C05NCF – 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

C05NCF is a comprehensive routine to find a solution of a system of nonlinear equations by a modification of the Powell hybrid method.

2 Specification

```
 SUBROUTINE C05NCF(FCN, N, X, FVEC, XTOL, MAXFEV, ML, MU, EPSFCN,
1                  DIAG, MODE, FACTOR, NPRINT, NFEV, FJAC, LDFJAC,
2                  R, LR, QTF, W, IFAIL)
 INTEGER           N, MAXFEV, ML, MU, MODE, NPRINT, NFEV, LDFJAC,
1                  LR, IFAIL
 real              X(N), FVEC(N), XTOL, EPSFCN, DIAG(N), FACTOR,
1                  FJAC(LDFJAC,N), R(LR), QTF(N), W(N,4)
 EXTERNAL          FCN
```

3 Description

The system of equations is defined as:

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

C05NCF is based upon the MINPACK routine HYBRD (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: if IFLAG > 0 on entry, FVEC must contain the function values $f_i(x)$ (unless IFLAG is set to a negative value by FCN).

If IFLAG $= 0$ on entry, FVEC must not be changed.

4: IFLAG — INTEGER *Input/Output*

On entry: IFLAG ≥ 0:

if IFLAG $= 0$, X and FVEC are available for printing (see NPRINT below);

if IFLAG > 0, FVEC must be updated

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 C05NCF 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 ≥ 0.0.

6: MAXFEV — INTEGER *Input*

On entry: the maximum number of calls to FCN with IFLAG $\neq 0$. C05NCF will exit with IFAIL $= 2$, if, at the end of an iteration, the number of calls to FCN exceeds MAXFEV.

Suggested value: MAXFEV $= 200 \times (\text{N}+1)$.

Constraint: MAXFEV > 0.

7: ML — INTEGER *Input*

On entry: the number of sub-diagonals within the band of the Jacobian matrix. (If the Jacobian is not banded, or you are unsure, set ML $=$ N $-$ 1.)

Constraint: ML ≥ 0.

8: MU — INTEGER *Input*

On entry: the number of super-diagonals within the band of the Jacobian matrix. (If the Jacobian is not banded, or you are unsure, set MU = N − 1.)

Constraint: MU $\geq$ 0.

9: EPSFCN — ***real*** *Input*

On entry: a rough estimate of the largest relative error in the functions. It is used in determining a suitable step for a forward difference approximation to the Jacobian. If EPSFCN is less than ***machine precision*** then ***machine precision*** is used. Consequently a value of 0.0 will often be suitable.

Suggested value: EPSFCN = 0.0.

10: DIAG(N) — ***real*** array *Input/Output*

On entry: if MODE = 2 (see below), DIAG must contain multiplicative scale factors for the variables.

Constraint: DIAG(i) > 0.0, for $i = 1, 2, \ldots, n$.

On exit: the scale factors actually used (computed internally if MODE $\neq$ 2).

11: MODE — INTEGER *Input*

On entry: indicates whether or not the user has provided scaling factors in DIAG. If MODE = 2 the scaling must have been specified in DIAG. Otherwise, the variables will be scaled internally.

12: FACTOR — ***real*** *Input*

On entry: FACTOR must specify a quantity to be used in determining the initial step bound. In most cases, FACTOR should lie between 0.1 and 100.0. (The step bound is FACTOR × $\|\text{DIAG}\times\text{X}\|_2$ if this is non-zero; otherwise the bound is FACTOR.)

Suggested value: FACTOR = 100.0.

Constraint: FACTOR > 0.0.

13: NPRINT — INTEGER *Input*

On entry: indicates whether special calls to FCN, with IFLAG set to 0, are to be made for printing purposes. If NPRINT $\leq$0, then no calls are made. If NPRINT > 0, then FCN is called at the beginning of the first iteration, every NPRINT iterations thereafter and immediately prior to the return from C05NCF.

14: NFEV — INTEGER *Output*

On exit: the number of calls made to FCN.

15: FJAC(LDFJAC,N) — ***real*** array *Output*

On exit: the orthogonal matrix Q produced by the QR factorization of the final approximate Jacobian.

16: LDFJAC — INTEGER *Input*

On entry: the first dimension of the array FJAC as declared in the (sub)program from which C05NCF is called.

Constraint: LDFJAC $\geq$ N.

17: R(LR) — ***real*** array *Output*

On exit: the upper triangular matrix R produced by the QR factorization of the final approximate Jacobian, stored row-wise.

18: LR — INTEGER *Input*

On entry: the dimension of the array R as declared in the (sub)program from which C05NCF is called.

Constraint: LR $\geq$ N $\times$ (N+1)/2.

19: QTF(N) — ***real*** array *Output*

On exit: the vector $Q^T f$.

