

# F11BAFP

## NAG Parallel Library Routine Document

**Note:** Before using this routine, please read the Users' Note for your implementation to check for implementation-dependent details. You are advised to enclose any calls to NAG Parallel Library routines between calls to Z01AAFP and Z01ABFP.

### 1 Description

**Note:** you should read the F11 Chapter Introduction before trying to use this routine. In particular, some of the notation and terminology used in this document was introduced in Section 2.2 of the F11 Chapter Introduction.

F11BAFP is the first in a suite of three routines for the iterative solution of a real general (unsymmetric) system of simultaneous linear equations  $Ax = b$  using the Restarted Generalised Minimal Residual method (RGMRES) (Saad and Schultz [3], Barrett *et al.* [1], Dias da Cunha and Hopkins [2]). F11BAFP is a set-up routine which must be called before F11BBFP, the iterative solver. The third routine in the suite, F11BCFP can be used to return additional information about the computation. These three routines are suitable for the solution of large sparse systems of equations.

### 2 Specification

```

SUBROUTINE F11BAFP(ICNTXT, METHOD, PRECON, NORM, DISTR, WEIGHT,
1                 ITERM, N, NLOC, M, TOL, MAXITN, ANORM, SIGMAX,
2                 MONIT, LWREQ, IFAIL)
DOUBLE PRECISION TOL, ANORM, SIGMAX
INTEGER          ICNTXT, ITERM, N, NLOC, M, MAXITN, MONIT,
1               LWREQ, IFAIL
CHARACTER*1     PRECON, NORM, DISTR, WEIGHT
CHARACTER*(*)  METHOD

```

### 3 Data Distribution

#### 3.1 Definitions

The following definitions are used in describing the data distribution within this document:

- $m_p$  – the number of rows in the logical grid of processors.
- $n_p$  – the number of columns in the logical grid of processors.
- $n$  – the order of the matrix  $A$ .
- $n_l(i, j)$  – the number of elements of the distributed vectors stored locally on the processor at location  $\{i, j\}$  of the grid.
- $m$  – the size of the RGMRES basis used: every  $m$  iterations, the solution procedure will be restarted. (See also Section 2.2 of the F11 Chapter Introduction.)

#### 3.2 Global and Local Arguments

The input arguments METHOD, PRECON, NORM, DISTR, WEIGHT, ITERM, N, M, TOL, MAXITN, ANORM, SIGMAX, MONIT and IFAIL are global and so must have the same value on entry to the routine on each processor. The output argument IFAIL is global and so will have the same value on exit from the routine on each processor. The argument NLOC must have the same value on all processors in a row or a column of the grid, if vectors are distributed along the columns or the rows of the grid, respectively. The remaining arguments are local.

## 4 Arguments

- 1:** ICNTXT — INTEGER *Local Input*  
*On entry:* the BLACS context used by the communication mechanism, usually returned by a call to Z01AAFP.
- 2:** METHOD — CHARACTER\*(\*) *Global Input*  
*On entry:* the iterative method to be used. The only possible choice in this Release is:  
 'RGMRES' Restarted Generalised Minimal Residual (RGMRES) method.  
*Constraint:* METHOD = 'RGMRES'.
- 3:** PRECON — CHARACTER\*1 *Global Input*  
*On entry:* determines whether preconditioning is used. The possible choices are:  
 'N' no preconditioning;  
 'P' preconditioning.  
*Constraint:* PRECON = 'N' or 'P'.
- 4:** NORM — CHARACTER\*1 *Global Input*  
*On entry:* defines the matrix and vector norm to be used in the termination criteria. The possible choices are:  
 '1' use the  $l_1$ -norm;  
 'I' use the  $l_\infty$ -norm;  
 '2' use the  $l_2$ -norm.  
*Suggested value:* NORM = '1', if ITERM = 1; NORM = '2', if ITERM = 2.  
*Constraint:* if ITERM = 1, then NORM = '1', 'I' or '2'; if ITERM = 2, then NORM = '2'.
- 5:** DISTR — CHARACTER\*1 *Global Input*  
*On entry:* defines how vectors are distributed across the processors in the grid (see Section 2.5.3 of the F11 Chapter Introduction). The possible choices are:  
 'A' vectors are distributed across all processors in the grid;  
 'C' vectors are distributed by column;  
 'R' vectors are distributed by row.  
 DISTR = 'A' must be chosen if F11BAFP is used in connection with any F11 routine other than F11BBFP and F11BCFP.  
*Suggested value:* DISTR = 'A'.  
*Constraint:* DISTR = 'A', 'C' or 'R'.
- 6:** WEIGHT — CHARACTER\*1 *Global Input*  
*On entry:* specifies whether a vector  $w$  of user-supplied weights is to be used in the vector norms used in the computation of termination criteria:  $\|v\|_p^{(w)} = \|v^{(w)}\|_p$ , where  $v_i^{(w)} = w_i v_i$ , for  $i = 1, \dots, n$ . The suffix  $p = 1, 2, \infty$  denotes the vector norm used, as specified by the parameter NORM. Note that weights cannot be used when ITERM = 2. The possible choices are:  
 'W' user-supplied weights to be used;  
 'N' all weights are implicitly set equal to one. Weights do not need to be supplied on initial entry to F11BBFP.  
 See also Section 4 of the document for F11BBFP.  
*Suggested value:* WEIGHT = 'N'.  
*Constraint:* if ITERM = 1, then WEIGHT = 'W' or 'N'; if ITERM = 2, then WEIGHT = 'N'.

