NAG Library Routine Document

C05RDF

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

C05RDF is a comprehensive reverse communication routine that finds a solution of a system of nonlinear equations by a modification of the Powell hybrid method. You must provide the Jacobian.

2 Specification

```
SUBROUTINE C05RDF (IREVCM, N, X, FVEC, FJAC, XTOL, MODE, DIAG, FACTOR,     &
                   R, QTF, IWSAV, RWSAV, IFAIL)

INTEGER            IREVCM, N, MODE, IWSAV(17), IFAIL
REAL (KIND=nag_wp) X(N), FVEC(N), FJAC(N,N), XTOL, DIAG(N), FACTOR,        &
                   R(N*(N+1)/2), QTF(N), RWSAV(4*N+10)
```

3 Description

The system of equations is defined as:

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

C05RDF is based on the MINPACK routine HYBRJ (see Moré *et al.* (1980)). It chooses the correction at each step as a convex combination of the Newton and scaled gradient directions. The Jacobian is updated by the rank-1 method of Broyden. For more details see Powell (1970).

4 References

Moré J J, Garbow B S and Hillstrom K E (1980) User guide for MINPACK-1 *Technical Report ANL-80-74* Argonne National Laboratory

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 Arguments

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 argument **IREVCM**. Between intermediate exits and re-entries, **all arguments other than FVEC and FJAC must remain unchanged**.

1: IREVCM – INTEGER *Input/Output*

On initial entry: must have the value 0.

On intermediate exit: specifies what action you must take before re-entering C05RDF **with** IREVCM **unchanged**. The value of IREVCM should be interpreted as follows:

$\mathrm{IREVCM} = 1$

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

$\mathrm{IREVCM} = 2$

Indicates that before re-entry to C05RDF, FVEC must contain the function values $f_i(x)$.

IREVCM = 3

Indicates that before re-entry to C05RDF, FJAC(i,j) must contain the value of $\frac{\partial f_i}{\partial x_j}$ at the point x, for $i = 1, 2, \ldots, n$ and $j = 1, 2, \ldots, n$.

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

Constraint: IREVCM = 0, 1, 2 or 3.

2: N – INTEGER *Input*

On entry: n, the number of equations.

Constraint: N > 0.

3: X(N) – REAL (KIND=nag_wp) array *Input/Output*

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

On intermediate exit: contains the current point.

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

4: FVEC(N) – REAL (KIND=nag_wp) array *Input/Output*

On initial entry: need not be set.

On intermediate re-entry: if IREVCM $\neq$ 2, 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: FJAC(N, N) – REAL (KIND=nag_wp) array *Input/Output*

On initial entry: need not be set.

On intermediate re-entry: if IREVCM $\neq$ 3, FJAC must not be changed.

If IREVCM = 3, FJAC(i,j) must contain the value of $\frac{\partial f_i}{\partial x_j}$ at the point x, for $i = 1, 2, \ldots, n$ and $j = 1, 2, \ldots, n$.

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

6: XTOL – REAL (KIND=nag_wp) *Input*

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

Suggested value: $\sqrt{\epsilon}$, where ϵ is the ***machine precision*** returned by X02AJF.

Constraint: XTOL $\geq$ 0.0.

7: MODE – INTEGER *Input*

On initial entry: indicates whether or not you have provided scaling factors in DIAG.

If MODE = 2 the scaling must have been supplied in DIAG.

Otherwise, if MODE = 1, the variables will be scaled internally.

Constraint: MODE = 1 or 2.

8: DIAG(N) – REAL (KIND=nag_wp) array *Input/Output*

On initial entry: if MODE = 2, DIAG must contain multiplicative scale factors for the variables.

If MODE $= 1$, DIAG need not be set.

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

On intermediate exit: DIAG must not be changed.

On final exit: the scale factors actually used (computed internally if MODE $= 1$).

9: FACTOR – REAL (KIND=nag_wp) *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 × $\|\mathrm{DIAG} \times \mathrm{X}\|_2$ if this is nonzero; otherwise the bound is FACTOR.)

