

CHAPTER 4  
Documentation

This chapter contains the double precision version of the MINPACK-1 documentation; both single and double precision versions of the documentation are available in machine-readable form with the MINPACK-1 package. The documentation appears in the following order:

Systems of nonlinear equations

HYBRD1, HYBRD, HYBRJ1, HYBRJ

Nonlinear least squares problems

LMDF1, LMDIF, LMDER1, LMDER, LMSTR1, LMSTR

Derivative checking

CHKDER



## Documentation for MINPACK subroutine HYBRD1

Double precision version

Argonne National Laboratory

Burton S. Garbow, Kenneth E. Hillstrom, Jorge J. More

March 1980

## 1. Purpose.

The purpose of HYBRD1 is to find a zero of a system of  $N$  non-linear functions in  $N$  variables by a modification of the Powell hybrid method. This is done by using the more general nonlinear equation solver HYBRD. The user must provide a subroutine which calculates the functions. The Jacobian is then calculated by a forward-difference approximation.

## 2. Subroutine and type statements.

```
SUBROUTINE HYBRD1(FCN,N,X,FVEC,TOL,INFO,WA,LWA)
INTEGER N,INFO,LWA
DOUBLE PRECISION TOL
DOUBLE PRECISION X(N),FVEC(N),WA(LWA)
EXTERNAL FCN
```

## 3. Parameters.

Parameters designated as input parameters must be specified on entry to HYBRD1 and are not changed on exit, while parameters designated as output parameters need not be specified on entry and are set to appropriate values on exit from HYBRD1.

FCN is the name of the user-supplied subroutine which calculates the functions. FCN must be declared in an EXTERNAL statement in the user calling program, and should be written as follows.

```
SUBROUTINE FCN(N,X,FVEC,IFLAG)
INTEGER N,IFLAG
DOUBLE PRECISION X(N),FVEC(N)
-----
CALCULATE THE FUNCTIONS AT X AND
RETURN THIS VECTOR IN FVEC.
-----
RETURN
END
```

The value of IFLAG should not be changed by FCN unless the user wants to terminate execution of HYBRD1. In this case set IFLAG to a negative integer.

N is a positive integer input variable set to the number of functions and variables.

X is an array of length N. On input X must contain an initial estimate of the solution vector. On output X contains the final estimate of the solution vector.

FVEC is an output array of length N which contains the functions evaluated at the output X.

TOL is a nonnegative input variable. Termination occurs when the algorithm estimates that the relative error between X and the solution is at most TOL. Section 4 contains more details about TOL.

INFO is an integer output variable. If the user has terminated execution, INFO is set to the (negative) value of IFLAG. See description of FCN. Otherwise, INFO is set as follows.

INFO = 0 Improper input parameters.

INFO = 1 Algorithm estimates that the relative error between X and the solution is at most TOL.

INFO = 2 Number of calls to FCN has reached or exceeded  $200*(N+1)$ .

INFO = 3 TOL is too small. No further improvement in the approximate solution X is possible.

INFO = 4 Iteration is not making good progress.

Sections 4 and 5 contain more details about INFO.

WA is a work array of length LWA.

LWA is a positive integer input variable not less than  $(N*(3*N+13))/2$ .

#### 4. Successful completion.

The accuracy of HYBRD1 is controlled by the convergence parameter TOL. This parameter is used in a test which makes a comparison between the approximation X and a solution XSOL. HYBRD1 terminates when the test is satisfied. If TOL is less than the machine precision (as defined by the MINPACK function DPMPAR(1)), then HYBRD1 only attempts to satisfy the test defined by the machine precision. Further progress is not usually possible. Unless high precision solutions are required, the recommended value for TOL is the square root of the machine precision.

The test assumes that the functions are reasonably well behaved.

If this condition is not satisfied, then HYBRD1 may incorrectly indicate convergence. The validity of the answer can be checked, for example, by rerunning HYBRD1 with a tighter tolerance.

Convergence test. If  $ENORM(Z)$  denotes the Euclidean norm of a vector  $Z$ , then this test attempts to guarantee that

$$ENORM(X-XSOL) \leq TOL * ENORM(XSOL).$$

If this condition is satisfied with  $TOL = 10^{(-K)}$ , then the larger components of  $X$  have  $K$  significant decimal digits and INFO is set to 1. There is a danger that the smaller components of  $X$  may have large relative errors, but the fast rate of convergence of HYBRD1 usually avoids this possibility.

#### 5. Unsuccessful completion.

Unsuccessful termination of HYBRD1 can be due to improper input parameters, arithmetic interrupts, an excessive number of function evaluations, errors in the functions, or lack of good progress.

Improper input parameters. INFO is set to 0 if  $N \leq 0$ , or  $TOL \leq 0$ , or  $LWA \leq (N*(3*N+13))/2$ .

Arithmetic interrupts. If these interrupts occur in the FCN subroutine during an early stage of the computation, they may be caused by an unacceptable choice of  $X$  by HYBRD1. In this case, it may be possible to remedy the situation by not evaluating the functions here, but instead setting the components of FVEC to numbers that exceed those in the initial FVEC, thereby indirectly reducing the step length. The step length can be more directly controlled by using instead HYBRD, which includes in its calling sequence the step-length-governing parameter FACTOR.

Excessive number of function evaluations. If the number of calls to FCN reaches  $200*(N+1)$ , then this indicates that the routine is converging very slowly as measured by the progress of FVEC, and INFO is set to 2. This situation should be unusual because, as indicated below, lack of good progress is usually diagnosed earlier by HYBRD1, causing termination with INFO = 4.

Errors in the functions. The choice of step length in the forward-difference approximation to the Jacobian assumes that the relative errors in the functions are of the order of the machine precision. If this is not the case, HYBRD1 may fail (usually with INFO = 4). The user should then use HYBRD instead, or one of the programs which require the analytic Jacobian (HYBRJ1 and HYBRJ).

Lack of good progress. HYBRD1 searches for a zero of the system by minimizing the sum of the squares of the functions. In so doing, it can become trapped in a region where the minimum does not correspond to a zero of the system and, in this situation, the iteration eventually fails to make good progress. In particular, this will happen if the system does not have a zero. If the system has a zero, rerunning HYBRD1 from a different starting point may be helpful.

## 6. Characteristics of the algorithm.

HYBRD1 is a modification of the Powell hybrid method. Two of its main characteristics involve the choice of the correction as a convex combination of the Newton and scaled gradient directions, and the updating of the Jacobian by the rank-1 method of Broyden. The choice of the correction guarantees (under reasonable conditions) global convergence for starting points far from the solution and a fast rate of convergence. The Jacobian is approximated by forward differences at the starting point, but forward differences are not used again until the rank-1 method fails to produce satisfactory progress.

Timing. The time required by HYBRD1 to solve a given problem depends on  $N$ , the behavior of the functions, the accuracy requested, and the starting point. The number of arithmetic operations needed by HYBRD1 is about  $11.5*(N^2)$  to process each call to FCN. Unless FCN can be evaluated quickly, the timing of HYBRD1 will be strongly influenced by the time spent in FCN.

- Storage. HYBRD1 requires  $(3*N^2 + 17*N)/2$  double precision storage locations, in addition to the storage required by the program. There are no internally declared storage arrays.

## 7. Subprograms required.

USER-supplied ..... FCN

MINPACK-supplied ... DOGLEG, DPMPAR, ENORM, FDJAC1, HYBRD,  
QFORM, QRFAC, R1MPYQ, R1UPDT

FORTTRAN-supplied ... DABS, DMAX1, DMIN1, DSQRT, MINO, MOD

## 8. References.

M. J. D. Powell, A Hybrid Method for Nonlinear Equations. Numerical Methods for Nonlinear Algebraic Equations, P. Rabinowitz, editor. Gordon and Breach, 1970.

## 9. Example.

The problem is to determine the values of  $x(1)$ ,  $x(2)$ , ...,  $x(9)$ , which solve the system of tridiagonal equations

$$\begin{aligned} (3-2*x(1))*x(1) & & -2*x(2) & & & = -1 \\ -x(i-1) + (3-2*x(i))*x(i) & & & & -2*x(i+1) & = -1, \quad i=2-8 \\ & & -x(8) + (3-2*x(9))*x(9) & & & = -1 \end{aligned}$$

```

C *****
C
C DRIVER FOR HYBRD1 EXAMPLE.
C DOUBLE PRECISION VERSION
C
C *****
C INTEGER J,N,INFO,LWA,NWRITE
C DOUBLE PRECISION TOL, FNORM
C DOUBLE PRECISION X(9), FVEC(9), WA(180)
C DOUBLE PRECISION ENORM, DPMPAR
C EXTERNAL FCN
C
C LOGICAL OUTPUT UNIT IS ASSUMED TO BE NUMBER 6.
C
C DATA NWRITE /6/
C
C N = 9
C
C THE FOLLOWING STARTING VALUES PROVIDE A ROUGH SOLUTION.
C
C DO 10 J = 1, 9
C   X(J) = -1.DO
10 CONTINUE
C
C LWA = 180
C
C SET TOL TO THE SQUARE ROOT OF THE MACHINE PRECISION.
C UNLESS HIGH PRECISION SOLUTIONS ARE REQUIRED,
C THIS IS THE RECOMMENDED SETTING.
C
C TOL = DSQRT(DPMPAR(1))
C
C CALL HYBRD1(FCN,N,X,FVEC,TOL,INFO,WA,LWA)
C ENORM = ENORM(N,FVEC)
C WRITE (NWRITE,1000) FNORM,INFO,(X(J),J=1,N)
C STOP
1000 FORMAT (5X,31H FINAL L2 NORM OF THE RESIDUALS,D15.7 //
*          5X,15H EXIT PARAMETER,16X,I10 //
*          5X,27H FINAL APPROXIMATE SOLUTION // (5X,3D15.7))
C
C LAST CARD OF DRIVER FOR HYBRD1 EXAMPLE.
C
C END
C SUBROUTINE FCN(N,X,FVEC,IFLAG)
C INTEGER N,IFLAG
C DOUBLE PRECISION X(N),FVEC(N)
C

```

```

C      SUBROUTINE FCN FOR HYBRD1 EXAMPLE.
C
      INTEGER K
      DOUBLE PRECISION ONE, TEMP, TEMP1, TEMP2, THREE, TWO, ZERO
      DATA ZERO, ONE, TWO, THREE /0.DO, 1.DO, 2.DO, 3.DO/
C
      DO 10 K = 1, N
         TEMP = (THREE - TWO*X(K))*X(K)
         TEMP1 = ZERO
         IF (K .NE. 1) TEMP1 = X(K-1)
         TEMP2 = ZERO
         IF (K .NE. N) TEMP2 = X(K+1)
         FVEC(K) = TEMP - TEMP1 - TWO*TEMP2 + ONE
10      CONTINUE
      RETURN
C
C      LAST CARD OF SUBROUTINE FCN.
C
      END

```

Results obtained with different compilers or machines  
may be slightly different.

FINAL L2 NORM OF THE RESIDUALS 0.1192636D-07

EXIT PARAMETER 1

FINAL APPROXIMATE SOLUTION

```

-0.5706545D+00 -0.6816283D+00 -0.7017325D+00
-0.7042129D+00 -0.7013690D+00 -0.6918656D+00
-0.6657920D+00 -0.5960342D+00 -0.4164121D+00

```

## Documentation for MINPACK subroutine HYBRD

Double precision version

Argonne National Laboratory

Burton S. Garbow, Kenneth E. Hillstom, Jorge J. More

March 1980

## 1. Purpose.

The purpose of HYBRD is to find a zero of a system of  $N$  non-linear functions in  $N$  variables by a modification of the Powell hybrid method. The user must provide a subroutine which calculates the functions. The Jacobian is then calculated by a forward-difference approximation.

## 2. Subroutine and type statements.

```

SUBROUTINE HYBRD(FCN,N,X,FVEC,XTOL,MAXFEV,ML,MU,EPSFCN,DIAG,
*             MODE,FACTOR,NPRINT,INFO,NFEV,FJAC,LDFJAC,
*             R,LR,QTF,WA1,WA2,WA3,WA4)
INTEGER N,MAXFEV,ML,MU,MODE,NPRINT,INFO,NFEV,LDFJAC,LR
DOUBLE PRECISION XTOL,EPSFCN,FACTOR
DOUBLE PRECISION X(N),FVEC(N),DIAG(N),FJAC(LDFJAC,N),R(LR),QTF(N),
*             WA1(N),WA2(N),WA3(N),WA4(N)
EXTERNAL FCN

```

## 3. Parameters.

Parameters designated as input parameters must be specified on entry to HYBRD and are not changed on exit, while parameters designated as output parameters need not be specified on entry and are set to appropriate values on exit from HYBRD.

FCN is the name of the user-supplied subroutine which calculates the functions. FCN must be declared in an EXTERNAL statement in the user calling program, and should be written as follows.

```

SUBROUTINE FCN(N,X,FVEC,IFLAG)
INTEGER N,IFLAG
DOUBLE PRECISION X(N),FVEC(N)
-----
CALCULATE THE FUNCTIONS AT X AND
RETURN THIS VECTOR IN FVEC.
-----
RETURN
END

```

The value of IFLAG should not be changed by FCN unless the

- user wants to terminate execution of HYBRD. In this case set IFLAG to a negative integer.
- N is a positive integer input variable set to the number of functions and variables.
- X is an array of length N. On input X must contain an initial estimate of the solution vector. On output X contains the final estimate of the solution vector.
- FVEC is an output array of length N which contains the functions evaluated at the output X.
- XTOL is a nonnegative input variable. Termination occurs when the relative error between two consecutive iterates is at most XTOL. Therefore, XTOL measures the relative error desired in the approximate solution. Section 4 contains more details about XTOL.
- MAXFEV is a positive integer input variable. Termination occurs when the number of calls to FCN is at least MAXFEV by the end of an iteration.
- ML is a nonnegative integer input variable which specifies the number of subdiagonals within the band of the Jacobian matrix. If the Jacobian is not banded, set ML to at least  $N - 1$ .
- MU is a nonnegative integer input variable which specifies the number of superdiagonals within the band of the Jacobian matrix. If the Jacobian is not banded, set MU to at least  $N - 1$ .
- EPSFCN is an input variable used in determining a suitable step for the forward-difference approximation. This approximation assumes that the relative errors in the functions are of the order of EPSFCN. If EPSFCN is less than the machine precision, it is assumed that the relative errors in the functions are of the order of the machine precision.
- DIAG is an array of length N. If MODE = 1 (see below), DIAG is internally set. If MODE = 2, DIAG must contain positive entries that serve as multiplicative scale factors for the variables.
- MODE is an integer input variable. If MODE = 1, the variables will be scaled internally. If MODE = 2, the scaling is specified by the input DIAG. Other values of MODE are equivalent to MODE = 1.
- FACTOR is a positive input variable used in determining the initial step bound. This bound is set to the product of FACTOR and the Euclidean norm of  $DIAG * X$  if nonzero, or else to FACTOR itself. In most cases FACTOR should lie in the interval (.1,100.). 100. is a generally recommended value.

NPRINT is an integer input variable that enables controlled printing of iterates if it is positive. In this case, FCN is called with IFLAG = 0 at the beginning of the first iteration and every NPRINT iterations thereafter and immediately prior to return, with X and FVEC available for printing. If NPRINT is not positive, no special calls of FCN with IFLAG = 0 are made.

INFO is an integer output variable. If the user has terminated execution, INFO is set to the (negative) value of IFLAG. See description of FCN. Otherwise, INFO is set as follows.

INFO = 0 Improper input parameters.

INFO = 1 Relative error between two consecutive iterates is at most XTOL.

INFO = 2 Number of calls to FCN has reached or exceeded MAXFEV.

INFO = 3 XTOL is too small. No further improvement in the approximate solution X is possible.

INFO = 4 Iteration is not making good progress, as measured by the improvement from the last five Jacobian evaluations.

INFO = 5 Iteration is not making good progress, as measured by the improvement from the last ten iterations.

Sections 4 and 5 contain more details about INFO.

NFEV is an integer output variable set to the number of calls to FCN.

FJAC is an output N by N array which contains the orthogonal matrix Q produced by the QR factorization of the final approximate Jacobian.

LDEFJAC is a positive integer input variable not less than N which specifies the leading dimension of the array FJAC.

R is an output array of length LR which contains the upper triangular matrix produced by the QR factorization of the final approximate Jacobian, stored rowwise.

