F11GAFP

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 2.3 of the F11 Chapter Introduction.

F11GAFP is the first in a suite of three routines for the iterative solution of a real symmetric system of simultaneous linear equations $Ax = b$ using either of two methods:

Conjugate Gradient Method. For this method (Hestenes and Stiefel [4], Golub and Van Loan [3], Barrett *et al.* [1], Dias da Cunha and Hopkins [2]), the matrix A should ideally be positive-definite. The application of the Conjugate Gradient method to indefinite matrices may lead to failure or to lack of convergence.

Lanczos Method (SYMMLQ). This method, based upon the algorithm SYMMLQ (Paige and Saunders [6], Barrett *et al.* [1]), is suitable for both positive-definite and indefinite matrices. It is more robust than the Conjugate Gradient method but less efficient when A is positive-definite.

F11GAFP is a set-up routine which must be called before F11GBFP, the iterative solver. The third routine in the suite, F11GCFP can be used to return additional information about the computation. These three routines are suitable for the solution of large sparse symmetric systems of equations.

2 Specification

```
      SUBROUTINE F11GAFP(ICNTXT, METHOD, PRECON, SIGCMP, NORM, DISTR,
     1                   WEIGHT, ITERM, N, NLOC,TOL, MAXITN, ANORM,
     2                   SIGMAX, SIGTOL, MAXITS, MONIT, LWREQ,IFAIL)
      DOUBLE PRECISION   TOL, ANORM, SIGMAX, SIGTOL
      INTEGER            ICNTXT, ITERM, N, NLOC, MAXITN, MAXITS, MONIT,
     1                   LWREQ, IFAIL
      CHARACTER*1        PRECON, SIGCMP, 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.

3.2 Global and Local Arguments

The input arguments METHOD, PRECON, SIGCMP, NORM, DISTR, WEIGHT, ITERM, N, TOL, MAXITN, ANORM, SIGMAX, SIGTOL, MAXITS, 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 input 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 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 possible choices are:

'CG' Conjugate Gradient method;
'SYMMLQ' Lanczos method (SYMMLQ).

Constraint: METHOD = 'CG' or 'SYMMLQ'.

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: SIGCMP — CHARACTER*1 *Global Input*

On entry: determines whether an estimate of $\sigma_1(\bar{A}) = \|E^{-1}AE^{-T}\|_2$, the largest singular value of the preconditioned matrix is to be computed using the bisection method on the sequence of symmetric tridiagonal matrices $\{T_k\}$ generated during the iteration. Note that $\bar{A} = A$ if a preconditioner is not used.
If SIGMAX > 0.0 (see below), then SIGCMP is not referenced, i.e., when $\sigma_1(\bar{A})$ is supplied in input. The possible choices are:

'S' $\sigma_1(\bar{A})$ is to be computed using the bisection method.
'N' The bisection method is not used.
If the termination criterion requires $\sigma_1(\bar{A})$ then a less expensive estimate is computed.

See Section 6 for further details.

Suggested value: SIGCMP = 'N'.

Constraint: SIGCMP = 'S' or 'N'.

5: 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_∞-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'.

6: 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 F11GAFP is used in connection with any F11 routine other than F11GBFP and F11GCFP.

Suggested value: DISTR = 'A'.

Constraint: DISTR = 'A', 'C' or 'R'.

7: 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 $i = 1,\ldots,n$. 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 are to be used and must be supplied on initial entry to F11GBFP;
'N' all weights are implicitly set equal to one. Weights do not need to be supplied on initial entry to F11GBFP.

See also Section 4 of the document for F11GBFP.

Suggested value: WEIGHT = 'N'.

