NAG Library Routine Document

F11GRF

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

F11GRF is a setup routine, the first in a suite of three routines for the iterative solution of a complex Hermitian system of simultaneous linear equations. F11GRF must be called before F11GSF, the iterative solver. The third routine in the suite, F11GTF, can be used to return additional information about the computation.

These three routines are suitable for the solution of large sparse complex Hermitian systems of equations.

2 Specification

```
SUBROUTINE F11GRF (METHOD, PRECON, SIGCMP, NORM, WEIGHT, ITERM, N, TOL,     &
                   MAXITN, ANORM, SIGMAX, SIGTOL, MAXITS, MONIT, LWREQ,     &
                   WORK, LWORK, IFAIL)

INTEGER                 ITERM, N, MAXITN, MAXITS, MONIT, LWREQ, LWORK,       &
                        IFAIL
REAL (KIND=nag_wp)      TOL, ANORM, SIGMAX, SIGTOL
COMPLEX (KIND=nag_wp)   WORK(LWORK)
CHARACTER(*)            METHOD
CHARACTER(1)            PRECON, SIGCMP, NORM, WEIGHT
```

3 Description

The suite consisting of the routines F11GRF, F11GSF and F11GTF is designed to solve the complex Hermitian system of simultaneous linear equations $Ax = b$ of order n, where n is large and the matrix of the coefficients A is sparse.

F11GRF is a setup routine which must be called before the iterative solver F11GSF. F11GTF, the third routine in the suite, can be used to return additional information about the computation. Either of two methods can be used:

1. **Conjugate Gradient Method (CG)**

   For this method (see Hestenes and Stiefel (1952), Golub and Van Loan (1996), Barrett *et al.* (1994) and Dias da Cunha and Hopkins (1994)), 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.

2. **Lanczos Method (SYMMLQ)**

   This method, based upon the algorithm SYMMLQ (see Paige and Saunders (1975) and Barrett *et al.* (1994)), 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.

Both CG and SYMMLQ methods start from the residual $r_0 = b - Ax_0$, where x_0 is an initial estimate for the solution (often $x_0 = 0$), and generate an orthogonal basis for the Krylov subspace span$\{A^k r_0\}$, for $k = 0, 1, \ldots$, by means of three-term recurrence relations (see Golub and Van Loan (1996)). A sequence of real symmetric tridiagonal matrices $\{T_k\}$ is also generated. Here and in the following, the index k denotes the iteration count. The resulting real symmetric tridiagonal systems of equations are usually more easily solved than the original problem. A sequence of solution iterates $\{x_k\}$ is thus generated such that the sequence of the norms of the residuals $\{\|r_k\|\}$ converges to a required tolerance. Note that, in general, the convergence is not monotonic.

In exact arithmetic, after n iterations, this process is equivalent to an orthogonal reduction of A to real symmetric tridiagonal form, $T_n = Q^{\mathrm{H}}AQ$; the solution x_n would thus achieve exact convergence. In finite-precision arithmetic, cancellation and round-off errors accumulate causing loss of orthogonality. These methods must therefore be viewed as genuinely iterative methods, able to converge to a solution **within a prescribed tolerance**.

The orthogonal basis is not formed explicitly in either method. The basic difference between the two methods lies in the method of solution of the resulting real symmetric tridiagonal systems of equations: the conjugate gradient method is equivalent to carrying out an LDL^{H} (Cholesky) factorization whereas the Lanczos method (SYMMLQ) uses an LQ factorization.

Faster convergence can be achieved using a **preconditioner** (see Golub and Van Loan (1996) and Barrett *et al.* (1994)). A preconditioner maps the original system of equations onto a different system, say

$$\bar{A}\bar{x} = \bar{b}, \tag{1}$$

with, hopefully, better characteristics with respect to its speed of convergence: for example, the condition number of the matrix of the coefficients can be improved or eigenvalues in its spectrum can be made to coalesce. An orthogonal basis for the Krylov subspace span$\{\bar{A}^k\bar{r}_0\}$, for $k = 0, 1, \ldots$, is generated and the solution proceeds as outlined above. The algorithms used are such that the solution and residual iterates of the original system are produced, not their preconditioned counterparts. Note that an unsuitable preconditioner or no preconditioning at all may result in a very slow rate, or lack, of convergence. However, preconditioning involves a trade-off between the reduction in the number of iterations required for convergence and the additional computational costs per iteration. Also, setting up a preconditioner may involve non-negligible overheads.

