NAG Fortran Library Routine Document

F11DCF

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

F11DCF solves a real sparse nonsymmetric system of linear equations, represented in coordinate storage format, using a restarted generalized minimal residual (RGMRES), conjugate gradient squared (CGS), stabilized bi-conjugate gradient (Bi-CGSTAB), or transpose-free quasi-minimal residual (TFQMR) method, with incomplete LU preconditioning.

2 Specification

```
      SUBROUTINE F11DCF(METHOD, N, NNZ, A, LA, IROW, ICOL, IPIVP, IPIVQ, ISTR,
     1                  IDIAG, B, M, TOL, MAXITN, X, RNORM, ITN, WORK, LWORK,
     2                  IFAIL)
      INTEGER           N, NNZ, LA, IROW(LA), ICOL(LA), IPIVP(N), IPIVQ(N),
     1                  ISTR(N+1), IDIAG(N), M, MAXITN, ITN, LWORK, IFAIL
      real              A(LA), B(N), TOL, X(N), RNORM, WORK(LWORK)
      CHARACTER*(*)     METHOD
```

3 Description

This routine solves a real sparse nonsymmetric linear system of equations:

$$Ax = b,$$

using a preconditioned RGMRES (Saad and Schultz (1986)), CGS (Sonneveld (1989)), Bi-CGSTAB(ℓ) (Van der Vorst (1989), Sleijpen and Fokkema (1993)), or TFQMR (Freund and Nachtigal (1991), Freund (1993)) method.

F11DCF uses the incomplete LU factorization determined by F11DAF as the preconditioning matrix. A call to F11DCF must always be preceded by a call to F11DAF. Alternative preconditioners for the same storage scheme are available by calling F11DEF.

The matrix A, and the preconditioning matrix M, are represented in coordinate storage (CS) format (see Section 2.1.1 of the F11 Chapter Introduction) in the arrays A, IROW and ICOL, as returned from F11DAF. The array A holds the non-zero entries in these matrices, while IROW and ICOL hold the corresponding row and column indices.

F11DCF is a black-box routine which calls F11BDF, F11BEF and F11BFF. If you wish to use an alternative storage scheme, preconditioner, or termination criterion, or require additional diagnostic information, you should call these underlying routines directly.

4 References

Freund R W (1993) A transpose-free quasi-mimimal residual algorithm for non-Hermitian linear sytems *SIAM J. Sci. Comput.* **14** 470–482

Freund R W and Nachtigal N (1991) QMR: a Quasi-Minimal Residual Method for Non-Hermitian Linear Systems *Numer. Math.* **60** 315–339

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

Salvini S A and Shaw G J (1996) An evaluation of new NAG Library solvers for large sparse unsymmetric linear systems *NAG Technical Report TR2/96*

Sleijpen G L G and Fokkema D R (1993) BiCGSTAB(ℓ) for linear equations involving matrices with complex spectrum *ETNA* **1** 11–32

Sonneveld P (1989) CGS, a fast Lanczos-type solver for nonsymmetric linear systems *SIAM J. Sci. Statist. Comput.* **10** 36–52

Van der Vorst H (1989) Bi-CGSTAB, a fast and smoothly converging variant of Bi-CG for the solution of nonsymmetric linear systems *SIAM J. Sci. Statist. Comput.* **13** 631–644

5 Parameters

1: METHOD – CHARACTER*(*) *Input*

On entry: specifies the iterative method to be used. The possible choices are:

if METHOD = 'RGMRES', restarted generalized minimum residual method;

if METHOD = 'CGS', conjugate gradient squared method;

if METHOD = 'BICGSTAB', bi-conjugate gradient stabilized (ℓ) method;

if METHOD = 'TFQMR', transpose-free quasi-minimal residual method.

Constraint: METHOD = 'RGMRES', 'CGS', 'BICGSTAB' or 'TFQMR'.

2: N – INTEGER *Input*

On entry: n, the order of the matrix A. This **must** be the same value as was supplied in the preceding call to F11DAF.

Constraint: N $\geq$ 1.

3: NNZ – INTEGER *Input*

On entry: the number of non-zero elements in the matrix A. This **must** be the same value as was supplied in the preceding call to F11DAF.

Constraint: $1 \leq \mathrm{NNZ} \leq \mathrm{N}^2$.

4: A(LA) – ***real*** array *Input*

On entry: the values returned in array A by a previous call to F11DAF.

5: LA – INTEGER *Input*

On entry: the dimension of the arrays A, IROW and ICOL as declared in the (sub)program from which F11DCF is called. This **must** be the same value as was supplied in the preceding call to F11DAF.

Constraint: LA $\geq 2 \times$ NNZ.

6: IROW(LA) – INTEGER array *Input*
7: ICOL(LA) – INTEGER array *Input*
8: IPIVP(N) – INTEGER array *Input/Output*
9: IPIVQ(N) – INTEGER array *Input/Output*
10: ISTR(N+1) – INTEGER array *Input*
11: IDIAG(N) – INTEGER array *Input*

On entry: the values returned in arrays IROW, ICOL, IPIVP, IPIVQ, ISTR and IDIAG by a previous call to F11DAF.