Constraint: if ITERM = 1, then WEIGHT = 'W' or 'N'; if ITERM = 2, then WEIGHT = 'N'.

8: ITERM — INTEGER *Global Input*

On entry: defines the termination criterion to be used. The possible choices are:

1 $\|r_k\|_p \leq \tau\,(\|b\|_p + \|A\|_p\,\|x_k\|_p)$, where $p = 1, \infty$ or 2;
2 $\|M^{-1}r_k\|_2 \leq \tau\,\sigma_1(\bar{A})\,\|x_k\|_2$ for the Conjugate Gradient method;
$\|\bar{r}_k\|_2 \leq \tau\,\max(1.0, \|b\|_2/\|r_0\|_2)\,(\|\bar{r}_0\|_2 + \sigma_1(\bar{A})\,\|\bar{x}_k - \bar{x}_{k-1}\|_2)$ for the Lanczos method (SYMMLQ).

Suggested value: ITERM = 1.

Constraint: ITERM = 1 or 2.

9: N — INTEGER *Global Input*

On entry: the order n of the matrix A.

Constraint: N $\geq$ 0.

10: 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:

DISTR = 'A': $\displaystyle\sum_{i=0}^{m_p-1}\sum_{j=0}^{n_p-1} n_l(i,j) = n$;

DISTR = 'C': $\displaystyle\sum_{i=0}^{m_p-1} n_l(i,j) = n$, for $j = 0,\ldots,n_p-1$ and $n_l(i,0) = n_l(i,1) = \ldots = n_l(i,n_p-1)$ for $i = 0,\ldots,m_p-1$;

DISTR = 'R': $\sum_{j=0}^{n_p-1} n_l(i,j) = n$, for $i = 0,\ldots,m_p-1$ and $n_l(0,j) = n_l(1,j) = \ldots = n_l(m_p-1,j)$ for $j = 0,\ldots,n_p-1$.

11: TOL — DOUBLE PRECISION *Global Input*

On entry: the tolerance τ for the termination criterion. If TOL ≤ 0.0, the default value TOL $= \sqrt{\epsilon}$ is used, where ϵ is the ***machine precision***. Otherwise, the actual value used is max(TOL,ϵ).

Constraint: TOL < 1.0.

12: MAXITN — INTEGER *Global Input*

On entry: the maximum number of iterations.

Constraint: MAXITN ≥ 0.

13: ANORM — DOUBLE PRECISION *Global Input*

On entry: the value of $\|A\|$ to be used in the termination criterion when ITERM = 1.
If ANORM ≤ 0.0, ITERM = 1 and NORM = '1' or 'I', then $\|A\|_1 = \|A\|_\infty$ is estimated internally by F11GBFP.
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 $\sigma_1(\bar{A}) = \|E^{-1}AE^{-T}\|_2$.
Note that $\sigma_1(\bar{A})$ is used not only when explicitly required by the termination criterion, i.e., when ITERM = 2, but also when explicitly required by SIGCMP = 'S' or when the Lanczos method (SYMMLQ) is employed and user-specified weights are not supplied (see the parameter WEIGHT above).
If SIGMAX ≤ 0.0 and $\sigma_1(\bar{A})$ is required, then it is estimated internally by F11GBFP using the method determined by the parameter SIGCMP.
If $\sigma_1(\bar{A})$ is not used, then SIGMAX is not referenced.
See Section 6 for further details.

15: SIGTOL — DOUBLE PRECISION *Global Input*

On entry: the tolerance used in assessing the convergence of the estimate of $\sigma_1(\bar{A}) = \|\bar{A}\|_2$ when the bisection method is used. If SIGTOL ≤ 0.0, the default value SIGTOL $= 10^{-2}$ is used. The actual value used is max(SIGTOL,ϵ), where ϵ is the ***machine precision***.
If SIGCMP = 'N' or SIGMAX > 0.0, then SIGTOL is not referenced.
See also Section 6.

Suggested value: SIGTOL $= 10^{-2}$ should be sufficient in most cases.

Constraint: if SIGCMP = 'S' and SIGMAX ≤ 0.0, then SIGTOL < 1.0.

16: MAXITS — INTEGER *Global Input*

On entry: the maximum iteration number k = MAXITS for which $\sigma_1(T_k)$ is computed by bisection.
If SIGCMP = 'N' or SIGMAX > 0.0, then MAXITS is not referenced.
See also Section 6.

Suggested value: MAXITS $= \min(10, n)$.

Constraint: if SIGCMP = 'S' and SIGMAX ≤ 0.0, then $1 \leq$ MAXITS $\leq$ MAXITN.

17: MONIT — INTEGER *Global Input*

On entry: the frequency at which the monitoring step is executed by F11GBFP.
If MONIT > 0, then F11GBFP will monitor the solution every MONIT iterations, starting from the MONITth iteration. If MONIT ≤ 0, no monitoring takes place.
There are some additional computational costs involved in monitoring the solution and residual vector when the Lanczos method (SYMMLQ) is used.