- 7: ITERM — INTEGER** *Global Input*  
*On entry:* defines the termination criterion to be used. The possible choices are:  
 ITERM = 1:  $\|r_k\|_p \leq \tau (\|b\|_p + \|A\|_p \|x_k\|_p)$ , where  $p = 1, \infty$  or 2;  
 ITERM = 2:  $\|M^{-1}r_k\|_2 \leq \tau (\|M^{-1}b\|_2 + \sigma_1(M^{-1}A)\|x_k\|_2)$ .  
*Suggested value:* ITERM = 2.  
*Constraint:* ITERM = 1 or 2.
- 8: N — INTEGER** *Global Input*  
*On entry:* the order  $n$  of the matrix  $A$ .  
*Constraint:*  $N \geq 0$ .
- 9: NLOC — INTEGER** *Local Input*  
*On entry:* the number of vector elements stored locally, i.e.,  $NLOC = n_l(i, j)$ , where  $i, j$  are the row and column indices, respectively, of the calling processor. Note that information about the distribution pattern is not required: only the number of vector elements stored locally must be supplied.  
*Constraint:*  $NLOC \geq 0$  and, according to the value of DISTR:
- $$\text{DISTR} = \text{'A'}: \sum_{i=0}^{m_p-1} \sum_{j=0}^{n_p-1} n_l(i, j) = n;$$
- $$\text{DISTR} = \text{'C'}: \sum_{i=0}^{m_p-1} n_l(i, j) = n, \text{ for } j = 0, \dots, n_p - 1 \text{ and } n_l(i, 0) = n_l(i, 1) = \dots = n_l(i, n_p - 1) \\ \text{for } i = 0, \dots, m_p - 1;$$
- $$\text{DISTR} = \text{'R'}: \sum_{j=0}^{n_p-1} n_l(i, j) = n, \text{ for } i = 0, \dots, m_p - 1 \text{ and } n_l(0, j) = n_l(1, j) = \dots = n_l(m_p - 1, j) \\ \text{for } j = 0, \dots, n_p - 1.$$
- 10: M — INTEGER** *Global Input*  
*On entry:* the size  $m$  of the RGMRES basis used: every  $m$  iterations, the solution procedure will be restarted.  
*Suggested value:*  $M = 20$ . However, the optimal value of  $M$  may depend very strongly on the matrix and on the preconditioner used.  
*Constraint:*  $M \geq 0$ .
- 11: TOL — DOUBLE PRECISION** *Global Input*  
*On entry:* the tolerance  $\tau$  for the termination criterion. If  $TOL \leq 0.0$ , the default value  $TOL = \sqrt{\epsilon}$  is used, where  $\epsilon$  is the **machine precision**. Otherwise, the actual value used is  $\max(TOL, \epsilon)$ .  
*Constraint:*  $TOL < 1.0$ .
- 12: MAXITN — INTEGER** *Global Input*  
*On entry:* the maximum number of iterations.  
*Constraint:*  $MAXITN \geq 0$ .
- 13: ANORM — DOUBLE PRECISION** *Global Input*  
*On entry:* the value of  $\|A\|_p$  to be used in the termination criterion when ITERM = 1, where  $p = 1, \infty, 2$ .  
 If  $ANORM \leq 0.0$ , ITERM = 1 and NORM = '1' or 'I', then  $\|A\|_1$  or  $\|A\|_\infty$ , respectively, is estimated internally by F11BBFP.  
 If ITERM = 2, then ANORM is not referenced.  
*Constraint:* if ITERM = 1 and NORM = '2', then  $ANORM > 0.0$ .

