NAG Fortran Library Routine Document

F11DSF

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

F11DSF solves a complex sparse non-Hermitian 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, without preconditioning, with Jacobi, or with SSOR preconditioning.

2 Specification

```
 SUBROUTINE F11DSF(METHOD, PRECON, N, NNZ, A, IROW, ICOL, OMEGA, B, M,
1                  TOL, MAXITN, X, RNORM, ITN, WORK, LWORK, IWORK, IFAIL)
 INTEGER           N, NNZ, IROW(NNZ), ICOL(NNZ), M, MAXITN, ITN, LWORK,
1                  IWORK(2*N+1), IFAIL
 real              OMEGA, TOL, RNORM
 complex           A(NNZ), B(N), X(N), WORK(LWORK)
 CHARACTER*(*)     METHOD
 CHARACTER*1       PRECON
```

3 Description

This routine solves a complex sparse non-Hermitian system of linear equations:

$$Ax = b,$$

using an 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.

F11DSF allows the following choices for the preconditioner:

no preconditioning;

Jacobi preconditioning (Young (1971));

symmetric successive-over-relaxation (SSOR) preconditioning (Young (1971)).

For incomplete LU (ILU) preconditioning see F11DQF.

The matrix A is represented in coordinate storage (CS) format (see Section 2.1.1 of the F11 Chapter Introduction) in the arrays A, IROW and ICOL. The array A holds the non-zero entries in the matrix, while IROW and ICOL hold the corresponding row and column indices.

F11DSF is a black-box routine which calls F11BRF, F11BSF and F11BTF. 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

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

Young D (1971) *Iterative Solution of Large Linear Systems* Academic Press, New York

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: PRECON – CHARACTER*1 *Input*

On entry: specifies the type of preconditioning to be used. The possible choices are:

if PRECON = 'N', no preconditioning;

if PRECON = 'J', Jacobi;

if PRECON = 'S', symmetric successive-over-relaxation (SSOR).

Constraint: PRECON = 'N', 'J' or 'S'.

3: N – INTEGER *Input*

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

Constraint: N ≥ 1.

4: NNZ – INTEGER *Input*

On entry: the number of non-zero elements in the matrix A.

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

5: A(NNZ) – ***complex*** array *Input*

On entry: the non-zero elements of the matrix A, ordered by increasing row index, and by increasing column index within each row. Multiple entries for the same row and column indices are not permitted. The routine F11ZNF may be used to order the elements in this way.

6: IROW(NNZ) – INTEGER array *Input*

7: ICOL(NNZ) – INTEGER array *Input*

On entry: the row and column indices of the non-zero elements supplied in A.

Constraints: IROW and ICOL must satisfy the following constraints (which may be imposed by a call to F11ZNF):

$1 \leq \mathrm{IROW}(i) \leq \mathrm{N}$ and $1 \leq \mathrm{ICOL}(i) \leq \mathrm{N}$, for $i = 1, 2, \ldots, \mathrm{NNZ}$;
$\mathrm{IROW}(i-1) < \mathrm{IROW}(i)$, or $\mathrm{IROW}(i-1) = \mathrm{IROW}(i)$ and $\mathrm{ICOL}(i-1) < \mathrm{ICOL}(i)$, for $i = 2, 3, \ldots, \mathrm{NNZ}$.

8: OMEGA – ***real*** *Input*

On entry: if PRECON = 'S', OMEGA is the relaxation parameter ω to be used in the SSOR method. Otherwise OMEGA need not be initialised and is not referenced.

Constraint: $0.0 < \text{OMEGA} < 2.0$.

9: B(N) – ***complex*** array *Input*

On entry: the right-hand side vector b.

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

11: TOL – ***real*** *Input*

On entry: the required tolerance. Let x_k denote the approximate solution at iteration k, and r_k the corresponding residual. The algorithm is considered to have converged at iteration k if

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

If $\text{TOL} \leq 0.0$, $\tau = \max(\sqrt{\epsilon}, \sqrt{n}\,\epsilon)$ is used, where ϵ is the ***machine precision***. Otherwise $\tau = \max(\text{TOL}, 10\epsilon, \sqrt{n}\,\epsilon)$ is used.

Constraint: $\text{TOL} < 1.0$.

12: MAXITN – INTEGER *Input*

On entry: the maximum number of iterations allowed.

Constraint: $\text{MAXITN} \geq 1$.

13: X(N) – ***complex*** array *Input/Output*

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

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

14: RNORM – ***real*** *Output*

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

15: ITN – INTEGER *Output*

On exit: the number of iterations carried out.

16: WORK(LWORK) – ***complex*** array *Workspace*
17: LWORK – INTEGER *Input*

On entry: the dimension of the array WORK as declared in the (sub)program from which F11DSF is called.

Constraint:

if METHOD = 'RGMRES', $\text{LWORK} \geq 4 \times \text{N} + \text{M} \times (\text{M} + \text{N} + 5) + \nu + 121$,
if METHOD = 'CGS', $\text{LWORK} \geq 8 \times \text{N} + \nu + 120$,
if METHOD = 'BICGSTAB', $\text{LWORK} \geq 2 \times \text{N} \times (\text{M} + 3) + \text{M} \times (\text{M} + 2) + \nu + 120$,
if METHOD = 'TFQMR', $\text{LWORK} \geq 11 \times \text{N} + \nu + 120$.

where $\nu = \text{N}$ for PRECON = 'J' or 'S', and 0 otherwise.