20: W(N,4) — ***real*** array *Workspace*

21: 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

This indicates an exit from C05NCF because 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 MAXFEV $\leq$ 0,
or ML $<$ 0,
or MU $<$ 0,
or FACTOR $\leq$ 0.0,
or LDFJAC $<$ N,
or LR $<$ N $\times$ (N+1)/2,
or MODE = 2 and DIAG(i) $\leq$ 0.0 for some i, $i = 1, 2, \ldots$,N.

IFAIL = 2

There have been at least MAXFEV 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, as measured by the improvement from the last 5 Jacobian evaluations.

IFAIL = 5

The iteration is not making good progress, as measured by the improvement from the last 10 iterations.

The values IFAIL = 4 and IFAIL = 5 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 C05NCF from a different starting point may avoid the region of difficulty.

7 Accuracy

If $\hat{x}$ is the true solution and D denotes the diagonal matrix whose entries are defined by the array DIAG, then C05NCF tries to ensure that

$$\|D(x-\hat{x})\|_2 \leq \mathrm{XTOL} \times \|D\hat{x}\|_2.$$

If this condition is satisfied with XTOL $= 10^{-k}$, then the larger components of Dx have k significant decimal digits. There is a danger that the smaller components of Dx may have large relative errors, but the fast rate of convergence of C05NCF usually avoids this possibility.

If XTOL is less than the ***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 C05NCF may incorrectly indicate convergence. The validity of the answer can be checked for example, by rerunning C05NCF with a tighter tolerance.

8 Further Comments

The time required by C05NCF 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 C05NCF to process each call of FCN is about $11.5 \times n^2$. Unless FCN can be evaluated quickly, the timing of C05NCF 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.

The number of function evaluations required to evaluate the Jacobian may be reduced if the user can specify ML and MU.

9 Example

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

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

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.

```
*     C05NCF Example Program Text
*     Mark 14 Revised.  NAG Copyright 1989.
*     .. Parameters ..
      INTEGER          N, LDFJAC, LR
      PARAMETER        (N=9,LDFJAC=N,LR=(N*(N+1))/2)
      INTEGER          NOUT
      PARAMETER        (NOUT=6)
*     .. Local Scalars ..
      real             EPSFCN, FACTOR, FNORM, XTOL
      INTEGER          IFAIL, J, MAXFEV, ML, MODE, MU, NFEV, NPRINT
*     .. Local Arrays ..
      real             DIAG(N), FJAC(LDFJAC,N), FVEC(N), QTF(N), R(LR),
     +                 W(N,4), X(N)
```

```
*     .. External Functions ..
      real             F06EJF, X02AJF
      EXTERNAL         F06EJF, X02AJF
*     .. External Subroutines ..
      EXTERNAL         C05NCF, FCN
*     .. Intrinsic Functions ..
      INTRINSIC        SQRT
*     .. Executable Statements ..
      WRITE (NOUT,*) 'C05NCF Example Program Results'
      WRITE (NOUT,*)
*     The following starting values provide a rough solution.
      DO 20 J = 1, N
         X(J) = -1.0e0
   20 CONTINUE
      XTOL = SQRT(X02AJF())
      DO 40 J = 1, N
         DIAG(J) = 1.0e0
   40 CONTINUE
      MAXFEV = 2000
      ML = 1
      MU = 1
      EPSFCN = 0.0e0
      MODE = 2
      FACTOR = 100.0e0
      NPRINT = 0
      IFAIL = 1
*
      CALL C05NCF(FCN,N,X,FVEC,XTOL,MAXFEV,ML,MU,EPSFCN,DIAG,MODE,
     +            FACTOR,NPRINT,NFEV,FJAC,LDFJAC,R,LR,QTF,W,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,99998) 'Number of function evaluations =', NFEV
         WRITE (NOUT,*)
         WRITE (NOUT,*) 'Final approximate solution'
         WRITE (NOUT,*)
         WRITE (NOUT,99997) (X(J),J=1,N)
      ELSE
         WRITE (NOUT,99996) 'IFAIL = ', IFAIL
         IF (IFAIL.GE.2) THEN
            WRITE (NOUT,*)
            WRITE (NOUT,*) 'Approximate solution'
            WRITE (NOUT,*)
            WRITE (NOUT,99997) (X(J),J=1,N)
         END IF
      END IF
      STOP
*
99999 FORMAT (1X,A,e12.4)
99998 FORMAT (1X,A,I10)
99997 FORMAT (1X,3F12.4)
99996 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 ..
      IF (IFLAG.EQ.0) THEN
*
*        Insert print statements here when NPRINT is positive.
*
         RETURN
      ELSE
         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
      END IF
      RETURN
      END
```

9.2 Program Data

None.

9.3 Program Results

```
 C05NCF Example Program Results

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

 Number of function evaluations =        14

 Final approximate solution

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