**14: SIGMAX — DOUBLE PRECISION** *Global Input*

*On entry:* the value of  $\sigma_1(\bar{A}) = \|M^{-1}A\|_2$ .

If either  $\text{SIGMAX} \leq 0.0$  and  $\text{ITERM} = 2$  or user-supplied weights are not used, then  $\sigma_1(\bar{A})$  is estimated internally by F11BBFP (see also the parameter WEIGHT below).

If  $\text{ITERM} = 1$  and  $\text{WEIGHT} = 'N'$ , then SIGMAX is not referenced.

**15: MONIT — INTEGER** *Global Input*

*On entry:* the frequency at which the monitoring step is executed by F11BBFP. When the RGMRES method is used, monitoring is possible only every  $m$  iterations, that is, at the end of a super-iteration (see Section 2.2 of the F11 Chapter Introduction).

If  $\text{MONIT} \geq m$  and  $k = \lfloor \text{MONIT}/m \rfloor$  (integer part) then F11BBFP will monitor the solution every  $k$  super-iterations, starting from the  $k$ th super-iteration.

If  $\text{MONIT} < m$ , then no monitoring takes place.

*Constraint:*  $\text{MONIT} \leq \text{MAXITN}$ .

**16: LWREQ — INTEGER** *Local Output*

*On exit:* the amount of workspace required by F11BBFP.

See also Section 4 of the document for F11BBFP.

**17: IFAIL — INTEGER** *Global Input/Global Output*

*On entry:* IFAIL must be set to 0, -1 or 1. For users not familiar with this parameter (described in the Essential Introduction) the recommended values are:

IFAIL = 0, if multigridding is **not** employed;

IFAIL = -1, if multigridding is employed.

*On exit:* IFAIL = 0 unless the routine detects an error (see Section 5).

## 5 Errors and Warnings

If on entry IFAIL = 0 or -1, explanatory error messages are output from the root processor (or processor {0,0} when the root processor is not available) on the current error message unit (as defined by X04AAF).

Errors detected by the routine:

IFAIL = -2000

The routine has been called with an invalid value of ICNTXT on one or more processors.

IFAIL = -1000

The logical processor grid and library mechanism (Library Grid) have not been correctly defined, see Z01AAFP.

IFAIL = - $i$

On entry, the  $i$ th argument had illegal value(s) on one or more processors. For global arguments, this may also be caused by the  $i$ th argument not having the same value on **all** processors (see also Section 3.2).

IFAIL = 1

F11BAFP has been called out of sequence. Either F11BAFP has been called twice without calling F11BBFP in between, or F11BBFP has not completed its current task.

## 6 Further Comments

To clarify the compatibility between the parameters ITERM, NORM, ANORM, SIGMAX and PRECON, the following are valid combinations:

ITERM = 1  
 if NORM = '1' or 'I', any value of ANORM is allowed;  
 if NORM = '2', then  $\|A\|_2$  must be supplied, i.e., ANORM > 0.0.  
 ITERM = 2  
 NORM = '2' and any value of SIGMAX is allowed.

If no preconditioner is used (PRECON = 'N'), then  $\bar{A} = A$  and  $\sigma_1(\bar{A}) = \sigma_1(A) = \|A\|_2$ .

The first termination criterion, corresponding to ITERM = 1, can be applied during any iteration, while the second criterion, corresponding to ITERM = 2, can only be applied if user-specified weights are not used, otherwise it can only be applied at the end of a super-iteration.

When  $\sigma_1(\bar{A})$  is not supplied on input (SIGMAX  $\leq$  0.0) and it is required by the termination criterion, the inexpensive estimate  $\sigma_1(\bar{A}) \approx \max_k \|U_k\|_1$  is used, where  $\{U_k\}$  is a sequence of upper triangular matrices orthogonally similar to the sequence of upper Hessenberg matrices  $\{H_k\}$  generated during the iteration (see also Sections 2.1 and 2.2 of the F11 Chapter Introduction). Note that only order of magnitude estimates are required by the termination criterion.

### 6.1 Algorithmic Detail

Not applicable.

### 6.2 Parallelism Detail

Not applicable.

### 6.3 Accuracy

Not applicable.

### 6.4 Computational costs

The computational costs of F11BAFP are negligible compared to the costs of F11BBFP.

