F11GDF – NAG Fortran Library Routine Document

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

F11GDF is a set-up routine, the first in a suite of three routines for the iterative solution of a symmetric system of simultaneous linear equations. F11GDF must be called before F11GEF, the iterative solver. The third routine in the suite, F11GFF, 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 F11GDF(METHOD, PRECON, SIGCMP, NORM, WEIGHT, ITERM, N,
1                  TOL, MAXITN, ANORM, SIGMAX, SIGTOL, MAXITS,
2                  MONIT, LWREQ, WORK, LWORK, IFAIL)
 INTEGER           ITERM, N, MAXITN, MAXITS, MONIT, LWREQ, LWORK,
1                  IFAIL
 real              TOL, ANORM, SIGMAX, SIGTOL
 CHARACTER*1       PRECON, SIGCMP, NORM, WEIGHT
 CHARACTER*(*)     METHOD
 real              WORK(LWORK)
```

3 Description

The suite consisting of the routines F11GDF, F11GEF, F11GFF is designed to solve the symmetric system of simultaneous linear equations $Ax = b$ of order n, where n is large and the matrix of the coefficients A is sparse.

F11GDF is a set-up routine which must be called before F11GEF, the iterative solver. The third routine in the suite, F11GFF can be used to return additional information about the computation. Either of two methods can be used:

Conjugate Gradient Method For this method (Hestenes and Stiefel [1], Golub and Van Loan [2], Barrett *et al.* [3], Dias da Cunha and Hopkins [4]), 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 [5], Barrett *et al.* [3]), 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 conjugate gradient and Lanczos (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 (Golub and Van Loan [2]). A sequence of symmetric tridiagonal matrices $\{T_k\}$ is also generated. Here and in the following, the index k denotes the iteration count. The resulting 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 symmetric tridiagonal form, $T_n = Q^T AQ$; 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 symmetric tridiagonal systems of equations: the conjugate gradient method is equivalent to carrying out an LDL^T (Cholesky) factorization whereas the Lanczos method (SYMMLQ) uses an LQ factorization.

Faster convergence can be achieved using a **preconditioner** (Golub and Van Loan [2], Barrett *et al.* [3]). 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 **symmetric and positive-definite**, i.e., representable by $M = EE^T$, where M is non-singular, and such that $\bar{A} = E^{-1}AE^{-T} \sim I_n$ in (1), where I_n is the identity matrix of order n. Also, we can define $\bar{r} = E^{-1}r$ and $\bar{x} = E^Tx$. 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 in not required at any stage.

The first termination criterion

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

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 (Higham [6]). 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

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

is available only for the Lanczos method (SYMMLQ). In (3), $\sigma_1(\bar{A}) = \|\bar{A}\|_2$ 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 $\sigma_1(\bar{A})$ is not supplied, facilities are provided for its estimation by $\sigma_1(\bar{A}) \sim \max_k \sigma_1(T_k)$. 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 $\sigma_1(\bar{A}) \sim \max_k \|T_k\|_1$ can be used, where $\sigma_1(\bar{A}) \leq \|T_k\|_1$ 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., $\|v\|_p^{(w)} = \|v^{(w)}\|_p$, where $(v^{(w)})_i = w_i\,v_i$, 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 set-up routine F11GDF must be called, followed by the solver F11GEF. F11GFF can be called either when F11GEF is carrying out a monitoring step or after F11GEF has completed its tasks. Incorrect sequencing will raise an error condition.

4 References

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

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

[3] 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

[4] 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

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

[6] 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

5 Parameters

1: METHOD — CHARACTER*(*) *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'.

2: PRECON — CHARACTER*1 *Input*

On entry: determines whether preconditioning is used. The possible choices are:

'N'	no preconditioning;
'P'	preconditioning.

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

3: SIGCMP — CHARACTER*1 *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 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.
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 (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. The possible choices are:

'1'	use the l_1 norm;
'I'	use the l_∞ norm;
'2'	use the l_2 norm.

Suggested value: NORM = 'I', if ITERM = 1; NORM = '2', if ITERM = 2 (see below).

Constraints: if ITERM = 1, then NORM = '1', 'I' or '2'; if ITERM = 2, then 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): $\|v\|_p^{(w)} = \|v^{(w)}\|_p$, 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. The possible choices are:

'W'	user-supplied weights are to be used and must be supplied on initial entry to F11GEF.
'N'	all weights are implicitly set equal to one. Weights do not need to be supplied on initial entry to F11GEF.

Suggested value: WEIGHT = 'N'.

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

6: ITERM — INTEGER *Input*

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

1	use the termination criterion defined in (2) (both conjugate gradient and Lanczos (SYMMLQ) methods);
2	use the termination criterion defined in (3) (Lanczos method (SYMMLQ) only).

Suggested value: ITERM = 1.

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

7: N — INTEGER *Input*

On entry: the order n of the matrix A.

Constraint: N > 0.

8: TOL — ***real*** *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*** *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 F11GEF.
If ITERM = 2, then ANORM is not referenced.

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

11: SIGMAX — ***real*** *Input*

On entry: if SIGMAX > 0.0, the value of $\sigma_1(\bar{A}) = \|E^{-1}AE^{-T}\|_2$.
If SIGMAX ≤ 0.0, $\sigma_1(\bar{A})$ is estimated by F11GEF 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*** *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 = 0.01 is used. The actual value used is max (SIGTOL,ϵ).
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, then 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, then 1 ≤ MAXITS ≤ MAXITN.

14: MONIT — INTEGER *Input*

On entry: if MONIT > 0, the frequency at which a monitoring step is executed by F11GEF: the current solution and residual iterates will be returned by F11GEF and a call to F11GFF made possible every MONIT iterations, starting from the MONIT-th. 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 ≤ MAXITN.

15: LWREQ — INTEGER *Output*

On exit: the minimum amount of workspace required by F11GEF. (See also Section 5 of the document for F11GEF.)

16: WORK (LWORK) — ***real*** array *Output*

On exit: the workspace WORK is initialised by F11GDF. It must **not** be modified before calling the next routine in the suite, namely F11GEF.

17: LWORK — INTEGER *Input*

On entry: the dimension of the array WORK as declared in the (sub)program from which F11GDF was called.

Constraint: LWORK ≥ 120.

Note: although the minimum value of LWORK ensures the correct functioning of F11GDF, a larger value is required by the other routines in the suite, namely F11GEF and F11GFF. The required value is as follows:

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

where

$p = 2 *$ 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. For users not familiar with this parameter (described in Chapter P01) the recommended value is 0.

On exit: IFAIL $= 0$ unless the routine detects an error (see Section 6).

6 Errors 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 detected by the routine:

IFAIL $= -i$

On entry, the i-th argument had an illegal value.

IFAIL $= 1$

F11GDF has been called out of sequence.

7 Accuracy

Not applicable.

8 Further Comments

When $\sigma_1(\bar{A})$ is not supplied (SIGMAX ≤ 0.0) but it is required, it is estimated by F11GEF 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

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

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.

9 Example

The example solves a 20×20 symmetric system of simultaneous linear equations using the conjugate gradient method, where the matrix of the coefficients A, has random sparsity pattern. An incomplete Cholesky preconditioner is used (F11JAF and F11JBF).

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.