LR is a positive integer input variable not less than  $(N*(N+1))/2$ .

QTF is an output array of length N which contains the vector  $(Q \text{ transpose}) * FVEC$ .

WA1, WA2, WA3, and WA4 are work arrays of length N.

#### 4. Successful completion.

The accuracy of HYBRD is controlled by the convergence parameter XTOL. This parameter is used in a test which makes a comparison between the approximation X and a solution XSOL. HYBRD terminates when the test is satisfied. If the convergence parameter is less than the machine precision (as defined by the MINPACK function DPMPAR(1)), then HYBRD only attempts to satisfy the test defined by the machine precision. Further progress is not usually possible.

The test assumes that the functions are reasonably well behaved. If this condition is not satisfied, then HYBRD may incorrectly indicate convergence. The validity of the answer can be checked, for example, by rerunning HYBRD with a tighter tolerance.

Convergence test. If ENORM(Z) denotes the Euclidean norm of a vector Z and D is the diagonal matrix whose entries are defined by the array DIAG, then this test attempts to guarantee that

$$\text{ENORM}(D*(X-XSOL)) \leq \text{XTOL} * \text{ENORM}(D*XSOL).$$

If this condition is satisfied with  $\text{XTOL} = 10^{**(-K)}$ , then the larger components of D\*X have K significant decimal digits and INFO is set to 1. There is a danger that the smaller components of D\*X may have large relative errors, but the fast rate of convergence of HYBRD usually avoids this possibility. Unless high precision solutions are required, the recommended value for XTOL is the square root of the machine precision.

#### 5. Unsuccessful completion.

Unsuccessful termination of HYBRD can be due to improper input parameters, arithmetic interrupts, an excessive number of function evaluations, or lack of good progress.

Improper input parameters. INFO is set to 0 if N  $\leq$  0, or XTOL  $<$  0.D0, or MAXFEV  $\leq$  0, or ML  $<$  0, or MU  $<$  0, or FACTOR  $\leq$  0.D0, or LDFJAC  $<$  N, or LR  $<$  (N\*(N+1))/2.

Arithmetic interrupts. If these interrupts occur in the FCN subroutine during an early stage of the computation, they may be caused by an unacceptable choice of X by HYBRD. In this case, it may be possible to remedy the situation by rerunning HYBRD with a smaller value of FACTOR.

Excessive number of function evaluations. A reasonable value for MAXFEV is 200\*(N+1). If the number of calls to FCN reaches MAXFEV, then this indicates that the routine is converging very slowly as measured by the progress of FVEC, and

INFO is set to 2. This situation should be unusual because, as indicated below, lack of good progress is usually diagnosed earlier by HYBRD, causing termination with INFO = 4 or INFO = 5.

Lack of good progress. HYBRD searches for a zero of the system by minimizing the sum of the squares of the functions. In so doing, it can become trapped in a region where the minimum does not correspond to a zero of the system and, in this situation, the iteration eventually fails to make good progress. In particular, this will happen if the system does not have a zero. If the system has a zero, rerunning HYBRD from a different starting point may be helpful.

## 6. Characteristics of the algorithm.

HYBRD is a modification of the Powell hybrid method. Two of its main characteristics involve the choice of the correction as a convex combination of the Newton and scaled gradient directions, and the updating of the Jacobian by the rank-1 method of Broyden. The choice of the correction guarantees (under reasonable conditions) global convergence for starting points far from the solution and a fast rate of convergence. The Jacobian is approximated by forward differences at the starting point, but forward differences are not used again until the rank-1 method fails to produce satisfactory progress.

Timing. The time required by HYBRD to solve a given problem depends on  $N$ , the behavior of the functions, the accuracy requested, and the starting point. The number of arithmetic operations needed by HYBRD is about  $11.5 \cdot (N^2)$  to process each call to FCN. Unless FCN can be evaluated quickly, the timing of HYBRD will be strongly influenced by the time spent in FCN.

Storage. HYBRD requires  $(3 \cdot N^2 + 17 \cdot N)/2$  double precision storage locations, in addition to the storage required by the program. There are no internally declared storage arrays.

## 7. Subprograms required.

USER-supplied ..... FCN

MINPACK-supplied ... DOGLEG, DPMPAR, ENORM, FDJAC1,  
QFORM, QRFAC, R1MPYQ, R1UPDT

FORTTRAN-supplied ... DABS, DMAX1, DMIN1, DSQRT, MINO, MOD

## 8. References.

M. J. D. Powell, A Hybrid Method for Nonlinear Equations.

Numerical Methods for Nonlinear Algebraic Equations,  
P. Rabinowitz, editor. Gordon and Breach, 1970.

### 9. Example.

The problem is to determine the values of  $x(1)$ ,  $x(2)$ , ...,  $x(9)$ , which solve the system of tridiagonal equations

$$\begin{aligned} (3-2*x(1))*x(1) & & -2*x(2) & & & & = -1 \\ -x(i-1) + (3-2*x(i))*x(i) & & & & -2*x(i+1) & & = -1, \quad i=2-8 \\ & & -x(8) + (3-2*x(9))*x(9) & & & & = -1 \end{aligned}$$

```

C *****
C
C DRIVER FOR HYBRD EXAMPLE.
C DOUBLE PRECISION VERSION
C
C *****
C INTEGER J,N,MAXFEV,ML,MU,MODE,NPRINT,INFO,NFEV,LDFJAC,LR,NWRITE
C DOUBLE PRECISION XTOL,EPSEFCN,FACTOR,FNORM
C DOUBLE PRECISION X(9),FVEC(9),DIAG(9),FJAC(9,9),R(45),QTF(9),
*      WA1(9),WA2(9),WA3(9),WA4(9)
C DOUBLE PRECISION ENORM,DPMPAR
C EXTERNAL FCN
C
C LOGICAL OUTPUT UNIT IS ASSUMED TO BE NUMBER 6.
C
C DATA NWRITE /6/
C
C N = 9
C
C THE FOLLOWING STARTING VALUES PROVIDE A ROUGH SOLUTION.
C
C DO 10 J = 1, 9
C   X(J) = -1.DO
10 CONTINUE
C
C LDFJAC = 9
C LR = 45
C
C SET XTOL TO THE SQUARE ROOT OF THE MACHINE PRECISION.
C UNLESS HIGH PRECISION SOLUTIONS ARE REQUIRED,
C THIS IS THE RECOMMENDED SETTING.
C
C XTOL = DSQRT(DPMPAR(1))
C
C MAXFEV = 2000
C ML = 1
C MU = 1
C EPSEFCN = 0.DO
C MODE = 2
C DO 20 J = 1, 9
C   DIAG(J) = 1.DO

```

```

20    CONTINUE
      FACTOR = 1.D2
      NPRINT = 0

```

C

```

      CALL HYBRD(FCN,N,X,FVEC,XTOL,MAXFEV,ML,MU,EPSEFCN,DIAG,
*             MODE,FACTOR,NPRINT,INFO,NFEV,FJAC,LDFJAC,
*             R,LR,QTF,WA1,WA2,WA3,WA4)
      FNORM = ENORM(N,FVEC)
      WRITE (NWRITE,1000) FNORM,NFEV,INFO,(X(J),J=1,N)
      STOP

```

```

1000 FORMAT (5X,31H FINAL L2 NORM OF THE RESIDUALS,D15.7 //
*          5X,31H NUMBER OF FUNCTION EVALUATIONS,I10 //
*          5X,15H EXIT PARAMETER,16X,I10 //
*          5X,27H FINAL APPROXIMATE SOLUTION // (5X,3D15.7))

```

C

C

C

```

      LAST CARD OF DRIVER FOR HYBRD EXAMPLE.

```

```

      END
      SUBROUTINE FCN(N,X,FVEC,IFLAG)
      INTEGER N,IFLAG
      DOUBLE PRECISION X(N),FVEC(N)

```

C

C

C

```

      SUBROUTINE FCN FOR HYBRD EXAMPLE.

```

```

      INTEGER K
      DOUBLE PRECISION ONE,TEMP,TEMP1,TEMP2,THREE,TWO,ZERO
      DATA ZERO,ONE,TWO,THREE /0.DO,1.DO,2.DO,3.DO/

```

C

C

C

C

C

```

      IF (IFLAG .NE. 0) GO TO 5

```

```

      INSERT PRINT STATEMENTS HERE WHEN NPRINT IS POSITIVE.

```

```

      RETURN

```

```

5    CONTINUE

```

```

      DO 10 K = 1, N
        TEMP = (THREE - TWO*X(K))*X(K)
        TEMP1 = ZERO
        IF (K .NE. 1) TEMP1 = X(K-1)
        TEMP2 = ZERO
        IF (K .NE. N) TEMP2 = X(K+1)
        FVEC(K) = TEMP - TEMP1 - TWO*TEMP2 + ONE

```

```

10   CONTINUE

```

```

      RETURN

```

C

C

C

```

      LAST CARD OF SUBROUTINE FCN.

```

```

      END

```

Results obtained with different compilers or machines  
may be slightly different.

```

FINAL L2 NORM OF THE RESIDUALS    0.1192636D-07

```

```

NUMBER OF FUNCTION EVALUATIONS          14

```

EXIT PARAMETER

1

FINAL APPROXIMATE SOLUTION

-0.5706545D+00 -0.6816283D+00 -0.7017325D+00  
-0.7042129D+00 -0.7013690D+00 -0.6918656D+00  
-0.6657920D+00 -0.5960342D+00 -0.4164121D+00

## Documentation for MINPACK subroutine HYBRJ1

Double precision version

Argonne National Laboratory

Burton S. Garbow, Kenneth E. Hillstom, Jorge J. More

March 1980

## 1. Purpose.

The purpose of HYBRJ1 is to find a zero of a system of  $N$  nonlinear functions in  $N$  variables by a modification of the Powell hybrid method. This is done by using the more general nonlinear equation solver HYBRJ. The user must provide a subroutine which calculates the functions and the Jacobian.

## 2. Subroutine and type statements.

```
SUBROUTINE HYBRJ1(FCN,N,X,FVEC,FJAC,LDFJAC,TOL,INFO,WA,LWA)
INTEGER N,LDFJAC,INFO,LWA
DOUBLE PRECISION TOL
DOUBLE PRECISION X(N),FVEC(N),FJAC(LDFJAC,N),WA(LWA)
EXTERNAL FCN
```

## 3. Parameters.

Parameters designated as input parameters must be specified on entry to HYBRJ1 and are not changed on exit, while parameters designated as output parameters need not be specified on entry and are set to appropriate values on exit from HYBRJ1.

FCN is the name of the user-supplied subroutine which calculates the functions and the Jacobian. FCN must be declared in an EXTERNAL statement in the user calling program, and should be written as follows.

```
SUBROUTINE FCN(N,X,FVEC,FJAC,LDFJAC,IFLAG)
INTEGER N,LDFJAC,IFLAG
DOUBLE PRECISION X(N),FVEC(N),FJAC(LDFJAC,N)
-----
IF IFLAG = 1 CALCULATE THE FUNCTIONS AT X AND
RETURN THIS VECTOR IN FVEC. DO NOT ALTER FJAC.
IF IFLAG = 2 CALCULATE THE JACOBIAN AT X AND
RETURN THIS MATRIX IN FJAC. DO NOT ALTER FVEC.
-----
RETURN
END
```

The value of IFLAG should not be changed by FCN unless the

user wants to terminate execution of HYBRJ1. In this case set IFLAG to a negative integer.

N is a positive integer input variable set to the number of functions and variables.

X is an array of length N. On input X must contain an initial estimate of the solution vector. On output X contains the final estimate of the solution vector.

FVEC is an output array of length N which contains the functions evaluated at the output X.

FJAC is an output N by N array which contains the orthogonal matrix Q produced by the QR factorization of the final approximate Jacobian. Section 6 contains more details about the approximation to the Jacobian.

LDFJAC is a positive integer input variable not less than N which specifies the leading dimension of the array FJAC.

TOL is a nonnegative input variable. Termination occurs when the algorithm estimates that the relative error between X and the solution is at most TOL. Section 4 contains more details about TOL.

INFO is an integer output variable. If the user has terminated execution, INFO is set to the (negative) value of IFLAG. See description of FCN. Otherwise, INFO is set as follows.

INFO = 0 Improper input parameters.

INFO = 1 Algorithm estimates that the relative error between X and the solution is at most TOL.

INFO = 2 Number of calls to FCN with IFLAG = 1 has reached  $100*(N+1)$ .

INFO = 3 TOL is too small. No further improvement in the approximate solution X is possible.

INFO = 4 Iteration is not making good progress.

Sections 4 and 5 contain more details about INFO.

WA is a work array of length LWA.

LWA is a positive integer input variable not less than  $(N*(N+13))/2$ .

#### 4. Successful completion.

The accuracy of HYBRJ1 is controlled by the convergence

parameter TOL. This parameter is used in a test which makes a comparison between the approximation X and a solution XSOL. HYBRJ1 terminates when the test is satisfied. If TOL is less than the machine precision (as defined by the MINPACK function DPMPAR(1)), then HYBRJ1 only attempts to satisfy the test defined by the machine precision. Further progress is not usually possible. Unless high precision solutions are required, the recommended value for TOL is the square root of the machine precision.

The 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 HYBRJ1 may incorrectly indicate convergence. The coding of the Jacobian can be checked by the MINPACK subroutine CHKDER. If the Jacobian is coded correctly, then the validity of the answer can be checked, for example, by rerunning HYBRJ1 with a tighter tolerance.

Convergence test. If ENORM(Z) denotes the Euclidean norm of a vector Z, then this test attempts to guarantee that

$$\text{ENORM}(X-XSOL) \leq \text{TOL} * \text{ENORM}(XSOL).$$

If this condition is satisfied with  $\text{TOL} = 10^{*(-K)}$ , then the larger components of X have K significant decimal digits and INFO is set to 1. There is a danger that the smaller components of X may have large relative errors, but the fast rate of convergence of HYBRJ1 usually avoids this possibility.

## 5. Unsuccessful completion.

Unsuccessful termination of HYBRJ1 can be due to improper input parameters, arithmetic interrupts, an excessive number of function evaluations, or lack of good progress.

Improper input parameters. INFO is set to 0 if N  $\leq$  0, or LDFJAC  $\leq$  N, or TOL  $\leq$  0.00, or LWA  $\leq$  (N\*(N+13))/2.

Arithmetic interrupts. If these interrupts occur in the FCN subroutine during an early stage of the computation, they may be caused by an unacceptable choice of X by HYBRJ1. In this case, it may be possible to remedy the situation by not evaluating the functions here, but instead setting the components of FVEC to numbers that exceed those in the initial FVEC, thereby indirectly reducing the step length. The step length can be more directly controlled by using instead HYBRJ, which includes in its calling sequence the step-length-governing parameter FACTOR.

Excessive number of function evaluations. If the number of calls to FCN with IFLAG = 1 reaches  $100*(N+1)$ , then this indicates that the routine is converging very slowly as measured

by the progress of FVEC, and INFO is set to 2. This situation should be unusual because, as indicated below, lack of good progress is usually diagnosed earlier by HYBRJ1, causing termination with INFO = 4.

Lack of good progress. HYBRJ1 searches for a zero of the system by minimizing the sum of the squares of the functions. In so doing, it can become trapped in a region where the minimum does not correspond to a zero of the system and, in this situation, the iteration eventually fails to make good progress. In particular, this will happen if the system does not have a zero. If the system has a zero, rerunning HYBRJ1 from a different starting point may be helpful.

## 6. Characteristics of the algorithm.

HYBRJ1 is a modification of the Powell hybrid method. Two of its main characteristics involve the choice of the correction as a convex combination of the Newton and scaled gradient directions, and the updating of the Jacobian by the rank-1 method of Broyden. The choice of the correction guarantees (under reasonable conditions) global convergence for starting points far from the solution and a fast rate of convergence. The Jacobian is calculated at the starting point, but it is not recalculated until the rank-1 method fails to produce satisfactory progress.

Timing. The time required by HYBRJ1 to solve a given problem depends on N, the behavior of the functions, the accuracy requested, and the starting point. The number of arithmetic operations needed by HYBRJ1 is about  $11.5*(N**2)$  to process each evaluation of the functions (call to FCN with IFLAG = 1) and  $1.3*(N**3)$  to process each evaluation of the Jacobian (call to FCN with IFLAG = 2). Unless FCN can be evaluated quickly, the timing of HYBRJ1 will be strongly influenced by the time spent in FCN.

