C05NDF – 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

C05NDF is a comprehensive reverse communication routine to find a solution of a system of nonlinear equations by a modification of the Powell hybrid method.

2 Specification

```
 SUBROUTINE C05NDF(IREVCM, N, X, FVEC, XTOL, ML, MU, EPSFCN, DIAG,
1                  MODE, FACTOR, FJAC, LDFJAC, R, LR, QTF, W, IFAIL)
 INTEGER           IREVCM, N, ML, MU, MODE, LDFJAC, LR, IFAIL
 real              X(N), FVEC(N), XTOL, EPSFCN, DIAG(N), FACTOR,
1                  FJAC(LDFJAC,N), R(LR), QTF(N), W(N,4)
```

3 Description

The system of equations is defined as:

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

C05NDF 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

Note: this routine uses **reverse communication.** Its use involves an initial entry, intermediate exits and re-entries, and a final exit, as indicated by the **parameter IREVCM**. Between intermediate exits and re-entries, **all parameters other than FVEC must remain unchanged**.

1: IREVCM — INTEGER *Input/Output*

On initial entry: IREVCM must have the value 0.

On intermediate exit: IREVCM specifies what action the user must take before re-entering C05NDF with IREVCM **unchanged**. The value of IREVCM should be interpreted as follows:

IREVCM = 1

indicates the start of a new iteration. No action is required by the user but X and FVEC are available for printing.

IREVCM = 2

indicates that before re-entry to C05NDF, FVEC must contain the function values $f_i(x)$.

On final exit: IREVCM = 0, and the algorithm has terminated.

Constraint: IREVCM = 0, 1 or 2.

2: N — INTEGER *Input*

On initial entry: the number of equations, n.

Constraint: N > 0.

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

On initial entry: an initial guess at the solution vector.

On intermediate exit: X contains the current point.

On final exit: the final estimate of the solution vector.

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

On initial entry: FVEC must be set to the values of the functions computed at the initial point X.

On intermediate re-entry: if IREVCM = 1, FVEC must not be changed. If IREVCM = 2, FVEC must be set to the values of the functions computed at the current point X.

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

5: XTOL — ***real*** *Input*

On initial 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: ML — INTEGER *Input*

On initial 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 $\geq$ 0.

7: MU — INTEGER *Input*

On initial 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.

8: EPSFCN — ***real*** *Input*

On initial entry: the order 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.

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

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

10: MODE — INTEGER *Input*

On initial 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.

11: FACTOR — ***real*** *Input*

On initial entry: 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 $\times \|\text{DIAG} \times \text{X}\|_2$ if this is non-zero; otherwise the bound is FACTOR.)

Suggested value: FACTOR = 100.0.

Constraint: FACTOR > 0.0.

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

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

13: LDFJAC — INTEGER *Input*

On initial entry: the first dimension of the array FJAC as declared in the (sub)program from which C05NDF is called.

Constraint: LDFJAC $\geq$ N.

14: R(LR) — ***real*** array *Output*

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

15: LR — INTEGER *Input*

On initial entry: the dimension of the array R as declared in the (sub)program from which C05NDF is called.

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

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

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

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

18: 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 = 1

On entry, N ≤ 0,
or XTOL < 0.0,
or ML < 0,
or MU < 0,
or FACTOR ≤ 0.0,
or LDFJAC $<$ N,
or LR $<$ N $\times$ (N + 1)/2,
or MODE = 2 and DIAG(i) ≤ 0.0 for some i, $i = 1, 2, \ldots,$ N.

IFAIL = 2

On entry, IREVCM < 0 or IREVCM > 2.

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 C05NDF 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 C05NDF tries to ensure that

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

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 C05NDF 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 that 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 C05NDF may incorrectly indicate convergence. The validity of the answer can be checked for example, by rerunning C05NDF with a tighter tolerance.

8 Further Comments

The time required by C05NDF 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 C05NDF to process the evaluation of functions in the main program in each exit is about $11.5 \times n^2$. The timing of C05NDF will be strongly influenced by the time spent in the evaluation of the functions.

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:

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

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.

```
*     C05NDF Example Program Text
*     Mark 14 Release.  NAG Copyright 1989.
*     .. Parameters ..
      INTEGER          N, LDFJAC, LR
      PARAMETER        (N=9,LDFJAC=N,LR=(N*(N+1))/2)
      INTEGER          NOUT
      PARAMETER        (NOUT=6)
      real             ONE, TWO, THREE
      PARAMETER        (ONE=1.0e0,TWO=2.0e0,THREE=3.0e0)
*     .. Local Scalars ..
      real             EPSFCN, FACTOR, FNORM, XTOL
      INTEGER          ICOUNT, IFAIL, IREVCM, J, K, ML, MODE, MU
*     .. 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         C05NDF
*     .. Intrinsic Functions ..
      INTRINSIC        SQRT
*     .. Executable Statements ..
      WRITE (NOUT,*) 'C05NDF Example Program Results'
*     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
      ML = 1
      MU = 1
      EPSFCN = 0.0e0
      MODE = 2
      FACTOR = 100.0e0
      ICOUNT = 0
      IFAIL = 1
      IREVCM = 0
*
   60 CALL C05NDF(IREVCM,N,X,FVEC,XTOL,ML,MU,EPSFCN,DIAG,MODE,FACTOR,
     +            FJAC,LDFJAC,R,LR,QTF,W,IFAIL)
*
      IF (IREVCM.EQ.1) THEN
         ICOUNT = ICOUNT + 1
*        Insert print statements here to monitor progess if desired.
         GO TO 60
      ELSE IF (IREVCM.EQ.2) THEN
*        Evaluate functions at given point
         DO 80 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)
```

```
   80    CONTINUE
         GO TO 60
      END IF
*
      WRITE (NOUT,*)
      IF (IFAIL.EQ.0) THEN
         FNORM = F06EJF(N,FVEC,1)
         WRITE (NOUT,99999) 'Final 2-norm of the residuals after',
     +     ICOUNT, ' iterations is ', FNORM
         WRITE (NOUT,*)
         WRITE (NOUT,*) 'Final approximate solution'
         WRITE (NOUT,99998) (X(J),J=1,N)
      ELSE
         WRITE (NOUT,99999) 'IFAIL =', IFAIL
         IF (IFAIL.GE.2) THEN
            WRITE (NOUT,*) 'Approximate solution'
            WRITE (NOUT,99998) (X(J),J=1,N)
         END IF
      END IF
      STOP
*
99999 FORMAT (1X,A,I4,A,e12.4)
99998 FORMAT (5X,3F12.4)
      END
```

9.2 Program Data

None.

9.3 Program Results

```
C05NDF Example Program Results

Final 2-norm of the residuals after  11 iterations is   0.1193E-07

Final approximate solution
         -0.5707     -0.6816     -0.7017
         -0.7042     -0.7014     -0.6919
         -0.6658     -0.5960     -0.4164
```