A preconditioner must be **Hermitian and positive definite**, i.e., representable by $M = EE^{\mathrm{H}}$, where M is nonsingular, and such that $\bar{A} = E^{-1}AE^{-\mathrm{H}} \sim I_n$ in (1), where I_n is the identity matrix of order n. Also, we can define $\bar{r} = E^{-1}r$ and $\bar{x} = E^{\mathrm{H}}x$. These are formal definitions, used only in the design of the algorithms; in practice, only the means to compute the matrix-vector products $v = Au$ and to solve the preconditioning equations $Mv = u$ are required, that is, explicit information about M, E or their inverses is not required at any stage.

The first termination criterion

$$\|r_k\|_p \leq \tau\Big(\|b\|_p + \|A\|_p \times \|x_k\|_p\Big) \tag{2}$$

is available for both conjugate gradient and Lanczos (SYMMLQ) methods. In (2), $p = 1, \infty$ or 2 and τ denotes a user-specified tolerance subject to $\max(10, \sqrt{n})\epsilon \leq \tau < 1$, where ϵ is the ***machine precision***. Facilities are provided for the estimation of the norm of the matrix of the coefficients $\|A\|_1 = \|A\|_\infty$, when this is not known in advance, used in (2), by applying Higham's method (see Higham (1988)). Note that $\|A\|_2$ cannot be estimated internally. This criterion uses an error bound derived from **backward** error analysis to ensure that the computed solution is the exact solution of a problem as close to the original as the termination tolerance requires. Termination criteria employing bounds derived from **forward** error analysis could be used, but any such criteria would require information about the condition number $\kappa(A)$ which is not easily obtainable.

The second termination criterion

$$\|\bar{r}_k\|_2 \leq \tau \max\big(1.0, \|b\|_2/\|r_0\|_2\big)\big(\|\bar{r}_0\|_2 + \sigma_1\big(\bar{A}\big) \times \|\Delta\bar{x}_k\|_2\big) \tag{3}$$

is available only for the Lanczos method (SYMMLQ). In (3), $\sigma_1\big(\bar{A}\big) = \big\|\bar{A}\big\|_2$ is the largest singular value of the (preconditioned) iteration matrix $\bar{A}$. This termination criterion monitors the progress of the solution of the preconditioned system of equations and is less expensive to apply than criterion (2). When $\sigma_1\big(\bar{A}\big)$ is not supplied, facilities are provided for its estimation by $\sigma_1\big(\bar{A}\big) \sim \max_k \sigma_1(T_k)$. 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 computed by bisection. Alternatively, the less expensive estimate $\sigma_1\big(\bar{A}\big) \sim \max_k \|T_k\|_1$ can be used, where $\sigma_1\big(\bar{A}\big) \leq \|T_k\|_1$ by Gerschgorin's theorem. Note that only order of magnitude estimates are required by the termination criterion.

Termination criterion (2) is the recommended choice, despite its (small) additional costs per iteration when using the Lanczos method (SYMMLQ). Also, if the norm of the initial estimate is much larger than the norm of the solution, that is, if $\|x_0\| \gg \|x\|$, a dramatic loss of significant digits could result in complete lack of convergence. The use of criterion (2) will enable the detection of such a situation, and the iteration will be restarted at a suitable point. No such restart facilities are provided for criterion (3).

Optionally, a vector w of user-specified weights can be used in the computation of the vector norms in termination criterion (2), i.e., $\|v\|_p^{(w)} = \left\|v^{(w)}\right\|_p$, where $\left(v^{(w)}\right)_i = w_i v_i$, for $i = 1, 2, \ldots, n$. Note that the use of weights increases the computational costs.

The sequence of calls to the routines comprising the suite is enforced: first, the setup routine F11GRF must be called, followed by the solver F11GSF. F11GTF can be called either when F11GSF is carrying out a monitoring step or after F11GSF has completed its tasks. Incorrect sequencing will raise an error condition.