```
*     F11GDF Example Program Text
*     NAG Fortran SMP Library, Release 2.  NAG Copyright 2000.
*     .. Parameters ..
      INTEGER          NIN, NOUT
      PARAMETER        (NIN=5,NOUT=6)
      INTEGER          NMAX, LA, LIWORK, LWORK
      PARAMETER        (NMAX=1000,LA=10000,LIWORK=2*LA+7*NMAX+1,
     +                 LWORK=6*NMAX)
*     .. Local Scalars ..
      DOUBLE PRECISION ANORM, DSCALE, DTOL, SIGERR, SIGMAX, SIGTOL,
     +                 STPLHS, STPRHS, TOL
      INTEGER          I, IFAIL, IFAIL1, IREVCM, ITERM, ITN, ITS, LFILL,
     +                 LWREQ, MAXITN, MAXITS, MONIT, N, NNZ, NNZC, NPIVM
      LOGICAL          LOOP
      CHARACTER        MIC, NORM, PRECON, PSTRAT, SIGCMP, WEIGHT
      CHARACTER*6      METHOD
*     .. Local Arrays ..
      DOUBLE PRECISION A(LA), B(NMAX), WGT(NMAX), WORK(LWORK), X(NMAX)
      INTEGER          ICOL(LA), IPIV(NMAX), IROW(LA), ISTR(NMAX+1),
     +                 IWORK(LIWORK)
*     .. External Subroutines ..
      EXTERNAL         F11GDF, F11GEF, F11GFF, F11JAF, F11JBF, F11XEF
*     .. Executable Statements ..
      WRITE (NOUT,*) 'F11GDF Example Program Results'
*
*     Skip heading in data file
*
      READ (NIN,*)
      READ (NIN,*) N
      IF (N.LE.NMAX) THEN
*
*     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 = 0.0D0
         SIGMAX = 0.0D0
         SIGTOL = 1.0D-2
         MAXITS = N
*
*     Read the parameters for the preconditioner
*
         READ (NIN,*) LFILL, DTOL
         READ (NIN,*) MIC, DSCALE
         READ (NIN,*) PSTRAT
```