Suggested value: FACTOR $= 100.0$.

Constraint: FACTOR > 0.0.

10: R$(\mathrm{N} \times (\mathrm{N} + 1)/2)$ – REAL (KIND=nag_wp) array *Input/Output*

On initial entry: need not be set.

On intermediate exit: must not be changed.

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

11: QTF(N) – REAL (KIND=nag_wp) array *Input/Output*

On initial entry: need not be set.

On intermediate exit: must not be changed.

On final exit: the vector $Q^{\mathrm{T}} f$.

12: IWSAV(17) – INTEGER array *Communication Array*
13: RWSAV$(4 \times \mathrm{N} + 10)$ – REAL (KIND=nag_wp) array *Communication Array*

The arrays IWSAV and RWSAV **must not** be altered between calls to C05RDF.

14: IFAIL – INTEGER *Input/Output*

On initial entry: IFAIL must be set to 0, -1 or 1. If you are unfamiliar with this argument you should refer to Section 3.4 in How to Use the NAG Library and its Documentation 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, because for this routine the values of the output arguments may be useful even if IFAIL $\neq 0$ on exit, the recommended value is -1. **When the value -1 or 1 is used it is essential to test the value of IFAIL on exit.**

On final 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 $= 2$

On entry, IREVCM $= \langle$*value*$\rangle$.
Constraint: IREVCM $= 0$, 1, 2 or 3.

IFAIL $= 3$

No further improvement in the solution is possible. XTOL is too small: XTOL $= \langle value \rangle$.

IFAIL $= 4$

The iteration is not making good progress, as measured by the improvement from the last $\langle value \rangle$ Jacobian evaluations. 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 C05RDF from a different starting point may avoid the region of difficulty.

IFAIL $= 5$

The iteration is not making good progress, as measured by the improvement from the last $\langle value \rangle$ iterations. 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 C05RDF from a different starting point may avoid the region of difficulty.

IFAIL $= 11$

On entry, N $= \langle value \rangle$.
Constraint: N > 0.

IFAIL $= 12$

On entry, XTOL $= \langle value \rangle$.
Constraint: XTOL ≥ 0.0.

IFAIL $= 13$

On entry, MODE $= \langle value \rangle$.
Constraint: MODE $= 1$ or 2.

IFAIL $= 14$

On entry, FACTOR $= \langle value \rangle$.
Constraint: FACTOR > 0.0.

IFAIL $= 15$

On entry, MODE $= 2$ and DIAG contained a non-positive element.

IFAIL $= -99$

An unexpected error has been triggered by this routine. Please contact NAG.

See Section 3.9 in How to Use the NAG Library and its Documentation for further information.

IFAIL $= -399$

Your licence key may have expired or may not have been installed correctly.

See Section 3.8 in How to Use the NAG Library and its Documentation for further information.

IFAIL $= -999$

Dynamic memory allocation failed.

See Section 3.7 in How to Use the NAG Library and its Documentation for further information.

7 Accuracy

If $\hat{x}$ is the true solution and D denotes the diagonal matrix whose entries are defined by the array DIAG, then C05RDF 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 C05RDF usually obviates 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 convergence test assumes that the functions and the Jacobian are coded consistently and that the functions are reasonably well behaved. If these conditions are not satisfied, then C05RDF may incorrectly indicate convergence. The coding of the Jacobian can be checked using C05ZDF. If the Jacobian is coded correctly, then the validity of the answer can be checked by rerunning C05RDF with a lower value for XTOL.

8 Parallelism and Performance

C05RDF is threaded by NAG for parallel execution in multithreaded implementations of the NAG Library.

C05RDF makes calls to BLAS and/or LAPACK routines, which may be threaded within the vendor library used by this implementation. Consult the documentation for the vendor library for further information.

Please consult the X06 Chapter Introduction for information on how to control and interrogate the OpenMP environment used within this routine. Please also consult the Users' Note for your implementation for any additional implementation-specific information.