4 References

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

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, UK

Golub G H and Van Loan C F (1996) *Matrix Computations* (3rd Edition) Johns Hopkins University Press, Baltimore

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

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

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

5 Parameters

1: METHOD – CHARACTER(*) *Input*

On entry: the iterative method to be used.

METHOD = 'CG'
Conjugate gradient method.

METHOD = 'SYMMLQ'
Lanczos method (SYMMLQ).

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

2: PRECON – CHARACTER(1) *Input*

On entry: determines whether preconditioning is used.

PRECON = 'N'
No preconditioning.

PRECON = 'P'
Preconditioning.

Constraint: PRECON = 'N' or 'P'.

3: SIGCMP – CHARACTER(1) *Input*

On entry: determines whether an estimate of $\sigma_1(\bar{A}) = \left\|E^{-1}AE^{-\mathrm{H}}\right\|_2$, the largest singular value of the preconditioned matrix of the coefficients, is to be computed using the bisection method on the sequence of tridiagonal matrices $\{T_k\}$ generated during the iteration. Note that $\bar{A} = A$ when a preconditioner is not used.

If SIGMAX > 0.0 (see below), i.e., when $\sigma_1(\bar{A})$ is supplied, the value of SIGCMP is ignored.

SIGCMP = 'S'
$\sigma_1(\bar{A})$ is to be computed using the bisection method.

SIGCMP = 'N'
The bisection method is not used.

If the termination criterion (3) is used, requiring $\sigma_1(\bar{A})$, an inexpensive estimate is computed and used (see Section 3).

Suggested value: SIGCMP = 'N'.

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

4: NORM – CHARACTER(1) *Input*

On entry: defines the matrix and vector norm to be used in the termination criteria.

NORM = '1'
Use the l_1 norm.

NORM = 'I'
Use the l_∞ norm.

NORM = '2'
Use the l_2 norm.

Suggested value:

if ITERM = 1, NORM = 'I';
if ITERM = 2, NORM = '2'.

Constraints:

if ITERM = 1, NORM = '1', 'I' or '2';
if ITERM = 2, NORM = '2'.

5: WEIGHT – CHARACTER(1) *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 criterion (2) (ITERM = 1): $\|v\|_p^{(w)} = \left\|v^{(w)}\right\|_p$, where $v_i^{(w)} = w_i v_i$, for $i = 1, 2, \ldots, n$. 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, i.e., when criterion (3) is used.

WEIGHT = 'W'
User-supplied weights are to be used and must be supplied on initial entry to F11GSF.

WEIGHT = 'N'
All weights are implicitly set equal to one. Weights do not need to be supplied on initial entry to F11GSF.

Suggested value: WEIGHT = 'N'.

Constraints:

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

6: ITERM – INTEGER *Input*

On entry: defines the termination criterion to be used.

ITERM = 1
: Use the termination criterion defined in (2) (both conjugate gradient and Lanczos (SYMMLQ) methods).

ITERM = 2
: Use the termination criterion defined in (3) (Lanczos method (SYMMLQ) only).

Suggested value: ITERM = 1.

Constraints:

if METHOD = 'CG', ITERM = 1;
if METHOD = 'SYMMLQ', ITERM = 1 or 2.

7: N – INTEGER *Input*

On entry: n, the order of the matrix A.

Constraint: N > 0.

8: TOL – REAL (KIND=nag_wp) *Input*

On entry: the tolerance τ for the termination criterion.

If TOL ≤ 0.0, $\tau = \max(\sqrt{\epsilon}, \sqrt{n}\epsilon)$ is used, where ϵ is the ***machine precision***.

Otherwise $\tau = \max(\text{TOL}, 10\epsilon, \sqrt{n}\epsilon)$ is used.

Constraint: TOL < 1.0.

9: MAXITN – INTEGER *Input*

On entry: the maximum number of iterations.

Constraint: MAXITN > 0.

10: ANORM – REAL (KIND=nag_wp) *Input*

On entry: if ANORM > 0.0, the value of $\|A\|_p$ to be used in the termination criterion (2) (ITERM = 1).

If ANORM ≤ 0.0, ITERM = 1 and NORM = '1' or 'I', then $\|A\|_1 = \|A\|_\infty$ is estimated internally by F11GSF.