Constraint: MONIT $\leq$ MAXITN.

18: LWREQ — INTEGER *Local Output*

On exit: the amount of workspace required by F11GBFP.
See also Section 4 of the document for F11GBFP.

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

IFAIL = 1

F11GAFP has been called out of sequence. Either F11GAFP has been called twice without calling F11GBFP in between, or F11GBFP has not completed its current task.

6 Further Comments

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

ITERM = 1 if NORM = '1' or 'I', any values of SIGCMP, ANORM and SIGMAX are allowed;
if NORM = '2', any values of SIGCMP and SIGMAX are allowed but $\|A\|_2$ must be supplied on entry, i.e., ANORM > 0.0.

ITERM = 2 NORM = '2' and any values of SIGCMP, ANORM and SIGMAX are allowed.

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

When $\sigma_1(\bar{A})$ is not supplied (SIGMAX $\leq$ 0.0) but it is required, it is estimated by F11GBFP using either of two methods, as specified by the parameter SIGCMP.

SIGCMP = 'S' $\sigma_1(\bar{A}) \approx \max_k \sigma_1(T_k)$, where the sequence of symmetric tridiagonal matrices $\{T_k\}$ is generated in the course of the iteration (see alsoSections 2 and 2.3 of the F11 Chapter Introduction). The interlacing property $\sigma_1(T_{k-1}) \leq \sigma_1(T_k)$ and Gerschgorin's theorem provide lower and upper bounds from which $\sigma_1(T_k)$ can be easily estimated by bisection.
When the differences between three successive values of $\sigma_1(T_k)$ differ, in a relative sense, by less

than the tolerance SIGTOL the computation is deemed to have converged, i.e., when $\max\left(\frac{|\sigma_1(T_k)-\sigma_1(T_{k-1})|}{\sigma_1(T_k)}, \frac{|\sigma_1(T_k)-\sigma_1(T_{k-2})|}{\sigma_1(T_k)}\right) \leq$ SIGTOL. The computation of $\sigma_1(\bar{A})$ is also terminated when the iteration count exceeds the maximum value allowed, i.e., $k \geq$ MAXITS.

Bisection is increasingly expensive with increasing iteration count. A reasonably large value of SIGTOL, of the order of the suggested value, is recommended and an excessive value of MAXITS should be avoided. Under these conditions, $\sigma_1(\bar{A})$ usually converges within very few iterations.

SIGCMP = 'N' the less expensive estimate $\sigma_1(\bar{A}) \approx \max_k \|T_k\|_1$ is used, where $\sigma_1(\bar{A}) \leq \|T_k\|_1$ by Gerschgorin's theorem.

If ITERM = 1, then $\sigma_1(\bar{A})$ is estimated only if SIGCMP = 'S' or if the Lanczos method (SYMMLQ) is used without user-specified weights. 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 F11GAFP are negligible compared to the costs of F11GBFP.

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] Golub G H and Van Loan C F (1989) *Matrix Computations* Johns Hopkins University Press (2nd Edition), Baltimore

[4] Hestenes M and Stiefel E (1952) Methods of conjugate gradients for solving linear systems *J. Res. Nat. Bur. Stand.* **49** 409–436

[5] Higham N J (1988) FORTRAN codes for estimating the one-norm of a real or complex matrix, with applications to condition estimation *ACM Trans. Math. Software* **14** 381–396

[6] Paige C C and Saunders M A (1975) Solution of sparse indefinite systems of linear equations *SIAM J. Numer. Anal.* **12** 617–629

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 matrix A is given by the block matrix

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

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

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

and

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

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

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

