3.99+4.01i & 0.59-0.48i \\ 0 & -0.77+2.83i & -1.06+1.94i & 3.33-1.04i \\ 0 & 0 & 4.48-1.09i & -0.46-1.72i \end{pmatrix}$$

and

$$B = \begin{pmatrix} -1.06+21.50i & 12.85+\ \ 2.84i \\ -22.72-53.90i & -70.22+21.57i \\ 28.24-38.60i & -20.73-\ \ 1.23i \\ -34.56+16.73i & 26.01+31.97i \end{pmatrix}.$$

Estimates for the backward errors, forward errors, condition number and pivot growth are also output, together with information on the equilibration of A.

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.

```
*     F07BPF Example Program Text
*     Mark 21 Release. NAG Copyright 2004.
*     .. Parameters ..
      INTEGER          NIN, NOUT
      PARAMETER        (NIN=5,NOUT=6)
      INTEGER          NMAX, KLMAX, KUMAX, NRHSMX
      PARAMETER        (NMAX=8,KLMAX=4,KUMAX=4,NRHSMX=2)
      INTEGER          LDAB, LDAFB, LDB, LDX
      PARAMETER        (LDAB=KLMAX+KUMAX+1,LDAFB=LDAB+KLMAX,LDB=NMAX,
     +                 LDX=NMAX)
*     .. Local Scalars ..
      DOUBLE PRECISION RCOND
      INTEGER          I, IFAIL, INFO, J, K, KL, KU, N, NRHS
      CHARACTER        EQUED
*     .. Local Arrays ..
      COMPLEX *16      AB(LDAB,NMAX), AFB(LDAFB,NMAX), B(LDB,NRHSMX),
     +                 WORK(2*NMAX), X(LDX,NRHSMX)
      DOUBLE PRECISION BERR(NRHSMX), C(NMAX), FERR(NRHSMX), R(NMAX),
     +                 RWORK(NMAX)
      INTEGER          IPIV(NMAX)
      CHARACTER        CLABS(1), RLABS(1)
*     .. External Subroutines ..
      EXTERNAL         X04DBF, ZGBSVX
*     .. Intrinsic Functions ..
      INTRINSIC        MAX, MIN
*     .. Executable Statements ..
      WRITE (NOUT,*) 'F07BPF Example Program Results'
      WRITE (NOUT,*)
*     Skip heading in data file
      READ (NIN,*)
      READ (NIN,*) N, NRHS, KL, KU
      IF (N.LE.NMAX .AND. KL.LE.KLMAX .AND. KU.LE.KUMAX .AND. NRHS.LE.
     +    NRHSMX) THEN
*
*        Read the band matrix A and B from data file
*
         K = KU + 1
         READ (NIN,*) ((AB(K+I-J,J),J=MAX(I-KL,1),MIN(I+KU,N)),I=1,N)
         READ (NIN,*) ((B(I,J),J=1,NRHS),I=1,N)
*
*        Solve the equations AX = B for X
*
         CALL ZGBSVX('Equilibration','No transpose',N,KL,KU,NRHS,AB,
     +               LDAB,AFB,LDAFB,IPIV,EQUED,R,C,B,LDB,X,LDX,RCOND,
     +               FERR,BERR,WORK,RWORK,INFO)
*
```

```
         IF ((INFO.EQ.0) .OR. (INFO.EQ.N+1)) THEN
*
*           Print solution, error bounds, condition number, the form
*           of equilibration and the pivot growth factor
*
            IFAIL = 0
            CALL X04DBF('General',' ',N,NRHS,X,LDX,'Bracketed','F7.4',
     +                  'Solution(s)','Integer',RLABS,'Integer',CLABS,
     +                  80,0,IFAIL)
*
            WRITE (NOUT,*)
            WRITE (NOUT,*) 'Backward errors (machine-dependent)'
            WRITE (NOUT,99999) (BERR(J),J=1,NRHS)
            WRITE (NOUT,*)
            WRITE (NOUT,*)
     +        'Estimated forward error bounds (machine-dependent)'
            WRITE (NOUT,99999) (FERR(J),J=1,NRHS)
            WRITE (NOUT,*)
            WRITE (NOUT,*) 'Estimate of reciprocal condition number'
            WRITE (NOUT,99999) RCOND
            WRITE (NOUT,*)
            IF (EQUED.EQ.'N') THEN
               WRITE (NOUT,*) 'A has not been equilibrated'
            ELSE IF (EQUED.EQ.'R') THEN
               WRITE (NOUT,*) 'A has been row scaled as diag(R)*A'
            ELSE IF (EQUED.EQ.'C') THEN
               WRITE (NOUT,*) 'A has been column scaled as A*diag(C)'
            ELSE IF (EQUED.EQ.'B') THEN
               WRITE (NOUT,*)
     +           'A has been row and column scaled as diag(R)*A*diag(C)'
            END IF
            WRITE (NOUT,*)
            WRITE (NOUT,*) 'Estimate of reciprocal pivot growth factor'
            WRITE (NOUT,99999) RWORK(1)
*
            IF (INFO.EQ.N+1) THEN
               WRITE (NOUT,*)
               WRITE (NOUT,*)
     +           'The matrix A is singular to working precision'
            END IF
         ELSE
            WRITE (NOUT,99998) 'The (', INFO, ',', INFO, ')',
     +        ' element of the factor U is zero'
         END IF
      ELSE
         WRITE (NOUT,*)
     +     'One or more of NMAX, KLMAX, KUMAX or NRHSMX is too small'
      END IF
      STOP
*
99999 FORMAT ((3X,1P,7E11.1))
99998 FORMAT (1X,A,I3,A,I3,A,A)
      END
```

9.2 Program Data

```
F07BPF Example Program Data
  4  2  1  2                                       :Values of N, NRHS, KL and KU
 (-1.65, 2.26) (-2.05,-0.85) ( 0.97,-2.84)
 ( 0.00, 6.30) (-1.48,-1.75) (-3.99, 4.01) ( 0.59,-0.48)
               (-0.77, 2.83) (-1.06, 1.94) ( 3.33,-1.04)
                             ( 4.48,-1.09) (-0.46,-1.72)  :End of matrix A
 ( -1.06, 21.50) ( 12.85,  2.84)
 (-22.72,-53.90) (-70.22, 21.57)
 ( 28.24,-38.60) (-20.73, -1.23)
 (-34.56, 16.73) ( 26.01, 31.97)                          :End of matrix B
```

9.3 Program Results

```
F07BPF Example Program Results

Solution(s)
                    1                  2
1  (-3.0000, 2.0000) ( 1.0000, 6.0000)
2  ( 1.0000,-7.0000) (-7.0000,-4.0000)
3  (-5.0000, 4.0000) ( 3.0000, 5.0000)
4  ( 6.0000,-8.0000) (-8.0000, 2.0000)

Backward errors (machine-dependent)
      9.8E-17    3.4E-17

Estimated forward error bounds (machine-dependent)
      3.7E-14    4.3E-14

Estimate of reciprocal condition number
      9.6E-03

A has not been equilibrated

Estimate of reciprocal pivot growth factor
      1.0E+00
```