If ITERM = 2, then ANORM is not referenced.

Constraint: if ITERM = 1 and NORM = 2, ANORM > 0.0.

11: SIGMAX – REAL (KIND=nag_wp) *Input*

On entry: if SIGMAX > 0.0, the value of $\sigma_1(\bar{A}) = \left\|E^{-1}AE^{-\mathrm{H}}\right\|_2$.

If SIGMAX ≤ 0.0, $\sigma_1(\bar{A})$ is estimated by F11GSF when either SIGCMP = 'S' or termination criterion (3) (ITERM = 2) is employed, though it will be used only in the latter case.

Otherwise, SIGMAX is not referenced.

12: SIGTOL – REAL (KIND=nag_wp) *Input*

On entry: the tolerance used in assessing the convergence of the estimate of $\sigma_1(\bar{A}) = \left\|\bar{A}\right\|_2$ when the bisection method is used.

If SIGTOL ≤ 0.0, the default value SIGTOL $= 0.01$ is used. The actual value used is $\max(\text{SIGTOL}, \epsilon)$.

If SIGCMP = 'N' or SIGMAX > 0.0, then SIGTOL is not referenced.

Suggested value: SIGTOL $= 0.01$ should be sufficient in most cases.

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

13: MAXITS – INTEGER *Input*

On entry: the maximum iteration number $k =$ MAXITS for which $\sigma_1(T_k)$ is computed by bisection (see also Section 3). If SIGCMP = 'N' or SIGMAX > 0.0, then MAXITS is not referenced.

Suggested value: MAXITS $= \min(10, n)$ when SIGTOL is of the order of its default value (0.01).

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

14: MONIT – INTEGER *Input*

On entry: if MONIT > 0, the frequency at which a monitoring step is executed by F11GSF: the current solution and residual iterates will be returned by F11GSF and a call to F11GTF made possible every MONIT iterations, starting from iteration number MONIT. Otherwise, no monitoring takes place. There are some additional computational costs involved in monitoring the solution and residual vectors when the Lanczos method (SYMMLQ) is used.

Constraint: MONIT $\leq$ MAXITN.

15: LWREQ – INTEGER *Output*

On exit: the minimum amount of workspace required by F11GSF. (See also Section 5 in F11GSF.)

16: WORK(LWORK) – COMPLEX (KIND=nag_wp) array *Communication Array*

On exit: the array WORK is initialized by F11GRF. It must **not** be modified before calling the next routine in the suite, namely F11GSF.

17: LWORK – INTEGER *Input*

On entry: the dimension of the array WORK as declared in the (sub)program from which F11GRF is called.

Constraint: LWORK ≥ 120.

Note: although the minimum value of LWORK ensures the correct functioning of F11GRF, a larger value is required by the other routines in the suite, namely F11GSF and F11GTF. The required value is as follows:

Method	**Requirements**
CG	LWORK $= 120 + 5n + p$
SYMMLQ	LWORK $= 120 + 6n + p$

where

$p = 2 \times (\text{MAXITS} + 1)$, when an estimate of $\sigma_1(A)$ (SIGMAX) is computed;

$p = 0$, otherwise.

18: IFAIL – INTEGER *Input/Output*

On entry: IFAIL must be set to 0, -1 or 1. If you are unfamiliar with this parameter you should refer to Section 3.3 in the Essential Introduction for details.

For environments where it might be inappropriate to halt program execution when an error is detected, the value -1 or 1 is recommended. If the output of error messages is undesirable, then the value 1 is recommended. Otherwise, if you are not familiar with this parameter, the recommended value is 0. **When the value -1 or 1 is used it is essential to test the value of IFAIL on exit.**

On exit: IFAIL $= 0$ unless the routine detects an error or a warning has been flagged (see Section 6).

6 Error Indicators and Warnings

If on entry IFAIL $= 0$ or -1, explanatory error messages are output on the current error message unit (as defined by X04AAF).

Errors or warnings detected by the routine:

IFAIL $= -i$

On entry, the ith argument had an illegal value.

IFAIL $= 1$

F11GRF has been called out of sequence.

IFAIL $= -99$

An unexpected error has been triggered by this routine. Please contact NAG.

