

NAG Library Routine Document

C05QDF

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

C05QDF is a comprehensive reverse communication routine that finds a solution of a system of nonlinear equations by a modification of the Powell hybrid method.

2 Specification

```
SUBROUTINE C05QDF (IREVCM, N, X, FVEC, XTOL, ML, MU, EPSFCN, MODE, DIAG,      &
                  FACTOR, FJAC, R, QTF, IWSAV, RWSAV, IFAIL)
INTEGER          IREVCM, N, ML, MU, MODE, IWSAV(17), IFAIL
REAL (KIND=nag_wp) X(N), FVEC(N), XTOL, EPSFCN, DIAG(N), FACTOR,      &
                  FJAC(N,N), R(N*(N+1)/2), QTF(N), RWSAV(4*N+10)
```

3 Description

The system of equations is defined as:

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

C05QDF is based on the MINPACK routine HYBRD (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. 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 (1970).

4 References

Moré J J, Garbow B S and Hillstrome 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 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: must have the value 0.

On intermediate exit: specifies what action you must take before re-entering C05QDF 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 you, but X and FVEC are available for printing.

IREVCM = 2

Indicates that before re-entry to C05QDF, 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 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 = 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 (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: $XTOL \geq 0.0$.
- 6: ML – INTEGER *Input*
On initial entry: the number of subdiagonals 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 superdiagonals 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 (KIND=nag_wp) *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** (returned by X02AJF) then **machine precision** is used. Consequently a value of 0.0 will often be suitable.
Suggested value: $EPSFCN = 0.0$.
- 9: 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 .

- 10: DIAG(N) – REAL (KIND=nag_wp) array *Input/Output*
On 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, $\text{DIAG}(i) > 0.0$, for $i = 1, 2, \dots, n$.
On exit: the scale factors actually used (computed internally if MODE = 1).
- 11: 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 $\text{FACTOR} \times \|\text{DIAG} \times \mathbf{X}\|_2$ if this is nonzero; otherwise the bound is FACTOR.)
Suggested value: FACTOR = 100.0.
Constraint: FACTOR > 0.0.
- 12: FJAC(N,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 orthogonal matrix Q produced by the QR factorization of the final approximate Jacobian.
- 13: R(N × (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.
- 14: 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 $Q^T f$.
- 15: IWSAV(17) – INTEGER array *Communication Array*
 16: RWSAV(4 × N + 10) – REAL (KIND=nag_wp) array *Communication Array*
 The arrays IWSAV and RWSAV **must not** be altered between calls to C05QDF.
- 17: IFAIL – INTEGER *Input/Output*
On initial 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, because for this routine the values of the output parameters may be useful even if $\text{IFAIL} \neq 0$ 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: $IREVCM = 0, 1$ or 2 .

$IFAIL = 3$

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.

$IFAIL = 5$

The iteration is not making good progress, as measured by the improvement from the last $\langle value \rangle$ iterations.

$IFAIL = 11$

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

$IFAIL = 12$

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

$IFAIL = 13$

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.

$IFAIL = 16$

On entry, $ML = \langle value \rangle$.
Constraint: $ML \geq 0$.

$IFAIL = 17$

On entry, $MU = \langle value \rangle$.
Constraint: $MU \geq 0$.

$IFAIL = -99$

An unexpected error has been triggered by this routine. Please contact NAG.
See Section 3.8 in the Essential Introduction for further information.

IFAIL = -399

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

See Section 3.7 in the Essential Introduction for further information.

IFAIL = -999

Dynamic memory allocation failed.

See Section 3.6 in the Essential Introduction for further information.

A value of IFAIL = 4 or 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 C05QDF 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 C05QDF tries to ensure that

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

If this condition is satisfied with $\text{XTOL} = 10^{-k}$, 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 C05QDF 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 are reasonably well behaved. If this condition is not satisfied, then C05QDF may incorrectly indicate convergence. The validity of the answer can be checked, for example, by rerunning C05QDF with a lower value for XTOL.

8 Parallelism and Performance

C05QDF is threaded by NAG for parallel execution in multithreaded implementations of the NAG Library.

C05QDF 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 C05QDF 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 C05QDF to process the evaluation of functions in the main program in each exit is approximately $11.5 \times n^2$. The timing of C05QDF 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.

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

10 Example

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

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

10.1 Program Text

```

Program c05qdfc
!      C05QDF Example Program Text
!      Mark 25 Release. NAG Copyright 2014.
!
!      .. Use Statements ..
Use nag_library, Only: c05qdf, dnrm2, nag_wp, x02ajf
!      .. Implicit None Statement ..
Implicit None
!      .. Parameters ..
Integer, Parameter          :: n = 9, nout = 6
!      .. Local Scalars ..
Real (Kind=nag_wp)         :: epsfcn, factor, fnorm, xtol
Integer                    :: i, icount, ifail, irevcn, ml, mode, mu
!      .. 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,*) 'C05QDF 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
ml = 1
mu = 1
epsfcn = 0.0E0_nag_wp
mode = 2
factor = 100.0E0_nag_wp
icount = 0
irevcn = 0
ifail = -1

revcomm: Do

    Call c05qdf(irevcn,n,x,fvec,xtol,ml,mu,epsfcn,mode,diag,factor,fjac,r, &
qtf,iwsav,rwsav,ifail)

    Select Case (irevcn)
    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 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 dnrn2 is f06ejf
        fnorm = dnrn2(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 c05qdf

```

10.2 Program Data

None.

10.3 Program Results

C05QDF 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