## 7 References

- [1] Barrett R, Berry M, Chan T F, Demmel J, Donato J, Dongarra J, Eijkhout V, Pozo R, Romine C and van der Vorst H (1994) *Templates for the Solution of Linear Systems: Building Blocks for Iterative Methods* SIAM, Philadelphia
- [2] Dias da Cunha R and Hopkins T (1994) PIM 1.1 — the parallel iterative method package for systems of linear equations user's guide — Fortran 77 version *Technical Report* Computing Laboratory, University of Kent at Canterbury, Kent CT2 7NZ, UK
- [3] Saad Y and Schultz M (1986) GMRES: A generalized minimal residual algorithm for solving nonsymmetric linear systems *SIAM J. Sci. Statist. Comput.* **7** 856–869

## 8 Example

To solve the linear system of equations  $Ax = b$  of order  $n = n_x^2$ , where  $n_x$  is a user-specified integer. The symmetric sparse coefficient matrix  $A$  is given by the block matrix

$$A = \begin{pmatrix} D & E & 0 & \cdots & \cdots & 0 \\ F & D & E & \ddots & & \vdots \\ 0 & F & D & \ddots & & \\ \vdots & \ddots & \ddots & \ddots & \ddots & \vdots \\ & & & \ddots & D & E & 0 \\ \vdots & & & \ddots & E & D & E \\ 0 & \cdots & \cdots & 0 & F & D \end{pmatrix},$$

where the matrix blocks  $D$ ,  $E$  and  $F$  of order  $n_x$  are defined in terms of the quantity  $h := (n_x + 1)^{-1}$  as follows:

$$D = \frac{1}{h^2} \begin{pmatrix} 4 & -h-1 & 0 & \cdots & \cdots & 0 \\ h-1 & 4 & -h-1 & \ddots & & \vdots \\ 0 & h-1 & 4 & \ddots & & \\ \vdots & \ddots & \ddots & \ddots & \ddots & \vdots \\ & & & \ddots & 4 & -h-1 & 0 \\ \vdots & & & \ddots & h-1 & 4 & -h-1 \\ 0 & \cdots & \cdots & 0 & h-1 & 4 \end{pmatrix},$$

$$E = -\frac{h+1}{h^2} I$$

and

$$F = \frac{h-1}{h^2} I,$$

where  $I$  denotes the identity matrix. The right-hand side vector is given by  $b = 10^2(1, 1, \dots, 1)^T$ .

**Note:** the listing of the Example Program presented below does not give a full pathname for the data file being opened, but in general the user must give the full pathname in this and any other OPEN statement.

### 8.1 Example Text

```
* F11BAFP Example Program Text
* NAG Parallel Library Release 2. NAG Copyright 1996.
* .. Parameters ..
INTEGER          NIN, NOUT
PARAMETER       (NIN=5,NOUT=6)
INTEGER          MLMAX, NLBMAX
PARAMETER       (MLMAX=1000,NLBMAX=10)
INTEGER          LA, LC
PARAMETER       (LA=10*MLMAX,LC=2*LA)
INTEGER          LIA, LWORK
PARAMETER       (LIA=2*LA,LWORK=20*MLMAX)
* .. Local Scalars ..
DOUBLE PRECISION ANORM, SIGMAX, STPLHS, STPRHS, TOL
INTEGER          I, ICNTXT, IFAIL, IREVCM, ITERM, ITN, IW, K, LW,
+               LWREQ, M, MAXITN, MB, ML, MLO, MLOMAX, MONIT, MP,
+               MYCOL, MYROW, N, NINTE, NINTI, NLB, NNZ, NNZC,
+               NP, NX
LOGICAL          LOOP, ROOT
CHARACTER        CHECK, DISTR, DUP, NORM, OPTIM, PRECON, SYMM,
+               WEIGHT, ZERO
```

```

      CHARACTER*10      METHOD
      CHARACTER*80      FORMAT
*
* .. Local Arrays ..
      DOUBLE PRECISION A(LA), C(LC), DTOL(NLBMAX), U(MLMAX), V(MLMAX),
+      WORK(LWORK)
      INTEGER          CA(1), IAINFO(LIA), ICOL(LA), ICOLC(LC), IERR(2),
+      IPIVP(MLMAX), IPIVQ(MLMAX), IROW(LA), IROWC(LC),
+      LFILL(NLBMAX), NPIVM(NLBMAX), RA(1)
      CHARACTER        MILU(NLBMAX), PSTRAT(NLBMAX)
*
* .. External Functions ..
      LOGICAL          Z01ACFP
      EXTERNAL         Z01ACFP
*
* .. External Subroutines ..
      EXTERNAL         BLACS_GRIDINFO, F01YAFP, F01YEFP, F11BAFP,
+      F11BBFP, F11BCFP, F11DAFP, F11DBFP, F11XAFP,
+      F11XBFP, F11ZAFP, GMAT, GVEC, PRINTI, X04YAFP,
+      Z01AAFP, Z01ABFP
*
* .. Executable Statements ..
      ROOT = Z01ACFP()
      IF (ROOT) THEN
         WRITE (NOUT,*) 'F11BAFP Example Program Results'
      END IF
*
* Open the input file on all processors
*
      OPEN (NIN,FILE='f11bafpe.d')
*
* Skip heading in data file
*
      READ (NIN,*)
      READ (NIN,*) MP, NP
*
* Create the processes and initialize
*
      IFAIL = 0
      CALL Z01AAFP(ICNTXT,MP,NP,IFAIL)
      CALL BLACS_GRIDINFO(ICNTXT,MP,NP,MYROW,MYCOL)
*
* Initialize some global data
*
      DUP = 'F'
      ZERO = 'R'
      SYMM = 'S'
      OPTIM = 'S'
      WEIGHT = 'N'
      DISTR = 'A'
      ANORM = 0.DO
      SIGMAX = 0.DO
      CHECK = 'N'
      ITN = 0
*
* Read the problem parameters
*
      READ (NIN,*) NX
      N = NX**2
*
* Read the algorithmic parameters
*