On exit: IPIVP and IPIVQ are used as internal workspace prior to being restored and hence are unchanged.

12: B(N) – ***real*** array *Input*

On entry: the right-hand side vector b.

13: M – INTEGER *Input*

On entry: if METHOD = 'RGMRES', M is the dimension of the restart subspace.

If METHOD = 'BICGSTAB', M is the order ℓ of the polynomial Bi-CGSTAB method; otherwise, M is not referenced.

Constraints:

if METHOD = 'RGMRES', $0 < \mathrm{M} \leq \min(\mathrm{N}, 50)$,
if METHOD = 'BICGSTAB', $0 < \mathrm{M} \leq \min(\mathrm{N}, 10)$.

14: TOL – ***real*** *Input*

f11TOL-a;

15: MAXITN – INTEGER *Input*

On entry: the maximum number of iterations allowed.

Constraint: MAXITN ≥ 1.

16: X(N) – ***real*** array *Input/Output*

On entry: an initial approximation to the solution vector x.

On exit: an improved approximation to the solution vector x.

17: RNORM – ***real*** *Output*

On exit: the final value of the residual norm $\|r_k\|_\infty$, where k is the output value of ITN.

18: ITN – INTEGER *Output*

On exit: the number of iterations carried out.

19: WORK(LWORK) – ***real*** array *Workspace*
20: LWORK – INTEGER *Input*

On entry: the dimension of the array WORK as declared in the (sub)program from which F11DCF is called.

Constraints:

if METHOD = 'RGMRES' then LWORK $\geq 4 \times \mathrm{N} + \mathrm{M} \times (\mathrm{M} + \mathrm{N} + 5) + 101$,
if METHOD = 'CGS' then LWORK $\geq 8 \times \mathrm{N} + 100$,
if METHOD = 'BICGSTAB' then LWORK $\geq 2 \times \mathrm{N} \times (\mathrm{M} + 3) + \mathrm{M} \times (\mathrm{M} + 2) + 100$,
if METHOD = 'TFQMR' then LWORK $\geq 11 \times \mathrm{N} + 100$.

21: IFAIL – INTEGER *Input/Output*

On entry: IFAIL must be set to 0, -1 or 1. Users who are unfamiliar with this parameter should refer to Chapter P01 for details.

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

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, for users 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.**

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 $= 1$

On entry, METHOD $\neq$ 'RGMRES', 'CGS', 'BICGSTAB', or 'TFQMR',
or N < 1,
or NNZ < 1,
or NNZ $> \mathrm{N}^2$,
or LA $< 2 \times$ NNZ,
or M < 1 and METHOD = 'RGMRES' or METHOD = 'BICGSTAB',
or M $> \min(\mathrm{N}, 50)$, with METHOD = 'RGMRES',
or M $> \min(\mathrm{N}, 10)$, with METHOD = 'BICGSTAB',
or TOL ≥ 1.0,
or MAXITN < 1,
or LWORK too small.

IFAIL $= 2$

On entry, the CS representation of A is invalid. Further details are given in the error message. Check that the call to F11DCF has been preceded by a valid call to F11DAF, and that the arrays A, IROW, and ICOL have not been corrupted between the two calls.

IFAIL $= 3$

On entry, the CS representation of the preconditioning matrix M is invalid. Further details are given in the error message. Check that the call to F11DCF has been preceded by a valid call to F11DAF, and that the arrays A, IROW, ICOL, IPIVP, IPIVQ, ISTR and IDIAG have not been corrupted between the two calls.

IFAIL $= 4$

The required accuracy could not be obtained. However, a reasonable accuracy may have been obtained, and further iterations could not improve the result. You should check the output value of RNORM for acceptability. This error code usually implies that your problem has been fully and satisfactorily solved to within or close to the accuracy available on your system. Further iterations are unlikely to improve on this situation.

IFAIL $= 5$

Required accuracy not obtained in MAXITN iterations.

IFAIL $= 6$

A serious error has occurred in an internal call to F11BDF, F11BEF or F11BFF. Check all subroutine calls and array sizes. Seek expert help.

7 Accuracy

On successful termination, the final residual $r_k = b - Ax_k$, where $k =$ ITN, satisfies the termination criterion

$$\|r_k\|_\infty \leq \tau \times (\|b\|_\infty + \|A\|_\infty \|x_k\|_\infty).$$

The value of the final residual norm is returned in RNORM.

8 Further Comments

The time taken by F11DCF for each iteration is roughly proportional to the value of NNZC returned from the preceding call to F11DAF.

The number of iterations required to achieve a prescribed accuracy cannot be easily determined a priori, as it can depend dramatically on the conditioning and spectrum of the preconditioned coefficient matrix $\bar{A} = \mathrm{M}^{-1}A$.

