C05NBF – 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

C05NBF is an easy-to-use routine to find a solution of a system of nonlinear equations by a modification of the Powell hybrid method.

2 Specification

```
SUBROUTINE C05NBF(FCN, N, X, FVEC, XTOL, WA, LWA, IFAIL)
INTEGER          N, LWA, IFAIL
real             X(N), FVEC(N), XTOL, WA(LWA)
EXTERNAL         FCN
```

3 Description

The system of equations is defined as:

$$f_i(x_1, x_2, \ldots, x_n) = 0, \text{ for } i = 1, 2, \ldots, n.$$

C05NBF is based upon the MINPACK routine HYBRD1 (Moré *et al.* [1]). It chooses the correction at each step as a convex combination of the Newton and scaled gradient directions. Under reasonable conditions this guarantees global convergence for starting points far from the solution and a fast rate of convergence. The Jacobian is updated by the rank-1 method of Broyden. At the starting point the Jacobian is approximated by forward differences, but these are not used again until the rank-1 method fails to produce satisfactory progress. For more details see Powell [2].

4 References

[1] Moré J J, Garbow B S, and Hillstrom K E (1974) User guide for MINPACK–1 *Technical Report ANL–80–74* Argonne National Laboratory

[2] Powell M J D (1970) A hybrid method for nonlinear algebraic equations *Numerical Methods for Nonlinear Algebraic Equations* (ed P Rabinowitz) Gordon and Breach

5 Parameters

1: FCN — SUBROUTINE, supplied by the user. *External Procedure*

FCN must return the values of the functions f_i at a point x.

Its specification is:

```
SUBROUTINE FCN(N, X, FVEC, IFLAG)
INTEGER        N, IFLAG
real           X(N), FVEC(N)
```

1: N — INTEGER *Input*

On entry: the number of equations, n.

2: X(N) — ***real*** array *Input*

On entry: the components of the point x at which the functions must be evaluated.

3: FVEC(N) — ***real*** array *Output*

On exit: the function values $f_i(x)$ (unless IFLAG is set to a negative value by FCN).

4: IFLAG — INTEGER *Input/Output*

On entry: IFLAG > 0.

On exit: in general, IFLAG should not be reset by FCN. If, however, the user wishes to terminate execution (perhaps because some illegal point X has been reached), then IFLAG should be set to a negative integer. This value will be returned through IFAIL.

FCN must be declared as EXTERNAL in the (sub)program from which C05NBF is called. Parameters denoted as *Input* must **not** be changed by this procedure.

2: N — INTEGER *Input*

On entry: the number of equations, n.

Constraint: N > 0.

3: X(N) — ***real*** array *Input/Output*

On entry: an initial guess at the solution vector.

On exit: the final estimate of the solution vector.

4: FVEC(N) — ***real*** array *Output*

On exit: the function values at the final point, X.

5: XTOL — ***real*** *Input*

On entry: the accuracy in X to which the solution is required.

Suggested value: the square root of the ***machine precision***.

Constraint: XTOL $\geq$ 0.0.

6: WA(LWA) — ***real*** array *Workspace*

7: LWA — INTEGER *Input*

On entry: the dimension of the array WA.

Constraint: LWA $\geq$ N $\times$ (3 $\times$ N + 13)/2.

8: 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 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 detected by the routine:

IFAIL < 0

The user has set IFLAG negative in FCN. The value of IFAIL will be the same as the user's setting of IFLAG.

IFAIL = 1

On entry, N $\leq$ 0,

or XTOL < 0.0,

or LWA < N × (3 × N + 13)/2.

IFAIL = 2

There have been at least 200 × (N + 1) evaluations of FCN. Consider restarting the calculation from the final point held in X.

IFAIL = 3

No further improvement in the approximate solution X is possible; XTOL is too small.

IFAIL = 4

The iteration is not making good progress. This failure exit may indicate that the system does not have a zero, or that the solution is very close to the origin (see Section 7). Otherwise, rerunning C05NBF from a different starting point may avoid the region of difficulty.

7 Accuracy

If $\hat{x}$ is the true solution, C05NBF tries to ensure that

$$\|x - \hat{x}\| \leq \mathrm{XTOL} \times \|\hat{x}\|.$$

If this condition is satisfied with XTOL $= 10^{-k}$, then the larger components of x have k significant decimal digits. There is a danger that the smaller components of x may have large relative errors, but the fast rate of convergence of C05NBF usually avoids this possibility.

If XTOL is less than ***machine precision***, and the above test is satisfied with the ***machine precision*** in place of XTOL, then the routine exits with IFAIL = 3.