Storage. HYBRJ1 requires  $(3*N**2 + 17*N)/2$  double precision storage locations, in addition to the storage required by the program. There are no internally declared storage arrays.

## 7. Subprograms required.

USER-supplied ..... FCN

MINPACK-supplied ... DOGLEG, DPMPAR, ENORM, HYBRJ,  
QFORM, QRFAC, R1MPYQ, R1UPDT

FORTRAN-supplied ... DABS, DMAX1, DMIN1, DSQRT, MINO, MOD

## 8. References.

M. J. D. Powell, A Hybrid Method for Nonlinear Equations.  
 Numerical Methods for Nonlinear Algebraic Equations,  
 P. Rabinowitz, editor. Gordon and Breach, 1970.

### 9. Example.

The problem is to determine the values of  $x(1)$ ,  $x(2)$ , ...,  $x(9)$ , which solve the system of tridiagonal equations

$$\begin{aligned} (3-2*x(1))*x(1) & & -2*x(2) & & & & = -1 \\ -x(i-1) + (3-2*x(i))*x(i) & & & & -2*x(i+1) & & = -1, \quad i=2-8 \\ & & -x(8) + (3-2*x(9))*x(9) & & & & = -1 \end{aligned}$$

```

C *****
C
C DRIVER FOR HYBRJ1 EXAMPLE.
C DOUBLE PRECISION VERSION
C
C *****
C INTEGER J,N,LDFJAC,INFO,LWA,NWRITE
C DOUBLE PRECISION TOL, FNORM
C DOUBLE PRECISION X(9), FVEC(9), FJAC(9,9), WA(99)
C DOUBLE PRECISION ENORM, DPMPAR
C EXTERNAL FCN
C
C LOGICAL OUTPUT UNIT IS ASSUMED TO BE NUMBER 6.
C
C DATA NWRITE /6/
C
C N = 9
C
C THE FOLLOWING STARTING VALUES PROVIDE A ROUGH SOLUTION.
C
C DO 10 J = 1, 9
C   X(J) = -1.DO
10 CONTINUE
C
C LDFJAC = 9
C LWA = 99
C
C SET TOL TO THE SQUARE ROOT OF THE MACHINE PRECISION.
C UNLESS HIGH PRECISION SOLUTIONS ARE REQUIRED,
C THIS IS THE RECOMMENDED SETTING.
C
C TOL = DSQRT(DPMPAR(1))
C
C CALL HYBRJ1(FCN,N,X,FVEC,FJAC,LDFJAC,TOL,INFO,WA,LWA)
C FNORM = ENORM(N,FVEC)
C WRITE (NWRITE,1000) FNORM,INFO,(X(J),J=1,N)
C STOP
1000 FORMAT (5X,31H FINAL L2 NORM OF THE RESIDUALS,D15.7 //
*          5X,15H EXIT PARAMETER,16X,I10 //
*          5X,27H FINAL APPROXIMATE SOLUTION // (5X,3D15.7))

```

```

C
C
C
LAST CARD OF DRIVER FOR HYBRJ1 EXAMPLE.

END
SUBROUTINE FCN(N,X,FVEC,FJAC,LDFJAC,IFLAG)
INTEGER N,LDFJAC,IFLAG
DOUBLE PRECISION X(N),FVEC(N),FJAC(LDFJAC,N)

C
C
C
SUBROUTINE FCN FOR HYBRJ1 EXAMPLE.

INTEGER J,K
DOUBLE PRECISION ONE,TEMP,TEMP1,TEMP2,THREE,TWO,ZERO
DATA ZERO,ONE,TWO,THREE,FOUR /0.DO,1.DO,2.DO,3.DO,4.DO/

C
IF (IFLAG .EQ. 2) GO TO 20
DO 10 K = 1, N
  TEMP = (THREE - TWO*X(K))*X(K)
  TEMP1 = ZERO
  IF (K .NE. 1) TEMP1 = X(K-1)
  TEMP2 = ZERO
  IF (K .NE. N) TEMP2 = X(K+1)
  FVEC(K) = TEMP - TEMP1 - TWO*TEMP2 + ONE
10  CONTINUE
   GO TO 50
20  CONTINUE
   DO 40 K = 1, N
     DO 30 J = 1, N
       FJAC(K,J) = ZERO
30   CONTINUE
     FJAC(K,K) = THREE - FOUR*X(K)
     IF (K .NE. 1) FJAC(K,K-1) = -ONE
     IF (K .NE. N) FJAC(K,K+1) = -TWO
40   CONTINUE
50  CONTINUE
   RETURN

C
C
C
LAST CARD OF SUBROUTINE FCN.

END

Results obtained with different compilers or machines
may be slightly different.

FINAL L2 NORM OF THE RESIDUALS  0.1192636D-07

EXIT PARAMETER                      1

FINAL APPROXIMATE SOLUTION

-0.5706545D+00 -0.6816283D+00 -0.7017325D+00
-0.7042129D+00 -0.7013690D+00 -0.6918656D+00
-0.6657920D+00 -0.5960342D+00 -0.4164121D+00

```

## Documentation for MINPACK subroutine HYBRJ

Double precision version

Argonne National Laboratory

Burton S. Garbow, Kenneth E. Hillstom, Jorge J. More

March 1980

## 1. Purpose.

The purpose of HYBRJ is to find a zero of a system of  $N$  non-linear functions in  $N$  variables by a modification of the Powell hybrid method. The user must provide a subroutine which calculates the functions and the Jacobian.

## 2. Subroutine and type statements.

```

SUBROUTINE HYBRJ(FCN,N,X,FVEC,FJAC,LDFJAC,XTOL,MAXFEV,DIAG,
*             MODE,FACTOR,NPRINT,INFO,NFEV,NJEV,R,LR,QTF,
*             WA1,WA2,WA3,WA4)
  INTEGER N,LDFJAC,MAXFEV,MODE,NPRINT,INFO,NFEV,NJEV,LR
  DOUBLE PRECISION XTOL,FACTOR
  DOUBLE PRECISION X(N),FVEC(N),FJAC(LDFJAC,N),DIAG(N),R(LR),QTF(N),
*             WA1(N),WA2(N),WA3(N),WA4(N)

```

## 3. Parameters.

Parameters designated as input parameters must be specified on entry to HYBRJ and are not changed on exit, while parameters designated as output parameters need not be specified on entry and are set to appropriate values on exit from HYBRJ.

FCN is the name of the user-supplied subroutine which calculates the functions and the Jacobian. FCN must be declared in an EXTERNAL statement in the user calling program, and should be written as follows.

```

SUBROUTINE FCN(N,X,FVEC,FJAC,LDFJAC,IFLAG)
  INTEGER N,LDFJAC,IFLAG
  DOUBLE PRECISION X(N),FVEC(N),FJAC(LDFJAC,N)
  -----
  IF IFLAG = 1 CALCULATE THE FUNCTIONS AT X AND
  RETURN THIS VECTOR IN FVEC. DO NOT ALTER FJAC.
  IF IFLAG = 2 CALCULATE THE JACOBIAN AT X AND
  RETURN THIS MATRIX IN FJAC. DO NOT ALTER FVEC.
  -----
  RETURN
  END

```

The value of IFLAG should not be changed by FCN unless the user wants to terminate execution of HYBRJ. In this case set IFLAG to a negative integer.

N is a positive integer input variable set to the number of functions and variables.

X is an array of length N. On input X must contain an initial estimate of the solution vector. On output X contains the final estimate of the solution vector.

FVEC is an output array of length N which contains the functions evaluated at the output X.

FJAC is an output N by N array which contains the orthogonal matrix Q produced by the QR factorization of the final approximate Jacobian. Section 6 contains more details about the approximation to the Jacobian.

LDFJAC is a positive integer input variable not less than N which specifies the leading dimension of the array FJAC.

XTOL is a nonnegative input variable. Termination occurs when the relative error between two consecutive iterates is at most XTOL. Therefore, XTOL measures the relative error desired in the approximate solution. Section 4 contains more details about XTOL.

MAXFEV is a positive integer input variable. Termination occurs when the number of calls to FCN with IFLAG = 1 has reached MAXFEV.

DIAG is an array of length N. If MODE = 1 (see below), DIAG is internally set. If MODE = 2, DIAG must contain positive entries that serve as multiplicative scale factors for the variables.

MODE is an integer input variable. If MODE = 1, the variables will be scaled internally. If MODE = 2, the scaling is specified by the input DIAG. Other values of MODE are equivalent to MODE = 1.

FACTOR is a positive input variable used in determining the initial step bound. This bound is set to the product of FACTOR and the Euclidean norm of DIAG\*X if nonzero, or else to FACTOR itself. In most cases FACTOR should lie in the interval (.1,100.). 100. is a generally recommended value.

NPRINT is an integer input variable that enables controlled printing of iterates if it is positive. In this case, FCN is called with IFLAG = 0 at the beginning of the first iteration and every NPRINT iterations thereafter and immediately prior to return, with X and FVEC available for printing. FVEC and FJAC should not be altered. If NPRINT is not positive, no

special calls of FCN with IFLAG = 0 are made.

INFO is an integer output variable. If the user has terminated execution, INFO is set to the (negative) value of IFLAG. See description of FCN. Otherwise, INFO is set as follows.

INFO = 0 Improper input parameters.

INFO = 1 Relative error between two consecutive iterates is at most XTOL.

INFO = 2 Number of calls to FCN with IFLAG = 1 has reached MAXFEV.

INFO = 3 XTOL is too small. No further improvement in the approximate solution X is possible.

INFO = 4 Iteration is not making good progress, as measured by the improvement from the last five Jacobian evaluations.

INFO = 5 Iteration is not making good progress, as measured by the improvement from the last ten iterations.

Sections 4 and 5 contain more details about INFO.

NFEV is an integer output variable set to the number of calls to FCN with IFLAG = 1.

NJEV is an integer output variable set to the number of calls to FCN with IFLAG = 2.

R is an output array of length LR which contains the upper triangular matrix produced by the QR factorization of the final approximate Jacobian, stored rowwise.

LR is a positive integer input variable not less than  $(N*(N+1))/2$ .

QTF is an output array of length N which contains the vector  $(Q \text{ transpose}) * FVEC$ .

WA1, WA2, WA3, and WA4 are work arrays of length N.

#### 4. Successful completion.

The accuracy of HYBRJ is controlled by the convergence parameter XTOL. This parameter is used in a test which makes a comparison between the approximation X and a solution XSOL. HYBRJ terminates when the test is satisfied. If the convergence parameter is less than the machine precision (as defined by the MINPACK function DPMPAR(1)), then HYBRJ only attempts to satisfy the test defined by the machine precision. Further progress is not

usually possible.

The 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 HYBRJ may incorrectly indicate convergence. The coding of the Jacobian can be checked by the MINPACK subroutine CHKDER. If the Jacobian is coded correctly, then the validity of the answer can be checked, for example, by rerunning HYBRJ with a tighter tolerance.

Convergence test. If  $ENORM(Z)$  denotes the Euclidean norm of a vector  $Z$  and  $D$  is the diagonal matrix whose entries are defined by the array  $DIAG$ , then this test attempts to guarantee that

$$ENORM(D*(X-XSOL)) \leq XTOL*ENORM(D*XSOL).$$

If this condition is satisfied with  $XTOL = 10^{*(-K)}$ , then the larger components of  $D*X$  have  $K$  significant decimal digits and  $INFO$  is set to 1. There is a danger that the smaller components of  $D*X$  may have large relative errors, but the fast rate of convergence of HYBRJ usually avoids this possibility. Unless high precision solutions are required, the recommended value for  $XTOL$  is the square root of the machine precision.

## 5. Unsuccessful completion.

Unsuccessful termination of HYBRJ can be due to improper input parameters, arithmetic interrupts, an excessive number of function evaluations, or lack of good progress.

Improper input parameters.  $INFO$  is set to 0 if  $N \leq 0$ , or  $LDFJAC < N$ , or  $XTOL < 0$ , or  $MAXFEV \leq 0$ , or  $FACTOR \leq 0$ , or  $LR < (N*(N+1))/2$ .

Arithmetic interrupts. If these interrupts occur in the FCN subroutine during an early stage of the computation, they may be caused by an unacceptable choice of  $X$  by HYBRJ. In this case, it may be possible to remedy the situation by rerunning HYBRJ with a smaller value of  $FACTOR$ .

Excessive number of function evaluations. A reasonable value for  $MAXFEV$  is  $100*(N+1)$ . If the number of calls to FCN with  $IFLAG = 1$  reaches  $MAXFEV$ , then this indicates that the routine is converging very slowly as measured by the progress of  $FVEC$ , and  $INFO$  is set to 2. This situation should be unusual because, as indicated below, lack of good progress is usually diagnosed earlier by HYBRJ, causing termination with  $INFO = 4$  or  $INFO = 5$ .

Lack of good progress. HYBRJ searches for a zero of the system by minimizing the sum of the squares of the functions. In so

doing, it can become trapped in a region where the minimum does not correspond to a zero of the system and, in this situation, the iteration eventually fails to make good progress. In particular, this will happen if the system does not have a zero. If the system has a zero, rerunning HYBRJ from a different starting point may be helpful.

## 6. Characteristics of the algorithm.

HYBRJ is a modification of the Powell hybrid method. Two of its main characteristics involve the choice of the correction as a convex combination of the Newton and scaled gradient directions, and the updating of the Jacobian by the rank-1 method of Broyden. The choice of the correction guarantees (under reasonable conditions) global convergence for starting points far from the solution and a fast rate of convergence. The Jacobian is calculated at the starting point, but it is not recalculated until the rank-1 method fails to produce satisfactory progress.

**Timing.** The time required by HYBRJ to solve a given problem depends on  $N$ , the behavior of the functions, the accuracy requested, and the starting point. The number of arithmetic operations needed by HYBRJ is about  $11.5*(N**2)$  to process each evaluation of the functions (call to FCN with IFLAG = 1) and  $1.3*(N**3)$  to process each evaluation of the Jacobian (call to FCN with IFLAG = 2). Unless FCN can be evaluated quickly, the timing of HYBRJ will be strongly influenced by the time spent in FCN.

**Storage.** HYBRJ requires  $(3*N**2 + 17*N)/2$  double precision storage locations, in addition to the storage required by the program. There are no internally declared storage arrays.

## 7. Subprograms required.

USER-supplied ..... FCN

MINPACK-supplied ... DOGLEG, DPMPAR, ENORM,  
QFORM, QRFAC, R1MPYQ, R1UPDT

FORTTRAN-supplied ... DABS, DMAX1, DMIN1, DSQRT, MINO, MOD

## 8. References.

M. J. D. Powell, A Hybrid Method for Nonlinear Equations. Numerical Methods for Nonlinear Algebraic Equations, P. Rabinowitz, editor. Gordon and Breach, 1970.

