NEQNF

Solves a system of nonlinear equations using a modified Powell hybrid algorithm and a finite-difference approximation to the Jacobian.

Required Arguments

FCN — User-supplied SUBROUTINE to evaluate the system of equations to be solved. The usage is CALL FCN (X, F, N), where

X – The point at which the functions are evaluated. (Input)
X should not be changed by FCN.

F – The computed function values at the point X. (Output)

FCN must be declared EXTERNAL in the calling program.

N — Length of X and F. (Input)

X — A vector of length N. (Output)
X contains the best estimate of the root found by NEQNF.

Optional Arguments

ERRREL — Stopping criterion. (Input)
The root is accepted if the relative error between two successive approximations to this root is less than ERRREL.
Default: ERRREL = 1.e-4 for single precision and 1.d-8 for double precision.

N – The number of equations to be solved and the number of unknowns. (Input)
Default: N = size (X,1).

ITMAX — The maximum allowable number of iterations. (Input)
The maximum number of calls to FCN is ITMAX * (N + 1). Suggested value ITMAX = 200.
Default: ITMAX = 200.

XGUESS — A vector of length N. (Input)
XGUESS contains the initial estimate of the root.
Default: XGUESS = 0.0.

FNORM — A scalar that has the value $F(1)^2 + \ldots + F(N)^2$ at the point X. (Output)

FORTRAN 90 Interface

Generic: CALL NEQNF (FCN, X [,…])

Specific: The specific interface names are S_NEQNF and D_NEQNF.

FORTRAN 77 Interface

Single: CALL NEQNF (FCN, ERRREL, N, ITMAX, XGUESS, X, FNORM)

Double: The double precision name is DNEQNF.

Description

Routine NEQNF is based on the MINPACK subroutine HYBRD1, which uses a modification of M.J.D. Powell's hybrid algorithm. This algorithm is a variation of Newton's method, which uses a finite-difference approximation to the Jacobian and takes precautions to avoi large step sizes or increasing residuals. For further description, see More et al. (1980).

Since a finite-difference method is used to estimate the Jacobian, for single precision calculation, the Jacobian may be so incorrect that algorithm terminates far from a root. In such cases, high precision arithmetic is recommended. Also, whenever the exact Jacobian can b easily provided, IMSL routine NEQNJ should be used instead.

Comments

1. Workspace may be explicitly provided, if desired, by use of N2QNF/DN2QNF. The reference is:

```
CALL N2QNF (FCN, ERRREL, N, ITMAX, XGUESS, X, FNORM, FVEC, FJAC, R, QTF, WK)
```

The additional arguments are as follows:

FVEC — A vector of length N. FVEC contains the functions evaluated at the point X.

FJAC — An N by N matrix. FJAC contains the orthogonal matrix Q produced by the QR factorization of the final approximate Jacobian.

R — A vector of length N * (N + 1)/2. R contains the upper triangular matrix produced by the QR factorization of the final approximate Jacobian. R is stored row-wise.

QTF — A vector of length N. QTF contains the vector TRANS(Q) * FVEC.

WK — A work vector of length 5 * N.

2. Informational errors

Type	Code	
4	1	The number of calls to FCN has exceeded ITMAX * (N + 1). A new initial guess may be tried.
4	2	ERRREL is too small. No further improvement in the approximate solution is possible.
4	3	The iteration has not made good progress. A new initial guess may be tried.

Example

The following 3 ´ 3 system of nonlinear equations

$$f_1(x) = x_1 + e^{x_1-1} + (x_2 + x_3)^2 - 27 = 0$$

$$f_2(x) = e^{x_2-2} / x_1 + x_3^2 - 10 = 0$$

$$f_3(x) = x_3 + \sin(x_2 - 2) + x_2^2 - 7 = 0$$

is solved with the initial guess (4.0, 4.0, 4.0).

```
      USE NEQNF_INT
      USE UMACH_INT

      IMPLICIT   NONE
!                                 Declare variables
      INTEGER    N
      PARAMETER  (N=3)
!
      INTEGER    K, NOUT
      REAL       FNORM, X(N), XGUESS(N)
      EXTERNAL   FCN
!                                 Set values of initial guess
!                                 XGUESS = (  4.0  4.0  4.0 )
!
      DATA XGUESS/4.0, 4.0, 4.0/
!
!
      CALL UMACH (2, NOUT)
!                                 Find the solution
      CALL NEQNF (FCN, X, xguess=xguess, fnorm=fnorm)
!                                 Output
      WRITE (NOUT,99999) (X(K),K=1,N), FNORM
99999 FORMAT ('  The solution to the system is', /, '  X = (', 3F5.1, &
            ')', /, '  with FNORM =', F5.4, //)
!
      END
!                                 User-defined subroutine
      SUBROUTINE FCN (X, F, N)
      INTEGER    N
      REAL       X(N), F(N)
!
      REAL       EXP, SIN
      INTRINSIC  EXP, SIN
!
      F(1) = X(1) + EXP(X(1)-1.0) + (X(2)+X(3))*(X(2)+X(3)) - 27.0
      F(2) = EXP(X(2)-2.0)/X(1) + X(3)*X(3) - 10.0
      F(3) = X(3) + SIN(X(2)-2.0) + X(2)*X(2) - 7.0
      RETURN
      END
```

Output

```
The solution to the system is
X = (  1.0  2.0  3.0)
with FNORM =.0000
```

Visual Numerics, Inc.
Visual Numerics - Developers of IMSL and PV-WAVE
http://www.vni.com/
PHONE: 713.784.3131
FAX:713.781.9260