NAG Library Routine Document

F11DQF

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

F11DQF 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, with incomplete LU preconditioning.

2 Specification

```
SUBROUTINE F11DQF (METHOD, N, NNZ, A, LA, IROW, ICOL, IPIVP, IPIVQ, ISTR,       &
                   IDIAG, B, M, TOL, MAXITN, X, RNORM, ITN, WORK, LWORK,        &
                   IFAIL)

INTEGER                 N, NNZ, LA, IROW(LA), ICOL(LA), IPIVP(N), IPIVQ(N),    &
                        ISTR(N+1), IDIAG(N), M, MAXITN, ITN, LWORK, IFAIL
REAL (KIND=nag_wp)      TOL, RNORM
COMPLEX (KIND=nag_wp)   A(LA), B(N), X(N), WORK(LWORK)
CHARACTER(*)            METHOD
```

3 Description

F11DQF solves a complex sparse non-Hermitian linear system of equations

$$Ax = b,$$

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

F11DQF uses the incomplete LU factorization determined by F11DNF as the preconditioning matrix. A call to F11DQF must always be preceded by a call to F11DNF. Alternative preconditioners for the same storage scheme are available by calling F11DSF.

The matrix A, and the preconditioning matrix M, are represented in coordinate storage (CS) format (see Section 2.1.1 in the F11 Chapter Introduction) in the arrays A, IROW and ICOL, as returned from F11DNF. The array A holds the nonzero entries in these matrices, while IROW and ICOL hold the corresponding row and column indices.

F11DQF 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-minimal residual algorithm for non-Hermitian linear systems *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

5 Parameters

1: METHOD – CHARACTER(*) *Input*

On entry: specifies the iterative method to be used.

METHOD = 'RGMRES'
Restarted generalized minimum residual method.

METHOD = 'CGS'
Conjugate gradient squared method.

METHOD = 'BICGSTAB'
Bi-conjugate gradient stabilized (ℓ) method.

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 F11DNF.

Constraint: N ≥ 1.

3: NNZ – INTEGER *Input*

On entry: the number of nonzero elements in the matrix A. This **must** be the same value as was supplied in the preceding call to F11DNF.

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

4: A(LA) – COMPLEX (KIND=nag_wp) array *Input*

On entry: the values returned in the array A by a previous call to F11DNF.

5: LA – INTEGER *Input*

On entry: the dimension of the arrays A, IROW and ICOL as declared in the (sub)program from which F11DQF is called. This **must** be the same value as was supplied in the preceding call to F11DNF.

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

6: IROW(LA) – INTEGER array *Input*
7: ICOL(LA) – INTEGER array *Input*
8: IPIVP(N) – INTEGER array *Input*
9: IPIVQ(N) – INTEGER array *Input*
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 F11DNF.

IPIVP and IPIVQ are restored on exit.

12: B(N) – COMPLEX (KIND=nag_wp) 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 (KIND=nag_wp) *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

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

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

Constraint: TOL < 1.0.

15: MAXITN – INTEGER *Input*

On entry: the maximum number of iterations allowed.

Constraint: MAXITN ≥ 1.

16: X(N) – COMPLEX (KIND=nag_wp) 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 (KIND=nag_wp) *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) – COMPLEX (KIND=nag_wp) array *Workspace*
20: LWORK – INTEGER *Input*

On entry: the dimension of the array WORK as declared in the (sub)program from which F11DQF is called.

Constraints:

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

21: 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 $= 1$

On entry, METHOD $\neq$ 'RGMRES', 'CGS', 'BICGSTAB', or 'TFQMR',
or N < 1,
or NNZ < 1,
or NNZ $>$ N^2,
or LA $< 2 \times$ NNZ,
or M < 1 and METHOD $=$ 'RGMRES' or METHOD $=$ 'BICGSTAB',
or M $> \min(\text{N}, 50)$, with METHOD $=$ 'RGMRES',
or M $> \min(\text{N}, 10)$, 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 F11DQF has been preceded by a valid call to F11DNF, 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 F11DQF has been preceded by a valid call to F11DNF 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 $= 6$

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

IFAIL $= 7$ (F11BRF, F11BSF or F11BTF)

A serious error has occurred in an internal call to one of the specified routines. Check all subroutine calls and array sizes. Seek expert help.

7 Accuracy

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

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

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

8 Further Comments

The time taken by F11DQF for each iteration is roughly proportional to the value of NNZC returned from the preceding call to F11DNF.

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 $\bar{A} = M^{-1}A$.

9 Example

This example solves a complex sparse non-Hermitian linear system of equations using the CGS method, with incomplete LU preconditioning.