## 9. Example.

The problem is to determine the values of  $x(1)$ ,  $x(2)$ , ...,  $x(9)$ , which solve the system of tridiagonal equations

$$\begin{aligned} (3-2*x(1))*x(1) & & -2*x(2) & & & = -1 \\ -x(i-1) + (3-2*x(i))*x(i) & & & & -2*x(i+1) & = -1, \quad i=2-8 \\ & & -x(8) + (3-2*x(9))*x(9) & & & = -1 \end{aligned}$$

```

C *****
C
C DRIVER FOR HYBRJ EXAMPLE.
C DOUBLE PRECISION VERSION
C
C *****
C INTEGER J,N,LDFJAC,MAXFEV,MODE,NPRINT,INFO,NFEV,NJEV,LR,NWRITE
C DOUBLE PRECISION XTOL,FACTOR,FNORM
C DOUBLE PRECISION X(9),FVEC(9),FJAC(9,9),DIAG(9),R(45),QTF(9),
* WA1(9),WA2(9),WA3(9),WA4(9)
C DOUBLE PRECISION ENORM,DPMPAR
C EXTERNAL FCN
C
C LOGICAL OUTPUT UNIT IS ASSUMED TO BE NUMBER 6.
C
C DATA NWRITE /6/
C
C N = 9
C
C THE FOLLOWING STARTING VALUES PROVIDE A ROUGH SOLUTION.
C
C DO 10 J = 1, 9
C   X(J) = -1.DO
10 CONTINUE
C
C LDFJAC = 9
C LR = 45
C
C SET XTOL TO THE SQUARE ROOT OF THE MACHINE PRECISION.
C UNLESS HIGH PRECISION SOLUTIONS ARE REQUIRED,
C THIS IS THE RECOMMENDED SETTING.
C
C XTOL = DSQRT(DPMPAR(1))
C
C MAXFEV = 1000
C MODE = 2
C DO 20 J = 1, 9
C   DIAG(J) = 1.DO
20 CONTINUE
C FACTOR = 1.D2
C NPRINT = 0
C
C CALL HYBRJ(FCN,N,X,FVEC,FJAC,LDFJAC,XTOL,MAXFEV,DIAG,
* MODE,FACTOR,NPRINT,INFO,NFEV,NJEV,R,LR,QTF,
* WA1,WA2,WA3,WA4)
C FNORM = ENORM(N,FVEC)
C WRITE (NWRITE,1000) FNORM,NFEV,NJEV,INFO,(X(J),J=1,N)

```

```

      STOP
1000 FORMAT (5X,31H FINAL L2 NORM OF THE RESIDUALS,D15.7 //
*          5X,31H NUMBER OF FUNCTION EVALUATIONS,I10 //
*          5X,31H NUMBER OF JACOBIAN EVALUATIONS,I10 //
*          5X,15H EXIT PARAMETER,16X,I10 //
*          5X,27H FINAL APPROXIMATE SOLUTION // (5X,3D15.7))
C
C      LAST CARD OF DRIVER FOR HYBRJ EXAMPLE.
C
      END
      SUBROUTINE FCN(N,X,FVEC,FJAC,LDFJAC,IFLAG)
      INTEGER N,LDFJAC,IFLAG
      DOUBLE PRECISION X(N),FVEC(N),FJAC(LDFJAC,N)
C
C      SUBROUTINE FCN FOR HYBRJ EXAMPLE.
C
      INTEGER J,K
      DOUBLE PRECISION ONE,TEMP,TEMP1,TEMP2,THREE,TWO,ZERO
      DATA ZERO,ONE,TWO,THREE,FOUR /0.DO,1.DO,2.DO,3.DO,4.DO/
C
      IF (IFLAG .NE. 0) GO TO 5
C
C      INSERT PRINT STATEMENTS HERE WHEN NPRINT IS POSITIVE.
C
      RETURN
5 CONTINUE
      IF (IFLAG .EQ. 2) GO TO 20
      DO 10 K = 1, N
          TEMP = (THREE - TWO*X(K))*X(K)
          TEMP1 = ZERO
          IF (K .NE. 1) TEMP1 = X(K-1)
          TEMP2 = ZERO
          IF (K .NE. N) TEMP2 = X(K+1)
          FVEC(K) = TEMP - TEMP1 - TWO*TEMP2 + ONE
10      CONTINUE
          GO TO 50
20 CONTINUE
      DO 40 K = 1, N
          DO 30 J = 1, N
              FJAC(K,J) = ZERO
30          CONTINUE
              FJAC(K,K) = THREE - FOUR*X(K)
              IF (K .NE. 1) FJAC(K,K-1) = -ONE
              IF (K .NE. N) FJAC(K,K+1) = -TWO
40          CONTINUE
50 CONTINUE
      RETURN
C
C      LAST CARD OF SUBROUTINE FCN.
C
      END

```

Results obtained with different compilers or machines  
may be slightly different.

FINAL L2 NORM OF THE RESIDUALS 0.1192636D-07

NUMBER OF FUNCTION EVALUATIONS 11

NUMBER OF JACOBIAN EVALUATIONS 1

EXIT PARAMETER 1

FINAL APPROXIMATE SOLUTION

-0.5706545D+00 -0.6816283D+00 -0.7017325D+00

-0.7042129D+00 -0.7013690D+00 -0.6918656D+00

-0.6657920D+00 -0.5960342D+00 -0.4164121D+00

## Documentation for MINPACK subroutine LMDER1

Double precision version

Argonne National Laboratory

Burton S. Garbow, Kenneth E. Hillstom, Jorge J. More

March 1980

## 1. Purpose.

The purpose of LMDER1 is to minimize the sum of the squares of M nonlinear functions in N variables by a modification of the Levenberg-Marquardt algorithm. This is done by using the more general least-squares solver LMDER. The user must provide a subroutine which calculates the functions and the Jacobian.

## 2. Subroutine and type statements.

```

SUBROUTINE LMDER1(FCN,M,N,X,FVEC,FJAC,LDFJAC,TOL,
*              INFO,IPVT,WA,LWA)
INTEGER M,N,LDFJAC,INFO,LWA
INTEGER IPVT(N)
DOUBLE PRECISION TOL
DOUBLE PRECISION X(N),FVEC(M),FJAC(LDFJAC,N),WA(LWA)
EXTERNAL FCN

```

## 3. Parameters.

Parameters designated as input parameters must be specified on entry to LMDER1 and are not changed on exit, while parameters designated as output parameters need not be specified on entry and are set to appropriate values on exit from LMDER1.

FCN is the name of the user-supplied subroutine which calculates the functions and the Jacobian. FCN must be declared in an EXTERNAL statement in the user calling program, and should be written as follows.

```

SUBROUTINE FCN(M,N,X,FVEC,FJAC,LDFJAC,IFLAG)
INTEGER M,N,LDFJAC,IFLAG
DOUBLE PRECISION X(N),FVEC(M),FJAC(LDFJAC,N)
-----
IF IFLAG = 1 CALCULATE THE FUNCTIONS AT X AND
RETURN THIS VECTOR IN FVEC. DO NOT ALTER FJAC.
IF IFLAG = 2 CALCULATE THE JACOBIAN AT X AND
RETURN THIS MATRIX IN FJAC. DO NOT ALTER FVEC.
-----
RETURN
END

```

The value of IFLAG should not be changed by FCN unless the user wants to terminate execution of LMDER1. In this case set IFLAG to a negative integer.

M is a positive integer input variable set to the number of functions.

N is a positive integer input variable set to the number of variables. N must not exceed M.

X is an array of length N. On input X must contain an initial estimate of the solution vector. On output X contains the final estimate of the solution vector.

FVEC is an output array of length M which contains the functions evaluated at the output X.

FJAC is an output M by N array. The upper N by N submatrix of FJAC contains an upper triangular matrix R with diagonal elements of nonincreasing magnitude such that

$$P^T (JAC^T JAC) P = R^T R,$$

where P is a permutation matrix and JAC is the final calculated Jacobian. Column j of P is column IPVT(j) (see below) of the identity matrix. The lower trapezoidal part of FJAC contains information generated during the computation of R.

LDEFJAC is a positive integer input variable not less than M which specifies the leading dimension of the array FJAC.

TOL is a nonnegative input variable. Termination occurs when the algorithm estimates either that the relative error in the sum of squares is at most TOL or that the relative error between X and the solution is at most TOL. Section 4 contains more details about TOL.

INFO is an integer output variable. If the user has terminated execution, INFO is set to the (negative) value of IFLAG. See description of FCN. Otherwise, INFO is set as follows.

INFO = 0 Improper input parameters.

INFO = 1 Algorithm estimates that the relative error in the sum of squares is at most TOL.

INFO = 2 Algorithm estimates that the relative error between X and the solution is at most TOL.

INFO = 3 Conditions for INFO = 1 and INFO = 2 both hold.

INFO = 4 FVEC is orthogonal to the columns of the Jacobian to machine precision.

INFO = 5 Number of calls to FCN with IFLAG = 1 has reached 100\*(N+1).

INFO = 6 TOL is too small. No further reduction in the sum of squares is possible.

INFO = 7 TOL is too small. No further improvement in the approximate solution X is possible.

Sections 4 and 5 contain more details about INFO.

IPVT is an integer output array of length N. IPVT defines a permutation matrix P such that  $JAC * P = Q * R$ , where JAC is the final calculated Jacobian, Q is orthogonal (not stored), and R is upper triangular with diagonal elements of nonincreasing magnitude. Column j of P is column IPVT(j) of the identity matrix.

WA is a work array of length LWA.

LWA is a positive integer input variable not less than  $5 * N + M$ .

#### 4. Successful completion.

The accuracy of LMDER1 is controlled by the convergence parameter TOL. This parameter is used in tests which make three types of comparisons between the approximation X and a solution XSOL. LMDER1 terminates when any of the tests is satisfied. If TOL is less than the machine precision (as defined by the MINPACK function DPMPAR(1)), then LMDER1 only attempts to satisfy the test defined by the machine precision. Further progress is not usually possible. Unless high precision solutions are required, the recommended value for TOL is the square root of the machine precision.

The tests assume that the functions and the Jacobian are coded consistently, and that the functions are reasonably well behaved. If these conditions are not satisfied, then LMDER1 may incorrectly indicate convergence. The coding of the Jacobian can be checked by the MINPACK subroutine CHKDER. If the Jacobian is coded correctly, then the validity of the answer can be checked, for example, by rerunning LMDER1 with a tighter tolerance.

First convergence test. If ENORM(Z) denotes the Euclidean norm of a vector Z, then this test attempts to guarantee that

$$ENORM(FVEC) \leq (1+TOL) * ENORM(FVECS),$$

where FVECS denotes the functions evaluated at XSOL. If this condition is satisfied with  $TOL = 10^{*-K}$ , then the final residual norm ENORM(FVEC) has K significant decimal digits and INFO is set to 1 (or to 3 if the second test is also

satisfied).

Second convergence test. If  $D$  is a diagonal matrix (implicitly generated by LMDER1) whose entries contain scale factors for the variables, then this test attempts to guarantee that

$$\text{ENORM}(D*(X-XSOL)) \leq \text{TOL} * \text{ENORM}(D*XSOL).$$

If this condition is satisfied with  $\text{TOL} = 10^{**(-K)}$ , then the larger components of  $D*X$  have  $K$  significant decimal digits and INFO is set to 2 (or to 3 if the first test is also satisfied). There is a danger that the smaller components of  $D*X$  may have large relative errors, but the choice of  $D$  is such that the accuracy of the components of  $X$  is usually related to their sensitivity.

Third convergence test. This test is satisfied when FVEC is orthogonal to the columns of the Jacobian to machine precision. There is no clear relationship between this test and the accuracy of LMDER1, and furthermore, the test is equally well satisfied at other critical points, namely maximizers and saddle points. Therefore, termination caused by this test (INFO = 4) should be examined carefully.

## 5. Unsuccessful completion.

Unsuccessful termination of LMDER1 can be due to improper input parameters, arithmetic interrupts, or an excessive number of function evaluations.

Improper input parameters. INFO is set to 0 if  $N \leq 0$ , or  $M < N$ , or  $\text{LDFJAC} < M$ , or  $\text{TOL} < 0$ , or  $\text{LWA} < 5*N+M$ .

Arithmetic interrupts. If these interrupts occur in the FCN subroutine during an early stage of the computation, they may be caused by an unacceptable choice of  $X$  by LMDER1. In this case, it may be possible to remedy the situation by not evaluating the functions here, but instead setting the components of FVEC to numbers that exceed those in the initial FVEC, thereby indirectly reducing the step length. The step length can be more directly controlled by using instead LMDER, which includes in its calling sequence the step-length-governing parameter FACTOR.

Excessive number of function evaluations. If the number of calls to FCN with IFLAG = 1 reaches  $100*(N+1)$ , then this indicates that the routine is converging very slowly as measured by the progress of FVEC, and INFO is set to 5. In this case, it may be helpful to restart LMDER1, thereby forcing it to disregard old (and possibly harmful) information.

## 6. Characteristics of the algorithm.

LMDER1 is a modification of the Levenberg-Marquardt algorithm. Two of its main characteristics involve the proper use of implicitly scaled variables and an optimal choice for the correction. The use of implicitly scaled variables achieves scale invariance of LMDER1 and limits the size of the correction in any direction where the functions are changing rapidly. The optimal choice of the correction guarantees (under reasonable conditions) global convergence from starting points far from the solution and a fast rate of convergence for problems with small residuals.

**Timing.** The time required by LMDER1 to solve a given problem depends on M and N, the behavior of the functions, the accuracy requested, and the starting point. The number of arithmetic operations needed by LMDER1 is about  $N^3$  to process each evaluation of the functions (call to FCN with IFLAG = 1) and  $M(N^2)$  to process each evaluation of the Jacobian (call to FCN with IFLAG = 2). Unless FCN can be evaluated quickly, the timing of LMDER1 will be strongly influenced by the time spent in FCN.

**Storage.** LMDER1 requires  $M \cdot N + 2 \cdot M + 6 \cdot N$  double precision storage locations and N integer storage locations, in addition to the storage required by the program. There are no internally declared storage arrays.

## 7. Subprograms required.

USER-supplied ..... FCN

MINPACK-supplied ... DPMPAR, ENORM, LMDER, LMPAR, ORFAC, QRSOLV

FORTTRAN-supplied ... DABS, DMAX1, DMIN1, DSQRT, MOD

## 8. References.

Jorge J. More, The Levenberg-Marquardt Algorithm, Implementation and Theory. Numerical Analysis, G. A. Watson, editor. Lecture Notes in Mathematics 630, Springer-Verlag, 1977.

## 9. Example.

The problem is to determine the values of  $x(1)$ ,  $x(2)$ , and  $x(3)$  which provide the best fit (in the least squares sense) of

$$x(i) + u(i)/(v(i)*x(2) + w(i)*x(3)), \quad i = 1, 15$$

to the data

$$y = (0.14, 0.18, 0.22, 0.25, 0.29, 0.32, 0.35, 0.39, \\ 0.37, 0.58, 0.73, 0.96, 1.34, 2.10, 4.39),$$

where  $u(i) = i$ ,  $v(i) = 16 - i$ , and  $w(i) = \min(u(i), v(i))$ . The  $i$ -th component of FVEC is thus defined by

$$y(i) - (x(1) + u(i)/(v(i)*x(2) + w(i)*x(3))).$$

```

C *****
C
C DRIVER FOR LMDER1 EXAMPLE.
C DOUBLE PRECISION VERSION
C
C *****
C INTEGER J,M,N,LDFJAC,INFO,LWA,NWRITE
C INTEGER IPVTV(3)
C DOUBLE PRECISION TOL, FNORM
C DOUBLE PRECISION X(3), FVEC(15), FJAC(15,3), WA(30)
C DOUBLE PRECISION ENORM, DPMPAR
C EXTERNAL FCN
C
C LOGICAL OUTPUT UNIT IS ASSUMED TO BE NUMBER 6.
C
C DATA NWRITE /6/
C
C M = 15
C N = 3
C
C THE FOLLOWING STARTING VALUES PROVIDE A ROUGH FIT.
C
C X(1) = 1.DO
C X(2) = 1.DO
C X(3) = 1.DO
C
C LDFJAC = 15
C LWA = 30
C
C SET TOL TO THE SQUARE ROOT OF THE MACHINE PRECISION.
C UNLESS HIGH PRECISION SOLUTIONS ARE REQUIRED,
C THIS IS THE RECOMMENDED SETTING.
C
C TOL = DSQRT(DPMPAR(1))
C
C CALL LMDER1(FCN,M,N,X,FVEC,FJAC,LDFJAC,TOL,
* INFO,IPVTV,WA,LWA)
C FNORM = ENORM(M,FVEC)
C WRITE (NWRITE,1000) FNORM,INFO,(X(J),J=1,N)
C STOP
1000 FORMAT (5X,31H FINAL L2 NORM OF THE RESIDUALS,D15.7 //
* 5X,15H EXIT PARAMETER,16X,I10 //
* 5X,27H FINAL APPROXIMATE SOLUTION // 5X,3D15.7)
C
C LAST CARD OF DRIVER FOR LMDER1 EXAMPLE.
C

```

```

END
SUBROUTINE FCN(M,N,X,FVEC,FJAC,LDFJAC,IFLAG)
INTEGER M,N,LDFJAC,IFLAG
DOUBLE PRECISION X(N),FVEC(M),FJAC(LDFJAC,N)

```

```

SUBROUTINE FCN FOR LMDER1 EXAMPLE.