See Section 3.8 in the Essential Introduction for further information.

IFAIL $= -399$

Your licence key may have expired or may not have been installed correctly.

See Section 3.7 in the Essential Introduction for further information.

IFAIL $= -999$

Dynamic memory allocation failed.

See Section 3.6 in the Essential Introduction for further information.

7 Accuracy

Not applicable.

8 Parallelism and Performance

Not applicable.

9 Further Comments

When $\sigma_1(\bar{A})$ is not supplied (SIGMAX ≤ 0.0) but it is required, it is estimated by F11GSF using either of the two methods described in Section 3, as specified by the parameter SIGCMP. In particular, if SIGCMP = 'S', then the computation of $\sigma_1(\bar{A})$ is deemed to have converged when the differences between three successive values of $\sigma_1(T_k)$ differ, in a relative sense, by less than the tolerance SIGTOL, i.e., when

$$\max\left(\frac{\left|\sigma_1^{(k)}-\sigma_1^{(k-1)}\right|}{\sigma_1^{(k)}},\frac{\left|\sigma_1^{(k)}-\sigma_1^{(k-2)}\right|}{\sigma_1^{(k)}}\right) \leq \text{SIGTOL}.$$

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.

10 Example

This example solves a complex Hermitian system of simultaneous linear equations using the conjugate gradient method, where the matrix of the coefficients A, has a random sparsity pattern. An incomplete Cholesky preconditioner is used (F11JAF and F11JBF).