```

```

      READ (NIN,*) METHOD
      READ (NIN,*) PRECON, NORM, ITERM, MONIT
      READ (NIN,*) M, MB
      READ (NIN,*) TOL, MAXITN
      READ (NIN,*) FORMAT
*
*   Close the input file
*
      CLOSE (NIN)
*
*   Generate the matrix of the coefficients
*
      CALL F01YAFP(ICNTXT,GMAT,N,MB,NNZ,A,LA,IROW,ICOL,IFAIL)
*
*   Set up auxiliary data for subsequent operations
*
      CALL F11ZAFP(ICNTXT,N,MB,NNZ,A,IROW,ICOL,DUP,ZERO,IAINFO,LIA,
+                IFAIL)
*
      ML = IAINFO(3)
      NLB = IAINFO(8)
*
*   Check whether number of rows and number of row blocks are
*   less than the corresponding maximum possible values defined by
*   MLMAX and NLBMAX
*
      IERR(1) = 0
      IERR(2) = 0
      IF (ML.GT.MLMAX) IERR(1) = 1
      IF (NLB.GT.NLBMAX) IERR(2) = 1
      CALL IGAMX2D(ICNTXT,'All',' ',1,1,IERR,2,RA,CA,1,-1,-1)
      IF (IERR(1).NE.0) THEN
         IF (ROOT) WRITE (NOUT,FMT=99996)
         GO TO 80
      ELSE IF (IERR(2).NE.0) THEN
         IF (ROOT) WRITE (NOUT,FMT=99995)
         GO TO 80
      END IF
*
*   Generate the right-hand side vector
*
      CALL F01YEFP(ICNTXT,GVEC,N,V,IAINFO,IFAIL)
*
*   Set up auxiliary data for matrix-vector multiplication
*
      CALL F11XAFP(ICNTXT,N,NNZ,A,IROW,ICOL,SYMM,OPTIM,IAINFO,LIA,IFAIL)
*
*   Set up block Jacobi preconditioner
*
      DO 20 K = 1, NLB
         LFILL(K) = -1
         DTOL(K) = 1.D-3
         PSTRAT(K) = 'C'
         MILU(K) = 'M'
20  CONTINUE

      CALL F11DAFP(ICNTXT,N,NNZ,A,IROW,ICOL,LFILL,DTOL,PSTRAT,MILU,
+                IPIVP,IPIVQ,NNZC,C,LC,IROWC,ICOLC,NPIVM,IAINFO,LIA,