```

```

INTEGER I
DOUBLE PRECISION TMP1,TMP2,TMP3,TMP4
DOUBLE PRECISION Y(15)
DATA Y(1),Y(2),Y(3),Y(4),Y(5),Y(6),Y(7),Y(8),
*     Y(9),Y(10),Y(11),Y(12),Y(13),Y(14),Y(15)
*     /1.4D-1,1.8D-1,2.2D-1,2.5D-1,2.9D-1,3.2D-1,3.5D-1,3.9D-1,
*     3.7D-1,5.8D-1,7.3D-1,9.6D-1,1.34D0,2.1D0,4.39D0/

```

```

IF (IFLAG .EQ. 2) GO TO 20
DO 10 I = 1, 15
  TMP1 = I
  TMP2 = 16 - I
  TMP3 = TMP1
  IF (I .GT. 8) TMP3 = TMP2
  FVEC(I) = Y(I) - (X(1) + TMP1/(X(2)*TMP2 + X(3)*TMP3))
10  CONTINUE
   GO TO 40
20  CONTINUE
   DO 30 I = 1, 15
     TMP1 = I
     TMP2 = 16 - I
     TMP3 = TMP1
     IF (I .GT. 8) TMP3 = TMP2
     TMP4 = (X(2)*TMP2 + X(3)*TMP3)**2
     FJAC(I,1) = -1.D0
     FJAC(I,2) = TMP1*TMP2/TMP4
     FJAC(I,3) = TMP1*TMP3/TMP4
30  CONTINUE
40  CONTINUE
   RETURN

```

```

LAST CARD OF SUBROUTINE FCN.

```

```

END

```

Results obtained with different compilers or machines  
may be slightly different.

FINAL L2 NORM OF THE RESIDUALS 0.9063596D-01

EXIT PARAMETER 1

FINAL APPROXIMATE SOLUTION

0.8241058D-01 0.1133037D+01 0.2343695D+01



## Documentation for MINPACK subroutine LMDER

Double precision version

Argonne National Laboratory

Burton S. Garbow, Kenneth E. Hillstrom, Jorge J. More

March 1980

## 1. Purpose.

The purpose of LMDER is to minimize the sum of the squares of M nonlinear functions in N variables by a modification of the Levenberg-Marquardt algorithm. The user must provide a subroutine which calculates the functions and the Jacobian.

## 2. Subroutine and type statements.

```

SUBROUTINE LMDER(FCN,M,N,X,FVEC,FJAC,LDFJAC,FTOL,XTOL,GTOL,
*           MAXFEV,DIAG,MODE,FACTOR,NPRINT,INFO,NFEV,NJEV,
*           IPVT,QTF,WA1,WA2,WA3,WA4)
INTEGER M,N,LDFJAC,MAXFEV,MODE,NPRINT,INFO,NFEV,NJEV
INTEGER IPVT(N)
DOUBLE PRECISION FTOL,XTOL,GTOL,FACTOR
DOUBLE PRECISION X(N),FVEC(M),FJAC(LDFJAC,N),DIAG(N),QTF(N),
*           WA1(N),WA2(N),WA3(N),WA4(M)

```

## 3. Parameters.

Parameters designated as input parameters must be specified on entry to LMDER and are not changed on exit, while parameters designated as output parameters need not be specified on entry and are set to appropriate values on exit from LMDER.

FCN is the name of the user-supplied subroutine which calculates the functions and the Jacobian. FCN must be declared in an EXTERNAL statement in the user calling program, and should be written as follows.

```

SUBROUTINE FCN(M,N,X,FVEC,FJAC,LDFJAC,IFLAG)
INTEGER M,N,LDFJAC,IFLAG
DOUBLE PRECISION X(N),FVEC(M),FJAC(LDFJAC,N)
-----
IF IFLAG = 1 CALCULATE THE FUNCTIONS AT X AND
RETURN THIS VECTOR IN FVEC. DO NOT ALTER FJAC.
IF IFLAG = 2 CALCULATE THE JACOBIAN AT X AND
RETURN THIS MATRIX IN FJAC. DO NOT ALTER FVEC.
-----
RETURN
END

```

The value of IFLAG should not be changed by FCN unless the user wants to terminate execution of LMDER. In this case set IFLAG to a negative integer.

M is a positive integer input variable set to the number of functions.

N is a positive integer input variable set to the number of variables. N must not exceed M.

X is an array of length N. On input X must contain an initial estimate of the solution vector. On output X contains the final estimate of the solution vector.

FVEC is an output array of length M which contains the functions evaluated at the output X.

FJAC is an output M by N array. The upper N by N submatrix of FJAC contains an upper triangular matrix R with diagonal elements of nonincreasing magnitude such that

$$P^T (JAC^T * JAC) * P = R^T * R,$$

where P is a permutation matrix and JAC is the final calculated Jacobian. Column j of P is column IPVT(j) (see below) of the identity matrix. The lower trapezoidal part of FJAC contains information generated during the computation of R.

LDFJAC is a positive integer input variable not less than M which specifies the leading dimension of the array FJAC.

FTOL is a nonnegative input variable. Termination occurs when both the actual and predicted relative reductions in the sum of squares are at most FTOL. Therefore, FTOL measures the relative error desired in the sum of squares. Section 4 contains more details about FTOL.

XTOL is a nonnegative input variable. Termination occurs when the relative error between two consecutive iterates is at most XTOL. Therefore, XTOL measures the relative error desired in the approximate solution. Section 4 contains more details about XTOL.

GTOL is a nonnegative input variable. Termination occurs when the cosine of the angle between FVEC and any column of the Jacobian is at most GTOL in absolute value. Therefore, GTOL measures the orthogonality desired between the function vector and the columns of the Jacobian. Section 4 contains more details about GTOL.

MAXFEV is a positive integer input variable. Termination occurs when the number of calls to FCN with IFLAG = 1 has reached MAXFEV.

DIAG is an array of length N. If MODE = 1 (see below), DIAG is internally set. If MODE = 2, DIAG must contain positive entries that serve as multiplicative scale factors for the variables.

MODE is an integer input variable. If MODE = 1, the variables will be scaled internally. If MODE = 2, the scaling is specified by the input DIAG. Other values of MODE are equivalent to MODE = 1.

FACTOR is a positive input variable used in determining the initial step bound. This bound is set to the product of FACTOR and the Euclidean norm of DIAG\*X if nonzero, or else to FACTOR itself. In most cases FACTOR should lie in the interval (.1,100.). 100. is a generally recommended value.

NPRINT is an integer input variable that enables controlled printing of iterates if it is positive. In this case, FCN is called with IFLAG = 0 at the beginning of the first iteration and every NPRINT iterations thereafter and immediately prior to return, with X, FVEC, and FJAC available for printing. FVEC and FJAC should not be altered. If NPRINT is not positive, no special calls of FCN with IFLAG = 0 are made.

INFO is an integer output variable. If the user has terminated execution, INFO is set to the (negative) value of IFLAG. See description of FCN. Otherwise, INFO is set as follows.

- INFO = 0 Improper input parameters.
- INFO = 1 Both actual and predicted relative reductions in the sum of squares are at most FTOL.
- INFO = 2 Relative error between two consecutive iterates is at most XTOL.
- INFO = 3 Conditions for INFO = 1 and INFO = 2 both hold.
- INFO = 4 The cosine of the angle between FVEC and any column of the Jacobian is at most GTOL in absolute value.
- INFO = 5 Number of calls to FCN with IFLAG = 1 has reached MAXFEV.
- INFO = 6 FTOL is too small. No further reduction in the sum of squares is possible.
- INFO = 7 XTOL is too small. No further improvement in the approximate solution X is possible.
- INFO = 8 GTOL is too small. FVEC is orthogonal to the columns of the Jacobian to machine precision.

Sections 4 and 5 contain more details about INFO.

NFEV is an integer output variable set to the number of calls to FCN with IFLAG = 1.

NJEV is an integer output variable set to the number of calls to FCN with IFLAG = 2.

IPVT is an integer output array of length N. IPVT defines a permutation matrix P such that  $JAC * P = Q * R$ , where JAC is the final calculated Jacobian, Q is orthogonal (not stored), and R is upper triangular with diagonal elements of nonincreasing magnitude. Column j of P is column IPVT(j) of the identity matrix.

QTF is an output array of length N which contains the first N elements of the vector (Q transpose)\*FVEC.

WA1, WA2, and WA3 are work arrays of length N.

WA4 is a work array of length M.

#### 4. Successful completion.

The accuracy of LMDER is controlled by the convergence parameters FTOL, XTOL, and GTOL. These parameters are used in tests which make three types of comparisons between the approximation X and a solution XSOL. LMDER terminates when any of the tests is satisfied. If any of the convergence parameters is less than the machine precision (as defined by the MINPACK function DPMPAR(1)), then LMDER only attempts to satisfy the test defined by the machine precision. Further progress is not usually possible.

The tests assume that the functions and the Jacobian are coded consistently, and that the functions are reasonably well behaved. If these conditions are not satisfied, then LMDER may incorrectly indicate convergence. The coding of the Jacobian can be checked by the MINPACK subroutine CHKDER. If the Jacobian is coded correctly, then the validity of the answer can be checked, for example, by rerunning LMDER with tighter tolerances.

First convergence test. If ENORM(Z) denotes the Euclidean norm of a vector Z, then this test attempts to guarantee that

$$ENORM(FVEC) \leq (1+FTOL) * ENORM(FVECS),$$

where FVECS denotes the functions evaluated at XSOL. If this condition is satisfied with  $FTOL = 10^{*(-K)}$ , then the final residual norm ENORM(FVEC) has K significant decimal digits and INFO is set to 1 (or to 3 if the second test is also satisfied). Unless high precision solutions are required, the recommended value for FTOL is the square root of the machine precision.

Second convergence test. If D is the diagonal matrix whose entries are defined by the array DIAG, then this test attempts to guarantee that

$$\text{ENORM}(D*(X-XSOL)) \leq \text{XTOL} * \text{ENORM}(D*XSOL).$$

If this condition is satisfied with  $\text{XTOL} = 10^{*(-K)}$ , then the larger components of  $D*X$  have K significant decimal digits and INFO is set to 2 (or to 3 if the first test is also satisfied). There is a danger that the smaller components of  $D*X$  may have large relative errors, but if  $\text{MODE} = 1$ , then the accuracy of the components of X is usually related to their sensitivity. Unless high precision solutions are required, the recommended value for XTOL is the square root of the machine precision.

Third convergence test. This test is satisfied when the cosine of the angle between FVEC and any column of the Jacobian at X is at most GTOL in absolute value. There is no clear relationship between this test and the accuracy of LMDER, and furthermore, the test is equally well satisfied at other critical points, namely maximizers and saddle points. Therefore, termination caused by this test ( $\text{INFO} = 4$ ) should be examined carefully. The recommended value for GTOL is zero.

## 5. Unsuccessful completion.

Unsuccessful termination of LMDER can be due to improper input parameters, arithmetic interrupts, or an excessive number of function evaluations.

Improper input parameters. INFO is set to 0 if  $N \leq 0$ , or  $M < N$ , or  $\text{LDFJAC} < M$ , or  $\text{FTOL} < 0$ , or  $\text{XTOL} < 0$ , or  $\text{GTOL} < 0$ , or  $\text{MAXFEV} \leq 0$ , or  $\text{FACTOR} \leq 0$ .

Arithmetic interrupts. If these interrupts occur in the FCN subroutine during an early stage of the computation, they may be caused by an unacceptable choice of X by LMDER. In this case, it may be possible to remedy the situation by rerunning LMDER with a smaller value of FACTOR.

Excessive number of function evaluations. A reasonable value for MAXFEV is  $100*(N+1)$ . If the number of calls to FCN with  $\text{IFLAG} = 1$  reaches MAXFEV, then this indicates that the routine is converging very slowly as measured by the progress of FVEC, and INFO is set to 5. In this case, it may be helpful to restart LMDER with MODE set to 1.

## 6. Characteristics of the algorithm.

LMDER is a modification of the Levenberg-Marquardt algorithm.

Two of its main characteristics involve the proper use of implicitly scaled variables (if MODE = 1) and an optimal choice for the correction. The use of implicitly scaled variables achieves scale invariance of LMDER and limits the size of the correction in any direction where the functions are changing rapidly. The optimal choice of the correction guarantees (under reasonable conditions) global convergence from starting points far from the solution and a fast rate of convergence for problems with small residuals.

Timing. The time required by LMDER to solve a given problem depends on M and N, the behavior of the functions, the accuracy requested, and the starting point. The number of arithmetic operations needed by LMDER is about  $N^3$  to process each evaluation of the functions (call to FCN with IFLAG = 1) and  $M(N^2)$  to process each evaluation of the Jacobian (call to FCN with IFLAG = 2). Unless FCN can be evaluated quickly, the timing of LMDER will be strongly influenced by the time spent in FCN.

Storage. LMDER requires  $M \cdot N + 2 \cdot M + 6 \cdot N$  double precision storage locations and N integer storage locations, in addition to the storage required by the program. There are no internally declared storage arrays.

## 7. Subprograms required.

USER-supplied ..... FCN

MINPACK-supplied ... DPMPAR, ENORM, LMPAR, QRFAC, QRSOLV

FORTTRAN-supplied ... DABS, DMAX1, DMIN1, DSQRT, MOD

## 8. References.

Jorge J. More, The Levenberg-Marquardt Algorithm, Implementation and Theory. Numerical Analysis, G. A. Watson, editor. Lecture Notes in Mathematics 630, Springer-Verlag, 1977.

## 9. Example.

The problem is to determine the values of  $x(1)$ ,  $x(2)$ , and  $x(3)$  which provide the best fit (in the least squares sense) of

$$x(1) + u(i)/(v(i)*x(2) + w(i)*x(3)), \quad i = 1, 15$$

to the data

$$y = (0.14, 0.18, 0.22, 0.25, 0.29, 0.32, 0.35, 0.39, \\ 0.37, 0.58, 0.73, 0.96, 1.34, 2.10, 4.39),$$

where  $u(i) = i$ ,  $v(i) = 16 - i$ , and  $w(i) = \min(u(i), v(i))$ . The  $i$ -th component of FVEC is thus defined by

$$y(i) = (x(1) + u(i)/(v(i)*x(2) + w(i)*x(3))).$$

```

C *****
C
C DRIVER FOR LMDER EXAMPLE.
C DOUBLE PRECISION VERSION
C
C *****
C INTEGER J,M,N,LDFJAC,MAXFEV,MODE,NPRINT,INFO,NFEV,NJEV,NWRITE
C INTEGER IPVT(3)
C DOUBLE PRECISION FTOL,XTOL,GTOL,FACTOR,FNORM
C DOUBLE PRECISION X(3),FVEC(15),FJAC(15,3),DIAG(3),QTF(3),
* WA1(3),WA2(3),WA3(3),WA4(15)
C DOUBLE PRECISION ENORM,DPMPAR
C EXTERNAL FCN
C
C LOGICAL OUTPUT UNIT IS ASSUMED TO BE NUMBER 6.
C
C DATA NWRITE /6/
C
C M = 15
C N = 3
C
C THE FOLLOWING STARTING VALUES PROVIDE A ROUGH FIT.
C
C X(1) = 1.D0
C X(2) = 1.D0
C X(3) = 1.D0
C
C LDFJAC = 15
C
C SET FTOL AND XTOL TO THE SQUARE ROOT OF THE MACHINE PRECISION
C AND GTOL TO ZERO. UNLESS HIGH PRECISION SOLUTIONS ARE
C REQUIRED, THESE ARE THE RECOMMENDED SETTINGS.
C
C FTOL = DSQRT(DPMPAR(1))
C XTOL = DSQRT(DPMPAR(1))
C GTOL = 0.D0
C
C MAXFEV = 400
C MODE = 1
C FACTOR = 1.D2
C NPRINT = 0
C
C CALL LMDER(FCN,M,N,X,FVEC,FJAC,LDFJAC,FTOL,XTOL,GTOL,
* MAXFEV,DIAG,MODE,FACTOR,NPRINT,INFO,NFEV,NJEV,
* IPVT,QTF,WA1,WA2,WA3,WA4)
C FNORM = ENORM(M,FVEC)
C WRITE (NWRITE,1000) FNORM,NFEV,NJEV,INFO,(X(J),J=1,N)
C STOP
1000 FORMAT (5X,31H FINAL L2 NORM OF THE RESIDUALS,D15.7 //

```

```

*      5X,31H NUMBER OF FUNCTION EVALUATIONS,I10 //
*      5X,31H NUMBER OF JACOBIAN EVALUATIONS,I10 //
*      5X,15H EXIT PARAMETER,16X,I10 //
*      5X,27H FINAL APPROXIMATE SOLUTION // 5X,3D15.7)

```

C  
C  
C

LAST CARD OF DRIVER FOR LMDER EXAMPLE.

```

END
SUBROUTINE FCN(M,N,X,FVEC,FJAC,LDFJAC,IFLAG)
INTEGER M,N,LDFJAC,IFLAG
DOUBLE PRECISION X(N),FVEC(M),FJAC(LDFJAC,N)

```

C  
C  
C

SUBROUTINE FCN FOR LMDER EXAMPLE.

```

INTEGER I
DOUBLE PRECISION TMP1,TMP2,TMP3,TMP4
DOUBLE PRECISION Y(15)
DATA Y(1),Y(2),Y(3),Y(4),Y(5),Y(6),Y(7),Y(8),
*      Y(9),Y(10),Y(11),Y(12),Y(13),Y(14),Y(15)
*      /1.4D-1,1.8D-1,2.2D-1,2.5D-1,2.9D-1,3.2D-1,3.5D-1,3.9D-1,
*      3.7D-1,5.8D-1,7.3D-1,9.6D-1,1.34D0,2.1D0,4.39D0/

```

C

IF (IFLAG .NE. 0) GO TO 5

C

INSERT PRINT STATEMENTS HERE WHEN NPRINT IS POSITIVE.