10.1 Program Text

```
    Program f11grfe

!     F11GRF Example Program Text

!     Mark 25 Release. NAG Copyright 2014.

!     .. Use Statements ..
      Use nag_library, Only: f11grf, f11gsf, f11gtf, f11jnf, f11jpf, f11xsf,   &
                             nag_wp
!     .. Implicit None Statement ..
      Implicit None
!     .. Parameters ..
      Integer, Parameter               :: nin = 5, nout = 6
!     .. Local Scalars ..
      Real (Kind=nag_wp)               :: anorm, dscale, dtol, sigerr, sigmax, &
                                          sigtol, stplhs, stprhs, tol
      Integer                          :: i, ifail, ifail1, irevcm, iterm,     &
                                          itn, its, la, lfill, liwork, lwork,  &
                                          lwreq, maxitn, maxits, monit, n,     &
                                          nnz, nnzc, npivm
      Character (6)                    :: method
      Character (1)                    :: mic, norm, precon, pstrat, sigcmp,   &
                                          weight
!     .. Local Arrays ..
      Complex (Kind=nag_wp), Allocatable :: a(:), b(:), work(:), x(:)
      Real (Kind=nag_wp), Allocatable  :: wgt(:)
      Integer, Allocatable             :: icol(:), ipiv(:), irow(:), istr(:),  &
                                          iwork(:)
!     .. Executable Statements ..
      Write (nout,*) 'F11GRF Example Program Results'

!     Skip heading in data file

      Read (nin,*)
      Read (nin,*) n
      Read (nin,*) nnz
      la = 2*nnz
      liwork = 2*la + 7*n + 1
      lwork = 200

      Allocate (a(la),b(n),work(lwork),x(n),wgt(n),icol(la),ipiv(n),irow(la), &
        istr(n+1),iwork(liwork))

!     Read or initialize the parameters for the iterative solver

      Read (nin,*) method
      Read (nin,*) precon, sigcmp, norm, weight, iterm
      Read (nin,*) tol, maxitn
      Read (nin,*) monit
      anorm = 0.0E0_nag_wp
      sigmax = 0.0E0_nag_wp
      sigtol = 1.0E-2_nag_wp
      maxits = n

!     Read the parameters for the preconditioner

      Read (nin,*) lfill, dtol
      Read (nin,*) mic, dscale
      Read (nin,*) pstrat

!     Read the non-zero elements of the matrix A
```

```
      Do i = 1, nnz
        Read (nin,*) a(i), irow(i), icol(i)
      End Do

!     Read right-hand side vector b and initial approximate solution x

      Read (nin,*) b(1:n)
      Read (nin,*) x(1:n)

!     Calculate incomplete Cholesky factorization

!     ifail: behaviour on error exit
!             =0 for hard exit, =1 for quiet-soft, =-1 for noisy-soft
      ifail = 0
      Call f11jnf(n,nnz,a,la,irow,icol,lfill,dtol,mic,dscale,pstrat,ipiv,istr, &
        nnzc,npivm,iwork,liwork,ifail)

!     Call F11GRF to initialize the solver

      ifail = 0
      Call f11grf(method,precon,sigcmp,norm,weight,iterm,n,tol,maxitn,anorm, &
        sigmax,sigtol,maxits,monit,lwreq,work,lwork,ifail)

!     Call repeatedly F11GSF to solve the equations
!     Note that the arrays B and X are overwritten

!     On final exit, X will contain the solution and B the residual
!     vector

      irevcm = 0
      lwreq = lwork

      ifail = 1
loop: Do
        Call f11gsf(irevcm,x,b,wgt,work,lwreq,ifail)

        If (irevcm/=4) Then
          ifail1 = -1
          Select Case (irevcm)
          Case (1)

            Call f11xsf(n,nnz,a,irow,icol,'No checking',x,b,ifail1)

          Case (2)

            Call f11jpf(n,a,la,irow,icol,ipiv,istr,'No checking',x,b,ifail1)

          Case (3)

            ifail1 = 0
            Call f11gtf(itn,stplhs,stprhs,anorm,sigmax,its,sigerr,work,lwreq, &
              ifail1)

            Write (nout,99999) itn, stplhs
            Write (nout,99998)
            Write (nout,99997)(x(i),b(i),i=1,n)
          End Select
          If (ifail1/=0) irevcm = 6
        Else If (ifail/=0) Then
          Write (nout,99993) ifail
          Go To 100
        Else
          Exit loop
        End If
      End Do loop

!     Obtain information about the computation

      ifail1 = 0
      Call f11gtf(itn,stplhs,stprhs,anorm,sigmax,its,sigerr,work,lwreq,ifail1)
```

```
!     Print the output data

      Write (nout,99996)
      Write (nout,99995) 'Number of iterations for convergence:    ', itn
      Write (nout,99994) 'Residual norm:                           ', stplhs
      Write (nout,99994) 'Right-hand side of termination criterion:', stprhs
      Write (nout,99994) '1-norm of matrix A:                      ', anorm
      Write (nout,99994) 'Largest singular value of A_bar:         ', sigmax

!     Output x

      Write (nout,99998)
      Write (nout,99997)(x(i),b(i),i=1,n)
100   Continue

99999 Format (/1X,'Monitoring at iteration no.',I4/1X,1P,'residual no','rm: ', &
        E14.4)
99998 Format (6X,'Solution vector',12X,'Residual vector')
99997 Format (1X,1P,'(',E11.4,',',E11.4,')',2X,'(',E11.4,',',E11.4,')')
99996 Format (/1X,'Final Results')
99995 Format (1X,A,I4)
99994 Format (1X,A,1P,E14.4)
99993 Format (1X/1X,' ** F11GSF returned with IFAIL = ',I5)
    End Program f11grfe
```

10.2 Program Data

```
F11GRF Example Program Data
  9                     N
  23                    NNZ
  'CG'                   METHOD
  'P'  'S'  '1'  'N'  1  PRECON, SIGCMP, NORM, WEIGHT, ITERM
  1.0D-6   20            TOL, MAXITN
  2                     MONIT
  0  0.0                LFILL, DTOL
  'N'  0.0              MIC, DSCALE
  'M'                    PSTRAT
 ( 6., 0.)   1    1
 (-1., 1.)   2    1
 ( 6., 0.)   2    2
 ( 0., 1.)   3    2
 ( 5., 0.)   3    3
 ( 5., 0.)   4    4
 ( 2.,-2.)   5    1
 ( 4., 0.)   5    5
 ( 1., 1.)   6    3
 ( 2., 0.)   6    4
 ( 6., 0.)   6    6
 (-4., 3.)   7    2
 ( 0., 1.)   7    5
 (-1., 0.)   7    6
 ( 6., 0.)   7    7
 (-1.,-1.)   8    4
 ( 0.,-1.)   8    6
 ( 9., 0.)   8    8
 ( 1., 3.)   9    1
 ( 1., 2.)   9    5
 (-1., 0.)   9    6
 ( 1., 4.)   9    8
 ( 9., 0.)   9    9     A(I), IROW(I), ICOL(I), I=1,...,NNZ
 (  8.,  54.)
 (-10., -92.)
 ( 25.,  27.)
 ( 26., -28.)
 ( 54.,  12.)
 ( 26., -22.)
 ( 47.,  65.)
 ( 71., -57.)
 ( 60.,  70.)           B(I), I=1,...,N
```

```
(  0.,   0.)
(  0.,   0.)
(  0.,   0.)
(  0.,   0.)
(  0.,   0.)
(  0.,   0.)
(  0.,   0.)
(  0.,   0.)
(  0.,   0.)           X(I), I=1,...,N
```