```

```

+           IFAIL)
*
*   Initialize the suite
*
CALL F11BAFP(ICNTXT,METHOD,PRECON,NORM,DISTR,WEIGHT,ITERM,N,ML,M,
+           TOL,MAXITN,ANORM,SIGMAX,MONIT,LWREQ,IFAIL)
*
*   Check workspace size
*
NINTI = IAINFO(6)
NINTE = IAINFO(7)
MLO = IAINFO(4)
MLOMAX = IAINFO(5)
IERR(1) = 0
IF (ROOT) THEN
  IF ((LWREQ+MAX(NINTI,NINTE,MLO+MLOMAX)).GT.LWORK) IERR(1) = 1
ELSE
  IF ((LWREQ+MAX(NINTI,NINTE,2*MLO)).GT.LWORK) IERR(1) = 1
END IF
CALL IGAMX2D(ICNTXT,'All',' ',1,1,IERR,1,RA,CA,1,-1,-1)
IF (IERR(1).NE.0) THEN
  WRITE (NOUT,FMT=99994)
  GO TO 80
END IF
*
*   Print a summary of input parameters and options
*
IF (ROOT) CALL PRINTI(NOUT,METHOD,PRECON,NORM,DISTR,ITERM,N,
+           MAXITN,TOL,M,MONIT)
*
*   Set initial approximation to solution
*
DO 40 I = 1, ML
  U(I) = 0.DO
40 CONTINUE
*
*   Solve the equations (F11BBFP)
*
LW = LWREQ
IW = LWREQ + 1
IREVCM = 0
*
LOOP = .TRUE.
60 CONTINUE

CALL F11BBFP(ICNTXT,IREVCM,U,V,WORK,LW,IFAIL)
*
IF (IREVCM.LE.-1) THEN
*
*   Compute  $v = A' * u$  (used only to compute  $\|A\|$ )
*
CALL F11XBFP(ICNTXT,'Transpose',N,NNZ,A,IROW,ICOL,CHECK,U,V,
+           IAINFO,WORK(IW),IFAIL)
*
ELSE IF (IREVCM.EQ.1) THEN
*
*   Compute  $v = A * u$ 
*

```

```

      CALL F11XBFP(ICNTXT,'No transpose',N,NNZ,A,IROW,ICOL,CHECK,U,V,
+               IAINFO,WORK(IW),IFAIL)
*
      ELSE IF (IREVCM.EQ.2) THEN
*
*      Solve  $M * v = u$ 
*
      CALL F11DBFP(ICNTXT,'No transpose',N,NNZC,C,IROWC,ICOLC,IPIVP,
+               IPIVQ,CHECK,U,V,IAINFO,WORK(IW),IFAIL)
*
*
      ELSE IF (IREVCM.EQ.3) THEN
*
*      Monitoring
*
      ITN = ITN + M*(MONIT/M)
      IF (ROOT) THEN
        WRITE (NOUT,'(/1X,'Monitoring at iteration no. ',I3)')
+       ITN
        WRITE (NOUT,
+       '(/1X,'Solution vector (last iterate)'/1X,30(''-')/)'')
      END IF
      CALL X04YAFP(ICNTXT,NOUT,N,U,FORMAT,IAINFO,WORK(IW),IFAIL)
      IF (ROOT) WRITE (NOUT,
+       '(/1X,'Residual vector (last iterate)'/1X,30(''-')/)'')
      CALL X04YAFP(ICNTXT,NOUT,N,V,FORMAT,IAINFO,WORK(IW),IFAIL)
*
      ELSE IF (IREVCM.EQ.4) THEN
*
*      Termination
*
      LOOP = .FALSE.
      END IF
      IF (LOOP) GO TO 60
*
*      Get additional information
*
      IFAIL = 0
      CALL F11BCFP(ICNTXT,ITN,STPLHS,STPRHS,ANORM,SIGMAX,IFAIL)
*
*      Produce a report
*
      IF (ROOT) THEN
        WRITE (NOUT,'(/1X,'Summary of results''/1X,18(''-')/)'')
        WRITE (NOUT,99999)
+       'Number of iterations carried out           -', ITN
        WRITE (NOUT,99997)
+       'Left-hand side of the termination criterion -',
+       STPLHS
        WRITE (NOUT,99997)
+       'Right-hand side of the termination criterion -',
+       STPRHS
        WRITE (NOUT,99998)
+       '1-norm of the matrix of the coefficients   -',
+       ANORM
        WRITE (NOUT,99998)
+       'Largest singular value of the preconditioned matrix -',
+       SIGMAX