C

C

```

RETURN
5 CONTINUE
IF (IFLAG .EQ. 2) GO TO 20
DO 10 I = 1, 15
  TMP1 = I
  TMP2 = 16 - I
  TMP3 = TMP1
  IF (I .GT. 8) TMP3 = TMP2
  FVEC(I) = Y(I) - (X(1) + TMP1/(X(2)*TMP2 + X(3)*TMP3))
10 CONTINUE
GO TO 40
20 CONTINUE
DO 30 I = 1, 15
  TMP1 = I
  TMP2 = 16 - I
  TMP3 = TMP1
  IF (I .GT. 8) TMP3 = TMP2
  TMP4 = (X(2)*TMP2 + X(3)*TMP3)**2
  FJAC(I,1) = -1.D0
  FJAC(I,2) = TMP1*TMP2/TMP4
  FJAC(I,3) = TMP1*TMP3/TMP4
30 CONTINUE
40 CONTINUE
RETURN

```

C

C

C

LAST CARD OF SUBROUTINE FCN.

END

Results obtained with different compilers or machines  
may be slightly different.

FINAL L2 NORM OF THE RESIDUALS 0.9063596D-01

NUMBER OF FUNCTION EVALUATIONS 6

NUMBER OF JACOBIAN EVALUATIONS 5

EXIT PARAMETER 1

FINAL APPROXIMATE SOLUTION

0.8241058D-01 0.1133037D+01 0.2343695D+01



## Documentation for MINPACK subroutine LMSTR1

Double precision version

Argonne National Laboratory

Burton S. Garbow, Kenneth E. Hillstom, Jorge J. More

March 1980

## 1. Purpose.

The purpose of LMSTR1 is to minimize the sum of the squares of M nonlinear functions in N variables by a modification of the Levenberg-Marquardt algorithm which uses minimal storage. This is done by using the more general least-squares solver LMSTR. The user must provide a subroutine which calculates the functions and the rows of the Jacobian.

## 2. Subroutine and type statements.

```

SUBROUTINE LMSTR1(FCN,M,N,X,FVEC,FJAC,LDFJAC,TOL,
*                INFO,IPVT,WA,LWA)
  INTEGER M,N,LDFJAC,INFO,LWA
  INTEGER IPVT(N)
  DOUBLE PRECISION TOL
  DOUBLE PRECISION X(N),FVEC(M),FJAC(LDFJAC,N),WA(LWA)
  EXTERNAL FCN

```

## 3. Parameters.

Parameters designated as input parameters must be specified on entry to LMSTR1 and are not changed on exit, while parameters designated as output parameters need not be specified on entry and are set to appropriate values on exit from LMSTR1.

FCN is the name of the user-supplied subroutine which calculates the functions and the rows of the Jacobian. FCN must be declared in an EXTERNAL statement in the user calling program, and should be written as follows.

```

SUBROUTINE FCN(M,N,X,FVEC,FJROW,IFLAG)
  INTEGER M,N,IFLAG
  DOUBLE PRECISION X(N),FVEC(M),FJROW(N)
  -----
  IF IFLAG = 1 CALCULATE THE FUNCTIONS AT X AND
  RETURN THIS VECTOR IN FVEC.
  IF IFLAG = I CALCULATE THE (I-1)-ST ROW OF THE
  JACOBIAN AT X AND RETURN THIS VECTOR IN FJROW.
  -----
  RETURN

```

END

The value of IFLAG should not be changed by FCN unless the user wants to terminate execution of LMSTR1. In this case set IFLAG to a negative integer.

M is a positive integer input variable set to the number of functions.

N is a positive integer input variable set to the number of variables. N must not exceed M.

X is an array of length N. On input X must contain an initial estimate of the solution vector. On output X contains the final estimate of the solution vector.

FVEC is an output array of length M which contains the functions evaluated at the output X.

FJAC is an output N by N array. The upper triangle of FJAC contains an upper triangular matrix R such that

$$P^T (JAC^T JAC) P = R^T R,$$

where P is a permutation matrix and JAC is the final calculated Jacobian. Column j of P is column IPVT(j) (see below) of the identity matrix. The lower triangular part of FJAC contains information generated during the computation of R.

LDFJAC is a positive integer input variable not less than N which specifies the leading dimension of the array FJAC.

TOL is a nonnegative input variable. Termination occurs when the algorithm estimates either that the relative error in the sum of squares is at most TOL or that the relative error between X and the solution is at most TOL. Section 4 contains more details about TOL.

INFO is an integer output variable. If the user has terminated execution, INFO is set to the (negative) value of IFLAG. See description of FCN. Otherwise, INFO is set as follows.

INFO = 0 Improper input parameters.

INFO = 1 Algorithm estimates that the relative error in the sum of squares is at most TOL.

INFO = 2 Algorithm estimates that the relative error between X and the solution is at most TOL.

INFO = 3 Conditions for INFO = 1 and INFO = 2 both hold.

INFO = 4 FVEC is orthogonal to the columns of the Jacobian to

machine precision.

INFO = 5 Number of calls to FCN with IFLAG = 1 has reached  $100*(N+1)$ .

INFO = 6 TOL is too small. No further reduction in the sum of squares is possible.

INFO = 7 TOL is too small. No further improvement in the approximate solution X is possible.

Sections 4 and 5 contain more details about INFO.

IPVT is an integer output array of length N. IPVT defines a permutation matrix P such that  $JAC*P = Q*R$ , where JAC is the final calculated Jacobian, Q is orthogonal (not stored), and R is upper triangular. Column j of P is column IPVT(j) of the identity matrix.

WA is a work array of length LWA.

LWA is a positive integer input variable not less than  $5*N+M$ .

#### 4. Successful completion.

The accuracy of LMSTR1 is controlled by the convergence parameter TOL. This parameter is used in tests which make three types of comparisons between the approximation X and a solution XSOL. LMSTR1 terminates when any of the tests is satisfied. If TOL is less than the machine precision (as defined by the MINPACK function DPMPAR(1)), then LMSTR1 only attempts to satisfy the test defined by the machine precision. Further progress is not usually possible. Unless high precision solutions are required, the recommended value for TOL is the square root of the machine precision.

The tests assume that the functions and the Jacobian are coded consistently, and that the functions are reasonably well behaved. If these conditions are not satisfied, then LMSTR1 may incorrectly indicate convergence. The coding of the Jacobian can be checked by the MINPACK subroutine CHKDER. If the Jacobian is coded correctly, then the validity of the answer can be checked, for example, by rerunning LMSTR1 with a tighter tolerance.

First convergence test. If ENORM(Z) denotes the Euclidean norm of a vector Z, then this test attempts to guarantee that

$$ENORM(FVEC) \leq (1+TOL)*ENORM(FVECS),$$

where FVECS denotes the functions evaluated at XSOL. If this condition is satisfied with  $TOL = 10^{*(-K)}$ , then the final residual norm ENORM(FVEC) has K significant decimal digits and

INFO is set to 1 (or to 3 if the second test is also satisfied).

Second convergence test. If D is a diagonal matrix (implicitly generated by LMSTR1) whose entries contain scale factors for the variables, then this test attempts to guarantee that

$$\text{ENORM}(D*(X-XSOL)) \leq \text{TOL} * \text{ENORM}(D*XSOL).$$

If this condition is satisfied with  $\text{TOL} = 10^{**(-K)}$ , then the larger components of  $D*X$  have K significant decimal digits and INFO is set to 2 (or to 3 if the first test is also satisfied). There is a danger that the smaller components of  $D*X$  may have large relative errors, but the choice of D is such that the accuracy of the components of X is usually related to their sensitivity.

Third convergence test. This test is satisfied when FVEC is orthogonal to the columns of the Jacobian to machine precision. There is no clear relationship between this test and the accuracy of LMSTR1, and furthermore, the test is equally well satisfied at other critical points, namely maximizers and saddle points. Therefore, termination caused by this test (INFO = 4) should be examined carefully.

## 5. Unsuccessful completion.

Unsuccessful termination of LMSTR1 can be due to improper input parameters, arithmetic interrupts, or an excessive number of function evaluations.

Improper input parameters. INFO is set to 0 if  $N \leq 0$ , or  $M < N$ , or  $\text{LDFJAC} < N$ , or  $\text{TOL} < 0$ , or  $\text{LWA} < 5*N+M$ .

Arithmetic interrupts. If these interrupts occur in the FCN subroutine during an early stage of the computation, they may be caused by an unacceptable choice of X by LMSTR1. In this case, it may be possible to remedy the situation by not evaluating the functions here, but instead setting the components of FVEC to numbers that exceed those in the initial FVEC, thereby indirectly reducing the step length. The step length can be more directly controlled by using instead LMSTR, which includes in its calling sequence the step-length-governing parameter FACTOR.

Excessive number of function evaluations. If the number of calls to FCN with IFLAG = 1 reaches  $100*(N+1)$ , then this indicates that the routine is converging very slowly as measured by the progress of FVEC, and INFO is set to 5. In this case, it may be helpful to restart LMSTR1, thereby forcing it to disregard old (and possibly harmful) information.

## 6. Characteristics of the algorithm.

LMSTR1 is a modification of the Levenberg-Marquardt algorithm. Two of its main characteristics involve the proper use of implicitly scaled variables and an optimal choice for the correction. The use of implicitly scaled variables achieves scale invariance of LMSTR1 and limits the size of the correction in any direction where the functions are changing rapidly. The optimal choice of the correction guarantees (under reasonable conditions) global convergence from starting points far from the solution and a fast rate of convergence for problems with small residuals.

**Timing.** The time required by LMSTR1 to solve a given problem depends on M and N, the behavior of the functions, the accuracy requested, and the starting point. The number of arithmetic operations needed by LMSTR1 is about  $N^3$  to process each evaluation of the functions (call to FCN with IFLAG = 1) and  $1.5(N^2)$  to process each row of the Jacobian (call to FCN with IFLAG .GE. 2). Unless FCN can be evaluated quickly, the timing of LMSTR1 will be strongly influenced by the time spent in FCN.

**Storage.** LMSTR1 requires  $N^2 + 2M + 6N$  double precision storage locations and N integer storage locations, in addition to the storage required by the program. There are no internally declared storage arrays.

## 7. Subprograms required.

USER-supplied ..... FCN

MINPACK-supplied ... DPMPAR, ENORM, LMSTR, LMPAR, QRFAC, QRSOLV, RWUPDT

FORTTRAN-supplied ... DABS, DMAX1, DMIN1, DSQRT, MOD

## 8. References.

Jorge J. More, The Levenberg-Marquardt Algorithm, Implementation and Theory. Numerical Analysis, G. A. Watson, editor. Lecture Notes in Mathematics 630, Springer-Verlag, 1977.

## 9. Example.

The problem is to determine the values of  $x(1)$ ,  $x(2)$ , and  $x(3)$  which provide the best fit (in the least squares sense) of

$$x(1) + u(i)/(v(i)*x(2) + w(i)*x(3)), \quad i = 1, 15$$

to the data

$$y = (0.14, 0.18, 0.22, 0.25, 0.29, 0.32, 0.35, 0.39, \\ 0.37, 0.58, 0.73, 0.96, 1.34, 2.10, 4.39),$$

where  $u(i) = i$ ,  $v(i) = 16 - i$ , and  $w(i) = \min(u(i), v(i))$ . The  $i$ -th component of FVEC is thus defined by

$$y(i) - (x(1) + u(i)/(v(i)*x(2) + w(i)*x(3))).$$

```

C *****
C
C DRIVER FOR LMSTR1 EXAMPLE.
C DOUBLE PRECISION VERSION
C
C *****
C INTEGER J,M,N,LDFJAC,INFO,LWA,NWRITE
C INTEGER IPV T(3)
C DOUBLE PRECISION TOL, FNORM
C DOUBLE PRECISION X(3), FVEC(15), FJAC(3,3), WA(30)
C DOUBLE PRECISION ENORM, DPMPAR
C EXTERNAL FCN
C
C LOGICAL OUTPUT UNIT IS ASSUMED TO BE NUMBER 6.
C
C DATA NWRITE /6/
C
C M = 15
C N = 3
C
C THE FOLLOWING STARTING VALUES PROVIDE A ROUGH FIT.
C
C X(1) = 1.DO
C X(2) = 1.DO
C X(3) = 1.DO
C
C LDFJAC = 3
C LWA = 30
C
C SET TOL TO THE SQUARE ROOT OF THE MACHINE PRECISION.
C UNLESS HIGH PRECISION SOLUTIONS ARE REQUIRED,
C THIS IS THE RECOMMENDED SETTING.
C
C TOL = DSQRT(DPMPAR(1))
C
C CALL LMSTR1(FCN,M,N,X,FVEC,FJAC,LDFJAC,TOL,
*          INFO,IPVT,WA,LWA)
C ENORM = ENORM(M,FVEC)
C WRITE (NWRITE,1000) FNORM,INFO,(X(J),J=1,N)
C STOP
1000 FORMAT (5X,31H FINAL L2 NORM OF THE RESIDUALS,D15.7 //
*          5X,15H EXIT PARAMETER,16X,I10 //
*          5X,27H FINAL APPROXIMATE SOLUTION // 5X,3D15.7)
C

```

```

C      LAST CARD OF DRIVER FOR LMSTR1 EXAMPLE.
C
      END
      SUBROUTINE FCN(M,N,X,FVEC,FJROW,IFLAG)
      INTEGER M,N,IFLAG
      DOUBLE PRECISION X(N),FVEC(M),FJROW(N)
C
C      SUBROUTINE FCN FOR LMSTR1 EXAMPLE.
C
      INTEGER I
      DOUBLE PRECISION TMP1,TMP2,TMP3,TMP4
      DOUBLE PRECISION Y(15)
      DATA Y(1),Y(2),Y(3),Y(4),Y(5),Y(6),Y(7),Y(8),
*         Y(9),Y(10),Y(11),Y(12),Y(13),Y(14),Y(15)
*         /1.4D-1,1.8D-1,2.2D-1,2.5D-1,2.9D-1,3.2D-1,3.5D-1,3.9D-1,
*         3.7D-1,5.8D-1,7.3D-1,9.6D-1,1.34D0,2.1D0,4.39D0/
C
      IF (IFLAG .GE. 2) GO TO 20
      DO 10 I = 1, 15
         TMP1 = I
         TMP2 = 16 - I
         TMP3 = TMP1
         IF (I .GT. 8) TMP3 = TMP2
         FVEC(I) = Y(I) - (X(1) + TMP1/(X(2)*TMP2 + X(3)*TMP3))
10      CONTINUE
         GO TO 40
20      CONTINUE
         I = IFLAG - 1
         TMP1 = I
         TMP2 = 16 - I
         TMP3 = TMP1
         IF (I .GT. 8) TMP3 = TMP2
         TMP4 = (X(2)*TMP2 + X(3)*TMP3)**2
         FJROW(1) = -1.D0
         FJROW(2) = TMP1*TMP2/TMP4
         FJROW(3) = TMP1*TMP3/TMP4
30      CONTINUE
40      CONTINUE
         RETURN
C
C      LAST CARD OF SUBROUTINE FCN.
C
      END

Results obtained with different compilers or machines
may be slightly different.

FINAL L2 NORM OF THE RESIDUALS  0.9063596D-01

EXIT PARAMETER                    1

FINAL APPROXIMATE SOLUTION

0.8241058D-01  0.1133037D+01  0.2343695D+01

```



## Documentation for MINPACK subroutine LMSTR

Double precision version

Argonne National Laboratory

Burton S. Garbow, Kenneth E. Hillstrom, Jorge J. More

March 1980

## 1. Purpose.

The purpose of LMSTR is to minimize the sum of the squares of M nonlinear functions in N variables by a modification of the Levenberg-Marquardt algorithm which uses minimal storage. The user must provide a subroutine which calculates the functions and the rows of the Jacobian.