```
*     F11GAFP 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, SIGERR, SIGMAX, SIGTOL, STPLHS, STPRHS,
     +                 TOL
      INTEGER          I, ICNTXT, IFAIL, IREVCM, ITERM, ITN, ITS, IW, K,
     +                 LW, LWREQ, MAXITN, MAXITS, 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, SIGCMP,
     +                 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, F11DAFP,
     +                 F11DBFP, F11GAFP, F11GBFP, F11GCFP, F11XAFP,
     +                 F11XBFP, F11ZAFP, GMAT, GVEC, PRINTI, X04YAFP,
     +                 Z01AAFP, Z01ABFP
*     .. Executable Statements ..
      ROOT = Z01ACFP()
      IF (ROOT) THEN
         WRITE (NOUT,*) 'F11GAFP Example Program Results'
      END IF
*
*     Open the input file on all processors
*
      OPEN (NIN,FILE='f11gafpe.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.D0
      SIGMAX = 0.D0
      CHECK = 'N'
      ITN = 0
*
*     Read the problem parameters
*
      READ (NIN,*) NX
      N = NX**2
*
*     Read the algorithmic parameters
*
      READ (NIN,*) METHOD
```

```
      READ (NIN,*) PRECON, SIGCMP, NORM, ITERM, MONIT
      READ (NIN,*) MB
      READ (NIN,*) TOL, MAXITN
      READ (NIN,*) SIGTOL, MAXITS
      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=99995)
         GO TO 80
      ELSE IF (IERR(2).NE.0) THEN
         IF (ROOT) WRITE (NOUT,FMT=99994)
         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 F11GAFP(ICNTXT,METHOD,PRECON,SIGCMP,NORM,DISTR,WEIGHT,ITERM,
     +             N,ML,TOL,MAXITN,ANORM,SIGMAX,SIGTOL,MAXITS,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=99993)
         GO TO 80
      END IF
*
*     Print a summary of input parameters and options
*
      IF (ROOT) CALL PRINTI(NOUT,METHOD,PRECON,SIGCMP,NORM,DISTR,ITERM,
     +                      N,MAXITN,TOL,SIGTOL,MONIT)
*
*     Set initial approximation to solution
*
      DO 40 I = 1, ML
         U(I) = 0.D0
   40 CONTINUE
*
*     Solve the equations
*
      LW = LWREQ
      IW = LWREQ + 1
      IREVCM = 0
*
      LOOP = .TRUE.
   60 CONTINUE

      CALL F11GBFP(ICNTXT,IREVCM,U,V,WORK,LW,IFAIL)
*
      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 + MONIT
            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 F11GCFP(ICNTXT,ITN,STPLHS,STPRHS,ANORM,SIGMAX,ITS,SIGERR,
     +                IFAIL)
*
*        Produce a report
*
         IF (ROOT) THEN
            WRITE (NOUT,'(/1X,''Summary of results''/1X,18(''-'')/)')
            WRITE (NOUT,99998)
     +        'Number of iterations carried out                    -', ITN
            WRITE (NOUT,99996)
     +        'Left-hand side of the termination criterion         -',
     +        STPLHS
            WRITE (NOUT,99996)
     +        'Right-hand side of the termination criterion        -',
     +        STPRHS
            WRITE (NOUT,99997)
     +        '1-norm of the matrix of the coefficients            -',
     +        ANORM
            WRITE (NOUT,99998)
     +        'Number of iterations used to compute SIGMAX         -', ITS
            WRITE (NOUT,99997)
     +        'Largest singular value of the preconditioned matrix  -',
     +        SIGMAX
            WRITE (NOUT,99997)
     +        'Relative error in the largest singular value        -',
     +        SIGERR
```