```

```

        WRITE (NOUT, '(/1X, ''Solution vector''/1X,15(''-''/))')
    END IF
    CALL X04YAFP(ICNTXT,NOUT,N,U,FORMAT,IAINFO,WORK(IW),IFAIL)
    IF (ROOT) WRITE (NOUT,
+   '(/1X, ''Residual vector''/1X,15(''-''/))')
    CALL X04YAFP(ICNTXT,NOUT,N,V,FORMAT,IAINFO,WORK(IW),IFAIL)
*
*   Completion
*
    80 CALL Z01ABFP(ICNTXT,'N',IFAIL)
*
*   End of example program
*
    STOP
*

99999 FORMAT (1X,A,I5)
99998 FORMAT (1X,A,2X,F7.3)
99997 FORMAT (1X,A,3X,1P,D9.2)
99996 FORMAT (1X, '** ERROR: Number of rows per processor too large')
99995 FORMAT (1X, '** ERROR: Number of row blocks per processor too ',
+   'large')
99994 FORMAT (1X, '** ERROR: LWORK too small')

    END
*
*****
*****
*
    SUBROUTINE PRINTI(NOUT,METHOD,PRECON,NORM,DISTR,ITERM,N,MAXITN,
+   TOL,M,MONIT)
*
*   Prints a summary of the input parameters and options.
*
*
*   .. Scalar Arguments ..
    DOUBLE PRECISION  TOL
    INTEGER            ITERM, M, MAXITN, MONIT, N, NOUT
    CHARACTER          DISTR, NORM, PRECON
    CHARACTER*10      METHOD
*
*   .. Executable Statements ..
    WRITE (NOUT,99999)
    WRITE (NOUT,99998)
+   'Method used (METHOD)                -',
+   METHOD
    WRITE (NOUT,99998)
+   'Use the preconditioner (PRECON)    -',
+   PRECON
    WRITE (NOUT,99998)
+   'Matrix and vector norm in use (NORM) -', NORM
    WRITE (NOUT,99998)
+   'Distribution of vectors (DISTR)    -',
+   DISTR
    WRITE (NOUT,99997)
+   'Termination criterion (ITERM)     -',
+   ITERM
    WRITE (NOUT,99997)
+   'Order of the system of equations (N) -', N

```

```

WRITE (NOUT,99996)
+ 'Tolerance (TOL)                                -', TOL
WRITE (NOUT,99997)
+ 'Maximum number of iterations allowed (MAXITN)   -',
+ MAXITN
WRITE (NOUT,99997)
+ 'Dimension of RGMRES orthogonal basis (M)        -', M
WRITE (NOUT,99997)
+ 'Monitoring frequency (MONIT)                  -',
+ MONIT
*
*   End of subroutine PRINTI
*
RETURN
*
*
99999 FORMAT (/1X,'Summary of input parameters and options',/1X,39('-'),
+           /)
99998 FORMAT (1X,A,4X,A)
99997 FORMAT (1X,A,I5)
99996 FORMAT (1X,A,3X,1P,D9.2)
END
*
*****
*****
*
SUBROUTINE GMAT(I1,I2,N,NNZL,AL,LAL,IROWL,ICOLL)
*
*
*   .. Scalar Arguments ..
INTEGER      I1, I2, LAL, N, NNZL
*
*   .. Array Arguments ..
DOUBLE PRECISION AL(LAL)
INTEGER      ICOLL(LAL), IROWL(LAL)
*
*   .. Local Scalars ..
DOUBLE PRECISION H, H2, H24, HM1, HP1
INTEGER      I, INZL, IX, IY, J, NX
LOGICAL      BOUNDX, BOUNDY
*
*   .. Intrinsic Functions ..
INTRINSIC    DBLE, DSQRT, MOD, NINT
*
*   .. Executable Statements ..
NNZL = 0
NX = NINT(DSQRT(DBLE(N)))
H = 1/DBLE(NX+1)
H2 = DBLE((NX+1)**2)
H24 = 4*H2
HM1 = H2*(H-1.D+0)
HP1 = -H2*(H+1.D+0)
DO 40 I = I1, I2
*
*       INZL = NNZL
*
*   Calculate number of non-zero elements in I-th row
*
*       IX = 1 + MOD(I-1,NX)
*       IY = 1 + (I-1)/NX
*       BOUNDX = (IX.EQ.1) .OR. (IX.EQ.NX)
*       BOUNDY = (IY.EQ.1) .OR. (IY.EQ.NX)