10.3 Program Results

```
 F11GRF Example Program Results

 Monitoring at iteration no.   2
 residual norm:     1.4937E+01
      Solution vector            Residual vector
 ( 2.1423E-01, 4.5333E+00)  (-1.8370E+00, 3.6956E+00)
 (-1.6589E+00,-1.2672E+01)  (-6.5005E-01, 2.5458E-01)
 ( 2.4101E+00, 7.4551E+00)  (-1.2616E-01,-1.3625E-01)
 ( 4.4400E+00,-6.4174E+00)  (-1.3120E-01, 1.4130E-01)
 ( 9.1135E+00, 3.7812E+00)  (-1.1471E+00, 7.3386E-01)
 ( 4.4419E+00,-4.0382E+00)  (-5.5054E-01,-1.0535E+00)
 ( 1.4757E+00, 1.2662E+00)  ( 1.7165E+00,-1.4614E+00)
 ( 8.4872E+00,-3.5347E+00)  (-3.5829E-01, 2.8764E-01)
 ( 5.9948E+00, 9.6851E-01)  (-3.0278E-01,-3.5324E-01)

 Monitoring at iteration no.   4
 residual norm:     1.4602E+00
      Solution vector            Residual vector
 ( 1.0061E+00, 8.9847E+00)  ( 1.1524E-02,-2.8188E-02)
 ( 1.9637E+00,-7.9768E+00)  ( 1.3513E-02,-1.7345E-01)
 ( 3.0067E+00, 7.0285E+00)  ( 1.8173E-02, 1.9627E-02)
 ( 3.9830E+00,-5.9636E+00)  ( 1.8900E-02,-2.0354E-02)
 ( 5.0390E+00, 5.0432E+00)  (-9.0877E-02,-1.0895E-01)
 ( 6.0488E+00,-4.0771E+00)  (-2.3890E-01, 3.2440E-01)
 ( 6.9710E+00, 3.0168E+00)  ( 1.9031E-01,-1.5499E-02)
 ( 8.0118E+00,-1.9806E+00)  ( 5.1611E-02,-4.1435E-02)
 ( 9.0074E+00, 9.6458E-01)  ( 4.3615E-02, 5.0884E-02)

 Final Results
 Number of iterations for convergence:       5
 Residual norm:                               9.0594E-14
 Right-hand side of termination criterion:    2.7340E-03
 1-norm of matrix A:                          2.2000E+01
 Largest singular value of A_bar:             1.9624E+00
      Solution vector            Residual vector
 ( 1.0000E+00, 9.0000E+00)  (-1.7764E-15, 0.0000E+00)
 ( 2.0000E+00,-8.0000E+00)  ( 3.5527E-15,-2.8422E-14)
 ( 3.0000E+00, 7.0000E+00)  (-3.5527E-15, 3.5527E-15)
 ( 4.0000E+00,-6.0000E+00)  ( 3.5527E-15,-7.1054E-15)
 ( 5.0000E+00, 5.0000E+00)  (-7.1054E-15, 3.5527E-15)
 ( 6.0000E+00,-4.0000E+00)  (-7.1054E-15, 0.0000E+00)
 ( 7.0000E+00, 3.0000E+00)  ( 0.0000E+00, 0.0000E+00)
 ( 8.0000E+00,-2.0000E+00)  ( 0.0000E+00,-7.1054E-15)
 ( 9.0000E+00, 1.0000E+00)  ( 0.0000E+00,-1.4211E-14)
```