## 2. Subroutine and type statements.

```

SUBROUTINE LMSTR(FCN,M,N,X,FVEC,FJAC,LDFJAC,FTOL,XTOL,GTOL,
*             MAXFEV,DIAG,MODE,FACTOR,NPRINT,INFO,NFEV,NJEV,
*             IPVT,QTF,WA1,WA2,WA3,WA4)
INTEGER M,N,LDFJAC,MAXFEV,MODE,NPRINT,INFO,NFEV,NJEV
INTEGER IPVT(N)
DOUBLE PRECISION FTOL,XTOL,GTOL,FACTOR
DOUBLE PRECISION X(N),FVEC(M),FJAC(LDFJAC,N),DIAG(N),QTF(N),
*             WA1(N),WA2(N),WA3(N),WA4(M)

```

## 3. Parameters.

Parameters designated as input parameters must be specified on entry to LMSTR and are not changed on exit, while parameters designated as output parameters need not be specified on entry and are set to appropriate values on exit from LMSTR.

FCN is the name of the user-supplied subroutine which calculates the functions and the rows of the Jacobian. FCN must be declared in an EXTERNAL statement in the user calling program, and should be written as follows.

```

SUBROUTINE FCN(M,N,X,FVEC,FJROW,IFLAG)
INTEGER M,N,IFLAG
DOUBLE PRECISION X(N),FVEC(M),FJROW(N)
-----
IF IFLAG = 1 CALCULATE THE FUNCTIONS AT X AND
RETURN THIS VECTOR IN FVEC.
IF IFLAG = I CALCULATE THE (I-1)-ST ROW OF THE
JACOBIAN AT X AND RETURN THIS VECTOR IN FJROW.
-----
RETURN

```

END

The value of IFLAG should not be changed by FCN unless the user wants to terminate execution of LMSTR. In this case set IFLAG to a negative integer.

M is a positive integer input variable set to the number of functions.

N is a positive integer input variable set to the number of variables. N must not exceed M.

X is an array of length N. On input X must contain an initial estimate of the solution vector. On output X contains the final estimate of the solution vector.

FVEC is an output array of length M which contains the functions evaluated at the output X.

FJAC is an output N by N array. The upper triangle of FJAC contains an upper triangular matrix R such that

$$P^T (JAC^T JAC) P = R^T R,$$

where P is a permutation matrix and JAC is the final calculated Jacobian. Column j of P is column IPVT(j) (see below) of the identity matrix. The lower triangular part of FJAC contains information generated during the computation of R.

LDEFJAC is a positive integer input variable not less than N which specifies the leading dimension of the array FJAC.

FTOL is a nonnegative input variable. Termination occurs when both the actual and predicted relative reductions in the sum of squares are at most FTOL. Therefore, FTOL measures the relative error desired in the sum of squares. Section 4 contains more details about FTOL.

XTOL is a nonnegative input variable. Termination occurs when the relative error between two consecutive iterates is at most XTOL. Therefore, XTOL measures the relative error desired in the approximate solution. Section 4 contains more details about XTOL.

GTOL is a nonnegative input variable. Termination occurs when the cosine of the angle between FVEC and any column of the Jacobian is at most GTOL in absolute value. Therefore, GTOL measures the orthogonality desired between the function vector and the columns of the Jacobian. Section 4 contains more details about GTOL.

MAXFEV is a positive integer input variable. Termination occurs when the number of calls to FCN with IFLAG = 1 has reached

MAXFEV.

DIAG is an array of length N. If MODE = 1 (see below), DIAG is internally set. If MODE = 2, DIAG must contain positive entries that serve as multiplicative scale factors for the variables.

MODE is an integer input variable. If MODE = 1, the variables will be scaled internally. If MODE = 2, the scaling is specified by the input DIAG. Other values of MODE are equivalent to MODE = 1.

FACTOR is a positive input variable used in determining the initial step bound. This bound is set to the product of FACTOR and the Euclidean norm of DIAG\*X if nonzero, or else to FACTOR itself. In most cases FACTOR should lie in the interval (.1,100.). 100. is a generally recommended value.

NPRINT is an integer input variable that enables controlled printing of iterates if it is positive. In this case, FCN is called with IFLAG = 0 at the beginning of the first iteration and every NPRINT iterations thereafter and immediately prior to return, with X and FVEC available for printing. If NPRINT is not positive, no special calls of FCN with IFLAG = 0 are made.

INFO is an integer output variable. If the user has terminated execution, INFO is set to the (negative) value of IFLAG. See description of FCN. Otherwise, INFO is set as follows.

INFO = 0 Improper input parameters.

INFO = 1 Both actual and predicted relative reductions in the sum of squares are at most FTOL.

INFO = 2 Relative error between two consecutive iterates is at most XTOL.

INFO = 3 Conditions for INFO = 1 and INFO = 2 both hold.

INFO = 4 The cosine of the angle between FVEC and any column of the Jacobian is at most GTOL in absolute value.

INFO = 5 Number of calls to FCN with IFLAG = 1 has reached MAXFEV.

INFO = 6 FTOL is too small. No further reduction in the sum of squares is possible.

INFO = 7 XTOL is too small. No further improvement in the approximate solution X is possible.

INFO = 8 GTOL is too small. FVEC is orthogonal to the columns of the Jacobian to machine precision.

Sections 4 and 5 contain more details about INFO.

NFEV is an integer output variable set to the number of calls to FCN with IFLAG = 1.

NJEV is an integer output variable set to the number of calls to FCN with IFLAG = 2.

IPVT is an integer output array of length N. IPVT defines a permutation matrix P such that  $JAC * P = Q * R$ , where JAC is the final calculated Jacobian, Q is orthogonal (not stored), and R is upper triangular. Column j of P is column IPVT(j) of the identity matrix.

QTF is an output array of length N which contains the first N elements of the vector  $(Q \text{ transpose}) * FVEC$ .

WA1, WA2, and WA3 are work arrays of length N.

WA4 is a work array of length M.

#### 4. Successful completion.

The accuracy of LMSTR is controlled by the convergence parameters FTOL, XTOL, and GTOL. These parameters are used in tests which make three types of comparisons between the approximation X and a solution XSOL. LMSTR terminates when any of the tests is satisfied. If any of the convergence parameters is less than the machine precision (as defined by the MINPACK function DPMPAR(1)), then LMSTR only attempts to satisfy the test defined by the machine precision. Further progress is not usually possible.

The tests assume that the functions and the Jacobian are coded consistently, and that the functions are reasonably well behaved. If these conditions are not satisfied, then LMSTR may incorrectly indicate convergence. The coding of the Jacobian can be checked by the MINPACK subroutine CHKDER. If the Jacobian is coded correctly, then the validity of the answer can be checked, for example, by rerunning LMSTR with tighter tolerances.

First convergence test. If ENORM(Z) denotes the Euclidean norm of a vector Z, then this test attempts to guarantee that

$$ENORM(FVEC) \leq (1+FTOL) * ENORM(FVECS),$$

where FVECS denotes the functions evaluated at XSOL. If this condition is satisfied with  $FTOL = 10^{*(-K)}$ , then the final residual norm ENORM(FVEC) has K significant decimal digits and INFO is set to 1 (or to 3 if the second test is also satisfied). Unless high precision solutions are required, the recommended value for FTOL is the square root of the machine

precision.

Second convergence test. If D is the diagonal matrix whose entries are defined by the array DIAG, then this test attempts to guarantee that

$$\text{ENORM}(D*(X-XSOL)) \leq \text{XTOL} * \text{ENORM}(D*XSOL).$$

If this condition is satisfied with  $\text{XTOL} = 10^{*(-K)}$ , then the larger components of  $D*X$  have K significant decimal digits and INFO is set to 2 (or to 3 if the first test is also satisfied). There is a danger that the smaller components of  $D*X$  may have large relative errors, but if  $\text{MODE} = 1$ , then the accuracy of the components of X is usually related to their sensitivity. Unless high precision solutions are required, the recommended value for XTOL is the square root of the machine precision.

Third convergence test. This test is satisfied when the cosine of the angle between FVEC and any column of the Jacobian at X is at most GTOL in absolute value. There is no clear relationship between this test and the accuracy of LMSTR, and furthermore, the test is equally well satisfied at other critical points, namely maximizers and saddle points. Therefore, termination caused by this test ( $\text{INFO} = 4$ ) should be examined carefully. The recommended value for GTOL is zero.

## 5. Unsuccessful completion.

Unsuccessful termination of LMSTR can be due to improper input parameters, arithmetic interrupts, or an excessive number of function evaluations.

Improper input parameters. INFO is set to 0 if  $N \leq 0$ , or  $M < N$ , or  $\text{LDFJAC} < N$ , or  $\text{FTOL} < 0.00$ , or  $\text{XTOL} < 0.00$ , or  $\text{GTOL} < 0.00$ , or  $\text{MAXFEV} \leq 0$ , or  $\text{FACTOR} \leq 0.00$ .

Arithmetic interrupts. If these interrupts occur in the FCN subroutine during an early stage of the computation, they may be caused by an unacceptable choice of X by LMSTR. In this case, it may be possible to remedy the situation by rerunning LMSTR with a smaller value of FACTOR.

Excessive number of function evaluations. A reasonable value for MAXFEV is  $100*(N+1)$ . If the number of calls to FCN with  $\text{IFLAG} = 1$  reaches MAXFEV, then this indicates that the routine is converging very slowly as measured by the progress of FVEC, and INFO is set to 5. In this case, it may be helpful to restart LMSTR with MODE set to 1.

## 6. Characteristics of the algorithm.

LMSTR is a modification of the Levenberg-Marquardt algorithm. Two of its main characteristics involve the proper use of implicitly scaled variables (if MODE = 1) and an optimal choice for the correction. The use of implicitly scaled variables achieves scale invariance of LMSTR and limits the size of the correction in any direction where the functions are changing rapidly. The optimal choice of the correction guarantees (under reasonable conditions) global convergence from starting points far from the solution and a fast rate of convergence for problems with small residuals.

**Timing.** The time required by LMSTR to solve a given problem depends on M and N, the behavior of the functions, the accuracy requested, and the starting point. The number of arithmetic operations needed by LMSTR is about  $N^3$  to process each evaluation of the functions (call to FCN with IFLAG = 1) and  $1.5(N^2)$  to process each row of the Jacobian (call to FCN with IFLAG .GE. 2). Unless FCN can be evaluated quickly, the timing of LMSTR will be strongly influenced by the time spent in FCN.

**Storage.** LMSTR requires  $N^2 + 2M + 6N$  double precision storage locations and N integer storage locations, in addition to the storage required by the program. There are no internally declared storage arrays.

## 7. Subprograms required.

USER-supplied ..... FCN

MINPACK-supplied ... DPMPAR, ENORM, LMPAR, QRFAC, QRSOLV, RWUPDT

FORTTRAN-supplied ... DABS, DMAX1, DMIN1, DSQRT, MOD

## 8. References.

Jorge J. More, The Levenberg-Marquardt Algorithm, Implementation and Theory. Numerical Analysis, G. A. Watson, editor. Lecture Notes in Mathematics 630, Springer-Verlag, 1977.

## 9. Example.

The problem is to determine the values of  $x(1)$ ,  $x(2)$ , and  $x(3)$  which provide the best fit (in the least squares sense) of

$$x(1) + u(i)/(v(i)*x(2) + w(i)*x(3)), \quad i = 1, 15$$

to the data

$$y = (0.14, 0.18, 0.22, 0.25, 0.29, 0.32, 0.35, 0.39, \\ 0.37, 0.58, 0.73, 0.96, 1.34, 2.10, 4.39),$$

where  $u(i) = i$ ,  $v(i) = 16 - i$ , and  $w(i) = \min(u(i), v(i))$ . The  $i$ -th component of FVEC is thus defined by

$$y(i) = (x(1) + u(i)/(v(i)*x(2) + w(i)*x(3))).$$

```

C *****
C
C DRIVER FOR LMSTR EXAMPLE.
C DOUBLE PRECISION VERSION
C
C *****
C INTEGER J, M, N, LDFJAC, MAXFEV, MODE, NPRINT, INFO, NFEV, NJEV, NWRITE
C INTEGER IPVT(3)
C DOUBLE PRECISION FTOL, XTOL, GTOL, FACTOR, FNORM
C DOUBLE PRECISION X(3), FVEC(15), FJAC(3,3), DIAG(3), QTF(3),
* WA1(3), WA2(3), WA3(3), WA4(15)
C DOUBLE PRECISION ENORM, DPMPAR
C EXTERNAL FCN
C
C LOGICAL OUTPUT UNIT IS ASSUMED TO BE NUMBER 6.
C
C DATA NWRITE /6/
C
C M = 15
C N = 3
C
C THE FOLLOWING STARTING VALUES PROVIDE A ROUGH FIT.
C
C X(1) = 1.D0
C X(2) = 1.D0
C X(3) = 1.D0
C
C LDFJAC = 3
C
C SET FTOL AND XTOL TO THE SQUARE ROOT OF THE MACHINE PRECISION
C AND GTOL TO ZERO. UNLESS HIGH PRECISION SOLUTIONS ARE
C REQUIRED, THESE ARE THE RECOMMENDED SETTINGS.
C
C FTOL = DSQRT(DPMPAR(1))
C XTOL = DSQRT(DPMPAR(1))
C GTOL = 0.D0
C
C MAXFEV = 400
C MODE = 1
C FACTOR = 1.D2
C NPRINT = 0
C
C CALL LMSTR(FCN, M, N, X, FVEC, FJAC, LDFJAC, FTOL, XTOL, GTOL,
* MAXFEV, DIAG, MODE, FACTOR, NPRINT, INFO, NFEV, NJEV,
* IPVT, QTF, WA1, WA2, WA3, WA4)
C FNORM = ENORM(M, FVEC)
C WRITE (NWRITE, 1000) FNORM, NFEV, NJEV, INFO, (X(J), J=1, N)
C STOP
1000 FORMAT (5X, 31H FINAL L2 NORM OF THE RESIDUALS, D15.7 //

```

```

*      5X,31H NUMBER OF FUNCTION EVALUATIONS,I10 //
*      5X,31H NUMBER OF JACOBIAN EVALUATIONS,I10 //
*      5X,15H EXIT PARAMETER,16X,I10 //
*      5X,27H FINAL APPROXIMATE SOLUTION // 5X,3D15.7)

```

C  
C  
C

LAST CARD OF DRIVER FOR LMSTR EXAMPLE.

```

END
SUBROUTINE FCN(M,N,X,FVEC,FJROW,IFLAG)
INTEGER M,N,IFLAG
DOUBLE PRECISION X(N),FVEC(M),FJROW(N)

```

C  
C  
C

SUBROUTINE FCN FOR LMSTR EXAMPLE.

```

INTEGER I
DOUBLE PRECISION TMP1,TMP2,TMP3,TMP4
DOUBLE PRECISION Y(15)
DATA Y(1),Y(2),Y(3),Y(4),Y(5),Y(6),Y(7),Y(8),
*      Y(9),Y(10),Y(11),Y(12),Y(13),Y(14),Y(15)
*      /1.4D-1,1.8D-1,2.2D-1,2.5D-1,2.9D-1,3.2D-1,3.5D-1,3.9D-1,
*      3.7D-1,5.8D-1,7.3D-1,9.6D-1,1.34D0,2.1D0,4.39D0/

```

C

IF (IFLAG .NE. 0) GO TO 5

C  
C  
C

INSERT PRINT STATEMENTS HERE WHEN NPRINT IS POSITIVE.

```

RETURN
5 CONTINUE
IF (IFLAG .GE. 2) GO TO 20
DO 10 I = 1, 15
  TMP1 = I
  TMP2 = 16 - I
  TMP3 = TMP1
  IF (I .GT. 8) TMP3 = TMP2
  FVEC(I) = Y(I) - (X(1) + TMP1/(X(2)*TMP2 + X(3)*TMP3))
10 CONTINUE
GO TO 40
20 CONTINUE
I = IFLAG - 1
  TMP1 = I
  TMP2 = 16 - I
  TMP3 = TMP1
  IF (I .GT. 8) TMP3 = TMP2
  TMP4 = (X(2)*TMP2 + X(3)*TMP3)**2
  FJROW(1) = -1.DO
  FJROW(2) = TMP1*TMP2/TMP4
  FJROW(3) = TMP1*TMP3/TMP4
30 CONTINUE
40 CONTINUE
RETURN

```

C  
C  
C

LAST CARD OF SUBROUTINE FCN.