Some illustrations of the application of F11DCF to linear systems arising from the discretization of two-dimensional elliptic partial differential equations, and to random-valued randomly structured linear systems, can be found in Salvini and Shaw (1996).

9 Example

This example program solves a sparse linear system of equations using the CGS method, with incomplete LU preconditioning.

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.

```
*     F11DCF Example Program Text
*     Mark 19 Revised. NAG Copyright 1999.
*     .. Parameters ..
      INTEGER          NIN, NOUT
      PARAMETER        (NIN=5,NOUT=6)
      INTEGER          NMAX, LA, LIWORK, LWORK
      PARAMETER        (NMAX=1000,LA=10000,LIWORK=7*NMAX+2,LWORK=10000)
*     .. Local Scalars ..
      real             DTOL, RNORM, TOL
      INTEGER          I, IFAIL, ITN, LFILL, LWREQ, M, MAXITN, N, NNZ,
     +                 NNZC, NPIVM
      CHARACTER        MILU, PSTRAT
      CHARACTER*8      METHOD
*     .. Local Arrays ..
      real             A(LA), B(NMAX), WORK(LWORK), X(NMAX)
      INTEGER          ICOL(LA), IDIAG(NMAX), IPIVP(NMAX), IPIVQ(NMAX),
     +                 IROW(LA), ISTR(NMAX+1), IWORK(LIWORK)
*     .. External Subroutines ..
      EXTERNAL         F11DAF, F11DCF
*     .. Intrinsic Functions ..
      INTRINSIC        MAX
*     .. Executable Statements ..
      WRITE (NOUT,*) 'F11DCF Example Program Results'
      WRITE (NOUT,*)
*     Skip heading in data file
      READ (NIN,*)
*
*     Read algorithmic parameters
*
      READ (NIN,*) N
      IF (N.LE.NMAX) THEN
         READ (NIN,*) NNZ
         READ (NIN,*) METHOD
         READ (NIN,*) LFILL, DTOL
         READ (NIN,*) PSTRAT
         READ (NIN,*) MILU
         READ (NIN,*) M, TOL, MAXITN
*
*        Check size of workspace
*
         LWREQ = MAX(4*N+M*(M+N+5)+101,8*N+100,2*N*(M+3)+M*(M+2)+100,
     +           11*N+100)
         IF (LWORK.LT.LWREQ) THEN
            WRITE (NOUT,'(A,I4)') 'LWORK must be at least', LWREQ
```

```
            STOP
         END IF
*
*        Read the matrix A
*
         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 LU factorization
*
         IFAIL = 0
         CALL F11DAF(N,NNZ,A,LA,IROW,ICOL,LFILL,DTOL,PSTRAT,MILU,IPIVP,
     +               IPIVQ,ISTR,IDIAG,NNZC,NPIVM,IWORK,LIWORK,IFAIL)
*
*        Solve Ax = b using F11DCF
*
         CALL F11DCF(METHOD,N,NNZ,A,LA,IROW,ICOL,IPIVP,IPIVQ,ISTR,IDIAG,
     +               B,M,TOL,MAXITN,X,RNORM,ITN,WORK,LWORK,IFAIL)
*
         WRITE (NOUT,'(A,I10,A)') ' Converged in', ITN, ' iterations'
         WRITE (NOUT,'(A,1P,D16.3)') ' Final residual norm =', RNORM
         WRITE (NOUT,*)
*
*        Output x
*
         WRITE (NOUT,*) '           X'
         DO 40 I = 1, N
            WRITE (NOUT,'(1X,1P,D16.4)') X(I)
   40    CONTINUE
      END IF
*
      STOP
      END
```

9.2 Program Data

```
F11DCF Example Program Data
  8                       N
  24                      NNZ
  'CGS'                   METHOD
  0  0.0                  LFILL, DTOL
  'C'                     PSTRAT
  'N'                     MILU
  4  1.0e-10   100        M, TOL, MAXITN
  2.   1    1
 -1.   1    4
  1.   1    8
  4.   2    1
 -3.   2    2
  2.   2    5
 -7.   3    3
  2.   3    6
  3.   4    1
 -4.   4    3
  5.   4    4
  5.   4    7
 -1.   5    2
  8.   5    5
 -3.   5    7
 -6.   6    1
  5.   6    3
  2.   6    6
 -5.   7    3
 -1.   7    5
```

```
  6.   7    7
 -1.   8    2
  2.   8    6
  3.   8    8          A(I), IROW(I), ICOL(I), I=1,...,NNZ
  6.   8.  -9.  46.
 17.  21.  22.  34.   B(I), I=1,...,N
  0.   0.   0.   0.
  0.   0.   0.   0.   X(I), I=1,...,N
```

9.3 Program Results

```
 F11DCF Example Program Results

 Converged in         4 iterations
 Final residual norm =       2.132E-14

           X
       1.0000E-00
       2.0000E+00
       3.0000E+00
       4.0000E+00
       5.0000E+00
       6.0000E+00
       7.0000E+00
       8.0000E+00
```