18: IWORK(2*N+1) – INTEGER array *Workspace*

19: 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 PRECON $\neq$ 'N', 'J' or 'S',
or N < 1,
or NNZ < 1,
or NNZ $>$ N^2,
or PRECON $=$ 'S' and OMEGA lies outside the interval (0.0,2.0),
or M < 1,
or M $> \min(\text{N}, 50)$, when METHOD $=$ 'RGMRES',
or M $> \min(\text{N}, 10)$, when METHOD $=$ 'BICGSTAB',
or TOL ≥ 1.0,
or MAXITN < 1,
or LWORK is too small.

IFAIL $= 2$

On entry, the arrays IROW and ICOL fail to satisfy the following constraints:

$1 \leq \text{IROW}(i) \leq \text{N}$ and $1 \leq \text{ICOL}(i) \leq \text{N}$, for $i = 1, 2, \ldots, \text{NNZ}$;

$\text{IROW}(i-1) < \text{IROW}(i)$, or $\text{IROW}(i-1) = \text{IROW}(i)$ and $\text{ICOL}(i-1) < \text{ICOL}(i)$, for $i = 2, 3, \ldots, \text{NNZ}$.

Therefore a non-zero element has been supplied which does not lie within the matrix A, is out of order, or has duplicate row and column indices. Call F11ZNF to reorder and sum or remove duplicates.

IFAIL $= 3$

On entry, the matrix A has a zero diagonal element. Jacobi and SSOR preconditioners are therefore not appropriate for this problem.

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

Algorithmic breakdown. A solution is returned, although it is possible that it is completely inaccurate.

IFAIL = 7

A serious error has occurred in an internal call to F11BRF, F11BSF or F11BTF. Check all subroutine calls and array sizes. Seek expert help.

7 Accuracy

On successful termination, the final residual $r_k = b - Ax_k$, where $k = \mathrm{ITN}$, 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 F11DSF for each iteration is roughly proportional to NNZ.

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

9 Example

This example program solves a complex sparse non-Hermitian system of equations using the CGS method, with no 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.

```
*     F11DSF Example Program Text.
*     Mark 19 Release. NAG Copyright 1999.
*     .. Parameters ..
      INTEGER          NIN, NOUT
      PARAMETER        (NIN=5,NOUT=6)
      INTEGER          NMAX, LA, LWORK
      PARAMETER        (NMAX=1000,LA=10000,LWORK=10000)
*     .. Local Scalars ..
      real             OMEGA, RNORM, TOL
      INTEGER          I, IFAIL, ITN, L, LWREQ, M, MAXITN, N, NNZ
      CHARACTER        PRECON
      CHARACTER*8      METHOD
*     .. Local Arrays ..
      complex          A(LA), B(NMAX), WORK(LWORK), X(NMAX)
      INTEGER          ICOL(LA), IROW(LA), IWORK(2*NMAX+1)
*     .. External Subroutines ..
      EXTERNAL         F11DSF
*     .. Intrinsic Functions ..
      INTRINSIC        MAX
*     .. Executable Statements ..
      WRITE (NOUT,*) 'F11DSF 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, PRECON
         READ (NIN,*) OMEGA
         READ (NIN,*) M, TOL, MAXITN
*
*        Check size of workspace
*
         L = N
         IF (PRECON.EQ.'N' .OR. PRECON.EQ.'n') L = 0
         LWREQ = MAX(4*N+M*(M+N+5)+L+121,8*N+L+120,2*N*(M+3)+M*(M+2)
     +           +L+120,11*N+L+120)
         IF (LWORK.LT.LWREQ) THEN
            WRITE (NOUT,*) '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 rhs vector b and initial approximate solution x
*
         READ (NIN,*) (B(I),I=1,N)
         READ (NIN,*) (X(I),I=1,N)
*
*        Solve Ax = b using F11DSF
*
         IFAIL = 0
         CALL F11DSF(METHOD,PRECON,N,NNZ,A,IROW,ICOL,OMEGA,B,M,TOL,
     +              MAXITN,X,RNORM,ITN,WORK,LWORK,IWORK,IFAIL)
*
         WRITE (NOUT,'(1X,A,I10,A)') 'Converged in', ITN, ' iterations'
         WRITE (NOUT,'(1X,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,'','',1P,D16.4,'')'')') X(I)
   40    CONTINUE
*
      END IF
      STOP
      END
```

9.2 Program Data

```
F11DSF Example Program Data
  5                     N
 16                     NNZ
  'CGS' 'N'             METHOD, PRECON
  1.05                  OMEGA
  1 1.e-10 1000         M, TOL, MAXITN
 ( 2., 3.)   1    1
 ( 1.,-1.)   1    2
 (-1., 0.)   1    4
 ( 0., 2.)   2    2
 (-2., 1.)   2    3
 ( 1., 0.)   2    5
 ( 0.,-1.)   3    1
 ( 5., 4.)   3    3
 ( 3.,-1.)   3    4
```

```
( 1., 0.)   3    5
(-2., 2.)   4    1
(-3., 1.)   4    4
( 0., 3.)   4    5
( 4.,-2.)   5    2
(-2., 0.)   5    3
(-6., 1.)   5    5      A(I), IROW(I), ICOL(I), I=1,...,NNZ
( -3., 3.)
(-11., 5.)
( 23.,48.)
(-41., 2.)
(-28.,-31.)             B(I), I=1,...,N
( 0., 0.)
( 0., 0.)
( 0., 0.)
( 0., 0.)
( 0., 0.)               X(I), I=1,...,N
```

9.3 Program Results

```
 F11DSF Example Program Results

 Converged in         5 iterations
 Final residual norm =       1.422E-10

                    X
(      1.0000E+00,      2.0000E+00)
(      2.0000E+00,      3.0000E+00)
(      3.0000E+00,      4.0000E+00)
(      4.0000E+00,      5.0000E+00)
(      5.0000E+00,      6.0000E+00)
```