9 Further Comments

The time required by C05RDF 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 C05RDF is approximately $11.5 \times n^2$ to process each evaluation of the functions and approximately $1.3 \times n^3$ to process each evaluation of the Jacobian. The timing of C05RDF is strongly influenced by the time spent evaluating the functions.

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

10 Example

This example determines the values $x_1, \ldots, x_9$ which satisfy the tridiagonal equations:

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

10.1 Program Text

```
    Program c05rdfe

!     C05RDF Example Program Text

!     Mark 26 Release. NAG Copyright 2016.

!     .. Use Statements ..
      Use nag_library, Only: c05rdf, dnrm2, nag_wp, x02ajf
!     .. Implicit None Statement ..
      Implicit None
!     .. Parameters ..
      Integer, Parameter               :: n = 9, nout = 6
!     .. Local Scalars ..
      Real (Kind=nag_wp)               :: factor, fnorm, xtol
      Integer                          :: i, icount, ifail, irevcm, k, mode
!     .. Local Arrays ..
      Real (Kind=nag_wp), Allocatable  :: diag(:), fjac(:,:), fvec(:), qtf(:), &
                                          r(:), rwsav(:), x(:)
      Integer, Allocatable             :: iwsav(:)
!     .. Intrinsic Procedures ..
      Intrinsic                        :: sqrt
!     .. Executable Statements ..
      Write (nout,*) 'C05RDF Example Program Results'

      Allocate (diag(n),fjac(n,n),fvec(n),qtf(n),r(n*(n+                       &
        1)/2),rwsav(4*n+10),iwsav(17),x(n))

!     The following starting values provide a rough solution.

      x(1:n) = -1.0E0_nag_wp

      xtol = sqrt(x02ajf())
      diag(1:n) = 1.0E0_nag_wp
      mode = 2
      factor = 100.0E0_nag_wp
      icount = 0
      irevcm = 0
      ifail = -1

revcomm: Do

        Call c05rdf(irevcm,n,x,fvec,fjac,xtol,mode,diag,factor,r,qtf,iwsav,    &
          rwsav,ifail)

        Select Case (irevcm)
        Case (1)
          icount = icount + 1

!         Insert print statements here to monitor progress if desired.

          Cycle revcomm
        Case (2)

!         Evaluate functions at given point

          fvec(1:n) = (3.0E0_nag_wp-2.0E0_nag_wp*x(1:n))*x(1:n) + 1.0E0_nag_wp
          fvec(2:n) = fvec(2:n) - x(1:(n-1))
          fvec(1:(n-1)) = fvec(1:(n-1)) - 2.0E0_nag_wp*x(2:n)
          Cycle revcomm
        Case (3)

!         Evaluate Jacobian at current point

          fjac(1:n,1:n) = 0.0E0_nag_wp

          Do k = 1, n
            fjac(k,k) = 3.0E0_nag_wp - 4.0E0_nag_wp*x(k)

            If (k/=1) Then
              fjac(k,k-1) = -1.0E0_nag_wp
            End If

            If (k/=n) Then
              fjac(k,k+1) = -2.0E0_nag_wp
            End If

          End Do

          Cycle revcomm
        Case Default
          Exit revcomm
        End Select

      End Do revcomm
      If (ifail==0 .Or. ifail==3 .Or. ifail==4 .Or. ifail==5) Then
        If (ifail==0) Then
!         The NAG name equivalent of dnrm2 is f06ejf
          fnorm = dnrm2(n,fvec,1)
          Write (nout,*)
          Write (nout,99999) 'Final 2-norm of the residuals after', icount,    &
            ' iterations is ', fnorm
          Write (nout,*)
          Write (nout,*) 'Final approximate solution'
        Else
          Write (nout,*)
          Write (nout,*) 'Approximate solution'
        End If
        Write (nout,*)
        Write (nout,99998)(x(i),i=1,n)
      End If

99999 Format (1X,A,I4,A,E12.4)
99998 Format (5X,3F12.4)
    End Program c05rdfe
```

10.2 Program Data

None.

10.3 Program Results

```
 C05RDF 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
```