Note. This convergence test is based purely on relative error, and may not indicate convergence if the solution is very close to the origin.

The test assumes that the functions are reasonably well behaved. If this condition is not satisfied, then C05NBF may incorrectly indicate convergence. The validity of the answer can be checked, for example, by rerunning C05NBF with a tighter tolerance.

8 Further Comments

The time required by C05NBF to solve a given problem depends on n, the behaviour of the functions, the accuracy requested and the starting point. The number of arithmetic operations executed by C05NBF to process each call of FCN is about $11.5 \times n^2$. Unless FCN can be evaluated quickly, the timing of C05NBF will be strongly influenced by the time spent in FCN.

Ideally the problem should be scaled so that at the solution the function values are of comparable magnitude.

9 Example

To determine the values $x_1, \ldots, x_9$ which satisfy the tridiagonal equations:

$$\begin{aligned}(3 - 2x_1)x_1 - 2x_2 &= -1, \\ -x_i - 1 + (3 - 2x_i)x_i - 2x_{i+1} &= -1, \qquad i = 2, 3, \ldots, 8 \\ -x_8 + (3 - 2x_9)x_9 &= -1.\end{aligned}$$

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.

```
*     C05NBF Example Program Text
*     Mark 14 Revised.  NAG Copyright 1989.
*     .. Parameters ..
      INTEGER          N, LWA
      PARAMETER        (N=9,LWA=(N*(3*N+13))/2)
      INTEGER          NOUT
      PARAMETER        (NOUT=6)
*     .. Local Scalars ..
      real             FNORM, TOL
      INTEGER          I, IFAIL, J
*     .. Local Arrays ..
      real             FVEC(N), WA(LWA), X(N)
*     .. External Functions ..
      real             F06EJF, X02AJF
      EXTERNAL         F06EJF, X02AJF
*     .. External Subroutines ..
      EXTERNAL         C05NBF, FCN
*     .. Intrinsic Functions ..
      INTRINSIC        SQRT
*     .. Executable Statements ..
      WRITE (NOUT,*) 'C05NBF Example Program Results'
      WRITE (NOUT,*)
*     The following starting values provide a rough solution.
      DO 20 J = 1, N
         X(J) = -1.0e0
   20 CONTINUE
      TOL = SQRT(X02AJF())
      IFAIL = 1
*
      CALL C05NBF(FCN,N,X,FVEC,TOL,WA,LWA,IFAIL)
*
      IF (IFAIL.EQ.0) THEN
         FNORM = F06EJF(N,FVEC,1)
         WRITE (NOUT,99999) 'Final 2-norm of the residuals =', FNORM
         WRITE (NOUT,*)
         WRITE (NOUT,*) 'Final approximate solution'
         WRITE (NOUT,*)
         WRITE (NOUT,99998) (X(J),J=1,N)
      ELSE
         WRITE (NOUT,99997) 'IFAIL = ', IFAIL
         IF (IFAIL.GT.1) THEN
            WRITE (NOUT,*)
            WRITE (NOUT,*) 'Approximate solution'
            WRITE (NOUT,*)
            WRITE (NOUT,99998) (X(I),I=1,N)
         END IF
      END IF
      STOP
*
99999 FORMAT (1X,A,e12.4)
99998 FORMAT (1X,3F12.4)
99997 FORMAT (1X,A,I2)
      END
*
      SUBROUTINE FCN(N,X,FVEC,IFLAG)
*     .. Parameters ..
      real           ONE, TWO, THREE
      PARAMETER      (ONE=1.0e0,TWO=2.0e0,THREE=3.0e0)
```

```
*     .. Scalar Arguments ..
      INTEGER           IFLAG, N
*     .. Array Arguments ..
      real              FVEC(N), X(N)
*     .. Local Scalars ..
      INTEGER           K
*     .. Executable Statements ..
      DO 20 K = 1, N
         FVEC(K) = (THREE-TWO*X(K))*X(K) + ONE
         IF (K.GT.1) FVEC(K) = FVEC(K) - X(K-1)
         IF (K.LT.N) FVEC(K) = FVEC(K) - TWO*X(K+1)
   20 CONTINUE
      RETURN
      END
```

9.2 Program Data

None.

9.3 Program Results

```
C05NBF Example Program Results

Final 2-norm of the residuals =  0.1193E-07

Final approximate solution

     -0.5707     -0.6816     -0.7017
     -0.7042     -0.7014     -0.6919
     -0.6658     -0.5960     -0.4164
```