```
            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,4X,A)
99998 FORMAT (1X,A,I5)
99997 FORMAT (1X,A,2X,F7.3)
99996 FORMAT (1X,A,3X,1P,D9.2)
99995 FORMAT (1X,'** ERROR: Number of rows per processor too large')
99994 FORMAT (1X,'** ERROR: Number of row blocks per processor too ',
     +       'large')
99993 FORMAT (1X,'** ERROR: LWORK too small')
      END
*
***********************************************************************
***********************************************************************
*
      SUBROUTINE PRINTI(NOUT,METHOD,PRECON,SIGCMP,NORM,DISTR,ITERM,N,
     +                  MAXITN,TOL,SIGTOL,MONIT)
*
*     Prints a summary of the input parameters and options.
*
*
*     .. Scalar Arguments ..
      DOUBLE PRECISION  SIGTOL, TOL
      INTEGER           ITERM, MAXITN, MONIT, N, NOUT
      CHARACTER         DISTR, NORM, PRECON, SIGCMP
      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)
     +  'Use bisection for the largest singular value (SIGCMP)  -',
     +  SIGCMP
      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,99996)
     +  'Tolerance for the largest singular value (SIGTOL)     -',
     +  SIGTOL
      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 H2, H24
      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)))
      H2 = DBLE((NX+1)**2)
      H24 = 4*H2
      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) = -H2
         END IF
*
         IF (IX.GT.1) THEN
            INZL = INZL + 1
            ICOLL(INZL) = I - 1
            AL(INZL) = -H2
         END IF
*
         IF (IX.LT.NX) THEN
            INZL = INZL + 1
            ICOLL(INZL) = I + 1
            AL(INZL) = -H2
         END IF
*
         IF (IY.LT.NX) THEN
            INZL = INZL + 1
            ICOLL(INZL) = I + NX
            AL(INZL) = -H2
         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

```
F11GAFP Example Program Data
    2   2                           :  MP, NP
    8                               :  NX
  'CG'                              :  METHOD
  'P'  'S' '2'    2    0            :  PRECON, SIGCMP, NORM, ITERM, MONIT
   16                               :  MB
   1.0D-06   50                     :  TOL, MAXITN
   1.0D-02    6                     :  SIGTOL, MAXITS
   '(8F8.4)'                        :  FORMAT
```

8.3 Example Results

```
 F11GAFP Example Program Results

 Summary of input parameters and options
 ---------------------------------------

 Method used (METHOD)                                  -    CG
 Use the preconditioner (PRECON)                       -    P
 Use bisection for the largest singular value (SIGCMP) -    S
 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)         -   50
 Tolerance for the largest singular value (SIGTOL)     -    1.00D-02
 Monitoring frequency (MONIT)                          -    0

 Summary of results
 ------------------

 Number of iterations carried out                    -    9
 Left-hand side of the termination criterion         -    1.84D-06
 Right-hand side of the termination criterion        -    5.76D-05
 1-norm of the matrix of the coefficients            -    0.000
```

```
Number of iterations used to compute SIGMAX         -    5
Largest singular value of the preconditioned matrix  -    1.568
Relative error in the largest singular value         -    0.121

Solution vector
---------------

 1.4983  2.3794  2.8836  3.1152  3.1152  2.8836  2.3794  1.4983
 2.3794  3.9010  4.8054  5.2274  5.2274  4.8054  3.9010  2.3794
 2.8836  4.8054  5.9748  6.5270  6.5270  5.9748  4.8054  2.8836
 3.1152  5.2274  6.5270  7.1443  7.1443  6.5270  5.2274  3.1152
 3.1152  5.2274  6.5270  7.1443  7.1443  6.5270  5.2274  3.1152
 2.8836  4.8054  5.9748  6.5270  6.5270  5.9748  4.8054  2.8836
 2.3794  3.9010  4.8054  5.2274  5.2274  4.8054  3.9010  2.3794
 1.4983  2.3794  2.8836  3.1152  3.1152  2.8836  2.3794  1.4983

Residual vector
---------------

 0.0000  0.0000  0.0000  0.0000  0.0000  0.0000  0.0000  0.0000
 0.0000 -0.0001  0.0002 -0.0001 -0.0001  0.0002 -0.0001  0.0000
-0.0001  0.0002 -0.0002  0.0001  0.0001 -0.0002  0.0002 -0.0001
 0.0001 -0.0002  0.0001 -0.0001 -0.0001  0.0001 -0.0002  0.0001
 0.0001 -0.0002  0.0001 -0.0001 -0.0001  0.0001 -0.0002  0.0001
-0.0001  0.0002 -0.0002  0.0001  0.0001 -0.0002  0.0002 -0.0001
 0.0000 -0.0001  0.0002 -0.0001 -0.0001  0.0002 -0.0001  0.0000
 0.0000  0.0000  0.0000  0.0000  0.0000  0.0000  0.0000  0.0000
```