```
*
*     Read the number of non-zero elements of the matrix A, then read
*     the non-zero elements
*
         READ (NIN,*) NNZ
         DO 20 I = 1, NNZ
            READ (NIN,*) A(I), IROW(I), ICOL(I)
   20    CONTINUE
*
*     Read right-hand side vector b and initial approximate solution x
*
         READ (NIN,*) (B(I),I=1,N)
         READ (NIN,*) (X(I),I=1,N)
*
*     Calculate incomplete Cholesky factorization
*
         IFAIL = 0
         CALL F11JAF(N,NNZ,A,LA,IROW,ICOL,LFILL,DTOL,MIC,DSCALE,PSTRAT,
     +               IPIV,ISTR,NNZC,NPIVM,IWORK,LIWORK,IFAIL)
*
*     Call F11GDF to initialize the solver
*
         IFAIL = 0
         CALL F11GDF(METHOD,PRECON,SIGCMP,NORM,WEIGHT,ITERM,N,TOL,
     +               MAXITN,ANORM,SIGMAX,SIGTOL,MAXITS,MONIT,LWREQ,WORK,
     +               LWORK,IFAIL)
*
*     Call repeatedly F11GEF 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
*
         IFAIL = 0
         IREVCM = 0
         LOOP = .TRUE.
*
         LWREQ = LWORK
   40    CONTINUE
         CALL F11GEF(IREVCM,X,B,WGT,WORK,LWREQ,IFAIL)
         IF (IREVCM.EQ.1) THEN
            IFAIL1 = -1
            CALL F11XEF(N,NNZ,A,IROW,ICOL,'No checking',X,B,IFAIL1)
            IF (IFAIL1.NE.0) IREVCM = 6
         ELSE IF (IREVCM.EQ.2) THEN
            IFAIL1 = -1
            CALL F11JBF(N,A,LA,IROW,ICOL,IPIV,ISTR,'No checking',X,B,
     +                  IFAIL1)
            IF (IFAIL1.NE.0) IREVCM = 6
         ELSE IF (IREVCM.EQ.3) THEN
            IFAIL1 = 0
            CALL F11GFF(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)
         ELSE IF (IREVCM.EQ.4) THEN
            LOOP = .FALSE.
```

```
         END IF
         IF (LOOP) GO TO 40
*
*     Obtain information about the computation
*
         IFAIL1 = 0
         CALL F11GFF(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)
      END IF
      STOP
*
99999 FORMAT (/1X,'Monitoring at iteration no.',I4,/1X,1P,'residual no',
     +       'rm: ',D14.4)
99998 FORMAT (2X,'Solution vector',2X,'Residual vector')
99997 FORMAT (1X,1P,D16.4,1X,D16.4)
99996 FORMAT (/1X,'Final Results')
99995 FORMAT (1X,A,I4)
99994 FORMAT (1X,A,1P,D14.4)
      END
```

9.2 Program Data

```
F11GDF Example Program Data
  7                     N
  '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
 16                     NNZ
  4.   1    1
  1.   2    1
  5.   2    2
  2.   3    3
  2.   4    2
  3.   4    4
 -1.   5    1
```

```
 1.   5    4
 4.   5    5
 1.   6    2
-2.   6    5
 3.   6    6
 2.   7    1
-1.   7    2
-2.   7    3
 5.   7    7         A(I), IROW(I), ICOL(I), I=1,...,NNZ
15.  18.  -8.  21.
11.  10.  29.        B(I), I=1,...,N
 0.   0.   0.   0.
 0.   0.   0.        X(I), I=1,...,N
```

9.3 Program Results

```
F11GDF Example Program Results

Monitoring at iteration no.   2
residual norm:     1.9938D+00
 Solution vector  Residual vector
      9.6320D-01      -2.2960D-01
      1.9934D+00       2.2254D-01
      3.0583D+00       9.5827D-02
      4.1453D+00      -2.5155D-01
      4.8289D+00      -1.7160D-01
      5.6630D+00       6.7533D-01
      7.1062D+00      -3.4737D-01

Monitoring at iteration no.   4
residual norm:     6.6574D-03
 Solution vector  Residual vector
      9.9940D-01      -1.0551D-03
      2.0011D+00      -2.4675D-03
      3.0008D+00      -1.7116D-05
      3.9996D+00       4.4929D-05
      4.9991D+00       2.1359D-03
      5.9993D+00      -8.7482D-04
      7.0007D+00       6.2045D-05

Final Results
Number of iterations for convergence:       5
Residual norm:                               9.7700D-15
Right-hand side of termination criterion:    3.9200D-04
1-norm of matrix A:                          1.0000D+01
Largest singular value of A_bar:             1.3596D+00
 Solution vector  Residual vector
      1.0000D+00       0.0
      2.0000D+00       3.5527D-15
      3.0000D+00      -8.8818D-16
      4.0000D+00       0.0
      5.0000D+00      -1.7764D-15
      6.0000D+00       0.0
      7.0000D+00       3.5527D-15
```