9.1 Program Text

```
    Program f11dqfe

!     F11DQF Example Program Text

!     Mark 24 Release. NAG Copyright 2012.

!     .. Use Statements ..
      Use nag_library, Only: f11dnf, f11dqf, nag_wp
!     .. Implicit None Statement ..
      Implicit None
!     .. Parameters ..
      Integer, Parameter               :: nin = 5, nout = 6
!     .. Local Scalars ..
      Real (Kind=nag_wp)               :: dtol, rnorm, tol
      Integer                          :: i, ifail, itn, la, lfill, liwork,    &
                                          lwork, m, maxitn, n, nnz, nnzc, npivm
      Character (8)                    :: method
      Character (1)                    :: milu, pstrat
!     .. Local Arrays ..
      Complex (Kind=nag_wp), Allocatable :: a(:), b(:), work(:), x(:)
      Integer, Allocatable             :: icol(:), idiag(:), ipivp(:),         &
                                          ipivq(:), irow(:), istr(:), iwork(:)
!     .. Intrinsic Procedures ..
      Intrinsic                        :: max
!     .. Executable Statements ..
      Write (nout,*) 'F11DQF Example Program Results'
      Write (nout,*)
!     Skip heading in data file
      Read (nin,*)

!     Read algorithmic parameters

      Read (nin,*) n, m
      Read (nin,*) nnz
      la = 2*nnz
      liwork = 7*n + 2
      lwork = max(4*n+m*(m+n+5)+121,8*n+120,2*n*(m+3)+m*(m+2)+120,11*n+120)

      Allocate (a(la),b(n),work(lwork),x(n),icol(la),idiag(n),ipivp(n), &
        ipivq(n),irow(la),istr(n+1),iwork(liwork))
      Read (nin,*) method
      Read (nin,*) lfill, dtol
```

```
      Read (nin,*) pstrat
      Read (nin,*) milu
      Read (nin,*) tol, maxitn

!     Read the matrix A

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

!     Read rhs vector b and initial approximate solution x

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

!     Calculate incomplete LU factorization

!     ifail: behaviour on error exit
!             =0 for hard exit, =1 for quiet-soft, =-1 for noisy-soft
      ifail = 0
      Call f11dnf(n,nnz,a,la,irow,icol,lfill,dtol,pstrat,milu,ipivp,ipivq, &
        istr,idiag,nnzc,npivm,iwork,liwork,ifail)

!     Solve Ax = b using F11DQF

      ifail = 0
      Call f11dqf(method,n,nnz,a,la,irow,icol,ipivp,ipivq,istr,idiag,b,m,tol, &
        maxitn,x,rnorm,itn,work,lwork,ifail)

      Write (nout,99999) itn
      Write (nout,99998) rnorm
      Write (nout,*)

!     Output x

      Write (nout,*) '                   X'
      Write (nout,99997) x(1:n)

99999 Format (1X,'Converged in',I10,' iterations')
99998 Format (1X,'Final residual norm =',1P,E16.3)
99997 Format (1X,'(',1P,E16.4,',',1P,E16.4,')')
    End Program f11dqfe
```

9.2 Program Data

```
F11DQF Example Program Data
  8   4                 N, M
  24                    NNZ
  'CGS'                 METHOD
  0  0.0                LFILL, DTOL
  'C'                   PSTRAT
  'N'                   MILU
  1.0D-10   100         TOL, MAXITN
 ( 2., 1.)   1    1
 (-1., 1.)   1    4
 ( 1.,-3.)   1    8
 ( 4., 7.)   2    1
 (-3., 0.)   2    2
 ( 2., 4.)   2    5
 (-7.,-5.)   3    3
 ( 2., 1.)   3    6
 ( 3., 2.)   4    1
 (-4., 2.)   4    3
 ( 0., 1.)   4    4
 ( 5.,-3.)   4    7
 (-1., 2.)   5    2
 ( 8., 6.)   5    5
 (-3.,-4.)   5    7
 (-6.,-2.)   6    1
 ( 5.,-2.)   6    3
```

```
( 2., 0.)   6    6
( 0.,-5.)   7    3
(-1., 5.)   7    5
( 6., 2.)   7    7
(-1., 4.)   8    2
( 2., 0.)   8    6
( 3., 3.)   8    8      A(I), IROW(I), ICOL(I), I=1,...,NNZ
(  7., 11.)
(  1., 24.)
(-13.,-18.)
(-10.,  3.)
( 23., 14.)
( 17., -7.)
( 15., -3.)
( -3., 20.)             B(I), I=1,...,N
( 0., 0.)
( 0., 0.)
( 0., 0.)
( 0., 0.)
( 0., 0.)
( 0., 0.)
( 0., 0.)
( 0., 0.)               X(I), I=1,...,N
```

9.3 Program Results

```
 F11DQF Example Program Results

 Converged in         4 iterations
 Final residual norm =       2.857E-11

                     X
(      1.0000E+00,      1.0000E+00)
(      2.0000E+00,     -1.0000E+00)
(      3.0000E+00,      1.0000E+00)
(      4.0000E+00,     -1.0000E+00)
(      3.0000E+00,     -1.0000E+00)
(      2.0000E+00,      1.0000E+00)
(      1.0000E+00,     -1.0000E+00)
(      3.0103E-12,      3.0000E+00)
```