```

```

      IF (BOUNDX .AND. BOUNDY) THEN
        NNZL = NNZL + 3
      ELSE IF ( .NOT. (BOUNDX .OR. BOUNDY)) THEN
        NNZL = NNZL + 5
      ELSE
        NNZL = NNZL + 4
      END IF

*
*   Check whether there is sufficient storage space
*
      IF (NNZL.GT.LAL) GO TO 40

*
*   Set non-zero elements in I-th row
*
      DO 20 J = INZL + 1, NNZL
        IROWL(J) = I
20    CONTINUE

*
      INZL = INZL + 1
      ICOLL(INZL) = I
      AL(INZL) = H24

*
      IF (IY.GT.1) THEN
        INZL = INZL + 1
        ICOLL(INZL) = I - NX
        AL(INZL) = HM1
      END IF

*
      IF (IX.GT.1) THEN
        INZL = INZL + 1
        ICOLL(INZL) = I - 1
        AL(INZL) = HM1
      END IF

*
      IF (IX.LT.NX) THEN
        INZL = INZL + 1
        ICOLL(INZL) = I + 1
        AL(INZL) = HP1
      END IF

*
      IF (IY.LT.NX) THEN
        INZL = INZL + 1
        ICOLL(INZL) = I + NX
        AL(INZL) = HP1
      END IF

*
      NNZL = INZL

*
40  CONTINUE

*
*   End of subroutine GMAT
*
      RETURN
      END

*
*****
*****
*****
*

```

```

SUBROUTINE GVEC(I1,I2,X)
*
*
* .. Scalar Arguments ..
INTEGER      I1, I2
* .. Array Arguments ..
DOUBLE PRECISION X(*)
* .. Local Scalars ..
INTEGER      I
* .. Executable Statements ..
DO 20 I = I1, I2
    X(I-I1+1) = 1.D+2
20 CONTINUE
*
* End of subroutine GVEC
*
RETURN
END

```

## 8.2 Example Data

```

F11BAFP Example Program Data
  2  2      : MP, NP
  8      : NX
'RGMRES'   : METHOD
'P' '2'  2  0 : PRECON, NORM, ITERM, MONIT
 10 16     : M, MB
 1.0D-06 100 : TOL, MAXITN
'(8F8.4)'  : FORMAT

```

## 8.3 Example Results

F11BAFP Example Program Results

Summary of input parameters and options

-----

Method used (METHOD)	-	RGMRES
Use the preconditioner (PRECON)	-	P
Matrix and vector norm in use (NORM)	-	2
Distribution of vectors (DISTR)	-	A
Termination criterion (ITERM)	-	2
Order of the system of equations (N)	-	64
Tolerance (TOL)	-	1.00D-06
Maximum number of iterations allowed (MAXITN)	-	100
Dimension of RGMRES orthogonal basis (M)	-	10
Monitoring frequency (MONIT)	-	0

Summary of results

-----

Number of iterations carried out	-	13
Left-hand side of the termination criterion	-	2.46D-05
Right-hand side of the termination criterion	-	9.97D-05
1-norm of the matrix of the coefficients	-	0.000
Largest singular value of the preconditioned matrix	-	2.584

Solution vector

-----

2.0231	3.0860	3.5805	3.6951	3.5227	3.1002	2.4225	1.4364
3.0860	4.7996	5.6149	5.8043	5.5143	4.8090	3.6933	2.1219
3.5805	5.6149	6.5941	6.8219	6.4696	5.6165	4.2777	2.4241
3.6951	5.8042	6.8219	7.0595	6.6941	5.8076	4.4167	2.4959
3.5227	5.5142	6.4696	6.6941	6.3566	5.5311	4.2250	2.4014
3.1002	4.8090	5.6165	5.8076	5.5311	4.8443	3.7394	2.1571
2.4225	3.6933	4.2777	4.4167	4.2250	3.7394	2.9382	1.7417
1.4364	2.1219	2.4241	2.4959	2.4014	2.1571	1.7417	1.0827

Residual vector

-----

0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000
0.0010	0.0006	0.0007	-0.0004	-0.0006	-0.0002	-0.0014	-0.0002
0.0010	0.0002	-0.0004	-0.0016	-0.0014	-0.0003	-0.0006	0.0005
0.0004	0.0009	0.0014	0.0004	0.0009	0.0017	-0.0002	-0.0006
0.0012	0.0011	0.0007	-0.0008	-0.0006	0.0007	0.0001	0.0004
0.0005	0.0001	0.0006	0.0001	0.0003	0.0004	-0.0012	-0.0003
0.0010	-0.0001	0.0001	-0.0002	0.0000	0.0001	-0.0008	0.0008
0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000

---