END

Results obtained with different compilers or machines  
may be slightly different.

FINAL L2 NORM OF THE RESIDUALS 0.9063596D-01

NUMBER OF FUNCTION EVALUATIONS 6

NUMBER OF JACOBIAN EVALUATIONS 5

EXIT PARAMETER 1

FINAL APPROXIMATE SOLUTION

0.8241058D-01 0.1133037D+01 0.2343695D+01



## Documentation for MINPACK subroutine LMDIF1

Double precision version

Argonne National Laboratory

Burton S. Garbow, Kenneth E. Hillstrom, Jorge J. More

March 1980

## 1. Purpose.

The purpose of LMDIF1 is to minimize the sum of the squares of M nonlinear functions in N variables by a modification of the Levenberg-Marquardt algorithm. This is done by using the more general least-squares solver LMDIF. The user must provide a subroutine which calculates the functions. The Jacobian is then calculated by a forward-difference approximation.

## 2. Subroutine and type statements.

```

SUBROUTINE LMDIF1(FCN,M,N,X,FVEC,TOL,INFO,IWA,WA,LWA)
INTEGER M,N,INFO,LWA
INTEGER IWA(N)
DOUBLE PRECISION TOL
DOUBLE PRECISION X(N),FVEC(M),WA(LWA)
EXTERNAL FCN

```

## 3. Parameters.

Parameters designated as input parameters must be specified on entry to LMDIF1 and are not changed on exit, while parameters designated as output parameters need not be specified on entry and are set to appropriate values on exit from LMDIF1.

FCN is the name of the user-supplied subroutine which calculates the functions. FCN must be declared in an EXTERNAL statement in the user calling program, and should be written as follows.

```

SUBROUTINE FCN(M,N,X,FVEC,IFLAG)
INTEGER M,N,IFLAG
DOUBLE PRECISION X(N),FVEC(M)
-----
CALCULATE THE FUNCTIONS AT X AND
RETURN THIS VECTOR IN FVEC.
-----
RETURN
END

```

The value of IFLAG should not be changed by FCN unless the user wants to terminate execution of LMDIF1. In this case set

IFLAG to a negative integer.

M is a positive integer input variable set to the number of functions.

N is a positive integer input variable set to the number of variables. N must not exceed M.

X is an array of length N. On input X must contain an initial estimate of the solution vector. On output X contains the final estimate of the solution vector.

FVEC is an output array of length M which contains the functions evaluated at the output X.

TOL is a nonnegative input variable. Termination occurs when the algorithm estimates either that the relative error in the sum of squares is at most TOL or that the relative error between X and the solution is at most TOL. Section 4 contains more details about TOL.

INFO is an integer output variable. If the user has terminated execution, INFO is set to the (negative) value of IFLAG. See description of FCN. Otherwise, INFO is set as follows.

INFO = 0 Improper input parameters.

INFO = 1 Algorithm estimates that the relative error in the sum of squares is at most TOL.

INFO = 2 Algorithm estimates that the relative error between X and the solution is at most TOL.

INFO = 3 Conditions for INFO = 1 and INFO = 2 both hold.

INFO = 4 FVEC is orthogonal to the columns of the Jacobian to machine precision.

INFO = 5 Number of calls to FCN has reached or exceeded  $200*(N+1)$ .

INFO = 6 TOL is too small. No further reduction in the sum of squares is possible.

INFO = 7 TOL is too small. No further improvement in the approximate solution X is possible.

Sections 4 and 5 contain more details about INFO.

IWA is an integer work array of length N.

WA is a work array of length LWA.

LWA is a positive integer input variable not less than

$M*N+5*N+M$ .

#### 4. Successful completion.

The accuracy of LMDIF1 is controlled by the convergence parameter TOL. This parameter is used in tests which make three types of comparisons between the approximation X and a solution XSOL. LMDIF1 terminates when any of the tests is satisfied. If TOL is less than the machine precision (as defined by the MINPACK function DPMPAR(1)), then LMDIF1 only attempts to satisfy the test defined by the machine precision. Further progress is not usually possible. Unless high precision solutions are required, the recommended value for TOL is the square root of the machine precision.

The tests assume that the functions are reasonably well behaved. If this condition is not satisfied, then LMDIF1 may incorrectly indicate convergence. The validity of the answer can be checked, for example, by rerunning LMDIF1 with a tighter tolerance.

First convergence test. If ENORM(Z) denotes the Euclidean norm of a vector Z, then this test attempts to guarantee that

$$\text{ENORM(FVEC)} \leq (1+\text{TOL}) * \text{ENORM(FVECS)},$$

where FVECS denotes the functions evaluated at XSOL. If this condition is satisfied with  $\text{TOL} = 10^{**(-K)}$ , then the final residual norm ENORM(FVEC) has K significant decimal digits and INFO is set to 1 (or to 3 if the second test is also satisfied).

Second convergence test. If D is a diagonal matrix (implicitly generated by LMDIF1) whose entries contain scale factors for the variables, then this test attempts to guarantee that

$$\text{ENORM}(D*(X-\text{XSOL})) \leq \text{TOL} * \text{ENORM}(D*\text{XSOL}).$$

If this condition is satisfied with  $\text{TOL} = 10^{**(-K)}$ , then the larger components of  $D*X$  have K significant decimal digits and INFO is set to 2 (or to 3 if the first test is also satisfied). There is a danger that the smaller components of  $D*X$  may have large relative errors, but the choice of D is such that the accuracy of the components of X is usually related to their sensitivity.

Third convergence test. This test is satisfied when FVEC is orthogonal to the columns of the Jacobian to machine precision. There is no clear relationship between this test and the accuracy of LMDIF1, and furthermore, the test is equally well satisfied at other critical points, namely maximizers and saddle points. Also, errors in the functions (see below) may result in the test being satisfied at a point not close to the

minimum. Therefore, termination caused by this test (INFO = 4) should be examined carefully.

#### 5. Unsuccessful completion.

Unsuccessful termination of LMDIF1 can be due to improper input parameters, arithmetic interrupts, an excessive number of function evaluations, or errors in the functions.

Improper input parameters. INFO is set to 0 if  $N \leq 0$ , or  $M \leq N$ , or  $TOL \leq 0$ , or  $LWA \leq M*N+5*N+M$ .

Arithmetic interrupts. If these interrupts occur in the FCN subroutine during an early stage of the computation, they may be caused by an unacceptable choice of X by LMDIF1. In this case, it may be possible to remedy the situation by not evaluating the functions here, but instead setting the components of FVEC to numbers that exceed those in the initial FVEC, thereby indirectly reducing the step length. The step length can be more directly controlled by using instead LMDIF, which includes in its calling sequence the step-length-governing parameter FACTOR.

Excessive number of function evaluations. If the number of calls to FCN reaches  $200*(N+1)$ , then this indicates that the routine is converging very slowly as measured by the progress of FVEC, and INFO is set to 5. In this case, it may be helpful to restart LMDIF1, thereby forcing it to disregard old (and possibly harmful) information.

Errors in the functions. The choice of step length in the forward-difference approximation to the Jacobian assumes that the relative errors in the functions are of the order of the machine precision. If this is not the case, LMDIF1 may fail (usually with INFO = 4). The user should then use LMDIF instead, or one of the programs which require the analytic Jacobian (LMDER1 and LMDER).

#### 6. Characteristics of the algorithm.

LMDIF1 is a modification of the Levenberg-Marquardt algorithm. Two of its main characteristics involve the proper use of implicitly scaled variables and an optimal choice for the correction. The use of implicitly scaled variables achieves scale invariance of LMDIF1 and limits the size of the correction in any direction where the functions are changing rapidly. The optimal choice of the correction guarantees (under reasonable conditions) global convergence from starting points far from the solution and a fast rate of convergence for problems with small residuals.

Timing. The time required by LMDIF1 to solve a given problem

depends on M and N, the behavior of the functions, the accuracy requested, and the starting point. The number of arithmetic operations needed by LMDIF1 is about  $N^3$  to process each evaluation of the functions (one call to FCN) and  $M(N^2)$  to process each approximation to the Jacobian (N calls to FCN). Unless FCN can be evaluated quickly, the timing of LMDIF1 will be strongly influenced by the time spent in FCN.

Storage. LMDIF1 requires  $M \cdot N + 2 \cdot M + 6 \cdot N$  double precision storage locations and N integer storage locations, in addition to the storage required by the program. There are no internally declared storage arrays.

#### 7. Subprograms required.

USER-supplied ..... FCN

MINPACK-supplied ... DPMPAR, ENORM, FDJAC2, LMDIF, LMPAR,  
QRFAC, QRSOLV

FORTTRAN-supplied ... DABS, DMAX1, DMIN1, DSQRT, MOD

#### 8. References.

Jorge J. More, The Levenberg-Marquardt Algorithm, Implementation and Theory. Numerical Analysis, G. A. Watson, editor. Lecture Notes in Mathematics 630, Springer-Verlag, 1977.

#### 9. Example.

The problem is to determine the values of  $x(1)$ ,  $x(2)$ , and  $x(3)$  which provide the best fit (in the least squares sense) of

$$x(1) + u(i)/(v(i) \cdot x(2) + w(i) \cdot x(3)), \quad i = 1, 15$$

to the data

$$y = (0.14, 0.18, 0.22, 0.25, 0.29, 0.32, 0.35, 0.39, \\ 0.37, 0.58, 0.73, 0.96, 1.34, 2.10, 4.39),$$

where  $u(i) = i$ ,  $v(i) = 16 - i$ , and  $w(i) = \min(u(i), v(i))$ . The  $i$ -th component of FVEC is thus defined by

$$y(i) - (x(1) + u(i)/(v(i) \cdot x(2) + w(i) \cdot x(3))).$$

```
C *****
C
C DRIVER FOR LMDIF1 EXAMPLE.
C DOUBLE PRECISION VERSION
C
```

```

C      *****
      INTEGER J,M,N,INFO,LWA,NWRITE
      INTEGER IWA(3)
      DOUBLE PRECISION TOL, FNORM
      DOUBLE PRECISION X(3), FVEC(15), WA(75)
      DOUBLE PRECISION ENORM, DPMPAR
      EXTERNAL FCN

C
C      LOGICAL OUTPUT UNIT IS ASSUMED TO BE NUMBER 6.
C
C      DATA NWRITE /6/

C      M = 15
      N = 3

C
C      THE FOLLOWING STARTING VALUES PROVIDE A ROUGH FIT.
C
      X(1) = 1.DO
      X(2) = 1.DO
      X(3) = 1.DO

C
      LWA = 75

C
C      SET TOL TO THE SQUARE ROOT OF THE MACHINE PRECISION.
      UNLESS HIGH PRECISION SOLUTIONS ARE REQUIRED,
      THIS IS THE RECOMMENDED SETTING.
C
      TOL = DSQRT(DPMPAR(1))

C
      CALL LMDIF1(FCN,M,N,X,FVEC,TOL,INFO,IWA,WA,LWA)
      FNORM = ENORM(M,FVEC)
      WRITE (NWRITE,1000) FNORM,INFO,(X(J),J=1,N)
      STOP
1000 FORMAT (5X,31H FINAL L2 NORM OF THE RESIDUALS,D15.7 //
*          5X,15H EXIT PARAMETER,16X,I10 //
*          5X,27H FINAL APPROXIMATE SOLUTION // 5X,3D15.7)

C
C      LAST CARD OF DRIVER FOR LMDIF1 EXAMPLE.
C
      END
      SUBROUTINE FCN(M,N,X,FVEC,IFLAG)
      INTEGER M,N,IFLAG
      DOUBLE PRECISION X(N),FVEC(M)

C
C      SUBROUTINE FCN FOR LMDIF1 EXAMPLE.
C
      INTEGER I
      DOUBLE PRECISION TMP1,TMP2,TMP3
      DOUBLE PRECISION Y(15)
      DATA Y(1),Y(2),Y(3),Y(4),Y(5),Y(6),Y(7),Y(8),
*          Y(9),Y(10),Y(11),Y(12),Y(13),Y(14),Y(15)
*          /1.4D-1,1.8D-1,2.2D-1,2.5D-1,2.9D-1,3.2D-1,3.5D-1,3.9D-1,
*          3.7D-1,5.8D-1,7.3D-1,9.6D-1,1.34D0,2.1D0,4.39D0/
C

```

```
DO 10 I = 1, 15
  TMP1 = I
  TMP2 = 16 - I
  TMP3 = TMP1
  IF (I .GT. 8) TMP3 = TMP2
  FVEC(I) = Y(I) - (X(1) + TMP1/(X(2)*TMP2 + X(3)*TMP3))
10  CONTINUE
  RETURN
```

C  
C  
C

LAST CARD OF SUBROUTINE FCN.

END

Results obtained with different compilers or machines  
may be slightly different.

FINAL L2 NORM OF THE RESIDUALS 0.9063596D-01

EXIT PARAMETER 1

FINAL APPROXIMATE SOLUTION

0.8241057D-01 0.1133037D+01 0.2343695D+01



## Documentation for MINPACK subroutine LMDIF

Double precision version

Argonne National Laboratory

Burton S. Garbow, Kenneth E. Hillstrom, Jorge J. More

March 1980

## 1. Purpose.

The purpose of LMDIF is to minimize the sum of the squares of M nonlinear functions in N variables by a modification of the Levenberg-Marquardt algorithm. The user must provide a subroutine which calculates the functions. The Jacobian is then calculated by a forward-difference approximation.

## 2. Subroutine and type statements.

```

SUBROUTINE LMDIF(FCN,M,N,X,FVEC,FTOL,XTOL,GTOL,MAXFEV,EPSFCN,
*              DIAG,MODE,FACTOR,NPRINT,INFO,NEEV,FJAC,LDFJAC,
*              IPVT,QTF,WA1,WA2,WA3,WA4)
INTEGER M,N,MAXFEV,MODE,NPRINT,INFO,NEEV,LDFJAC
INTEGER IPVT(N)
DOUBLE PRECISION FTOL,XTOL,GTOL,EPSFCN,FACTOR
DOUBLE PRECISION X(N),FVEC(M),DIAG(N),FJAC(LDFJAC,N),QTF(N),
*              WA1(N),WA2(N),WA3(N),WA4(M)
EXTERNAL FCN

```

## 3. Parameters.

Parameters designated as input parameters must be specified on entry to LMDIF and are not changed on exit, while parameters designated as output parameters need not be specified on entry and are set to appropriate values on exit from LMDIF.

FCN is the name of the user-supplied subroutine which calculates the functions. FCN must be declared in an EXTERNAL statement in the user calling program, and should be written as follows.

```

SUBROUTINE FCN(M,N,X,FVEC,IFLAG)
INTEGER M,N,IFLAG
DOUBLE PRECISION X(N),FVEC(M)
-----
CALCULATE THE FUNCTIONS AT X AND
RETURN THIS VECTOR IN FVEC.
-----
RETURN
END

```

The value of IFLAG should not be changed by FCN unless the user wants to terminate execution of LMDIF. In this case set IFLAG to a negative integer.

M is a positive integer input variable set to the number of functions.

N is a positive integer input variable set to the number of variables. N must not exceed M.

X is an array of length N. On input X must contain an initial estimate of the solution vector. On output X contains the final estimate of the solution vector.

FVEC is an output array of length M which contains the functions evaluated at the output X.

FTOL is a nonnegative input variable. Termination occurs when both the actual and predicted relative reductions in the sum of squares are at most FTOL. Therefore, FTOL measures the relative error desired in the sum of squares. Section 4 contains more details about FTOL.

XTOL is a nonnegative input variable. Termination occurs when the relative error between two consecutive iterates is at most XTOL. Therefore, XTOL measures the relative error desired in the approximate solution. Section 4 contains more details about XTOL.

GTOL is a nonnegative input variable. Termination occurs when the cosine of the angle between FVEC and any column of the Jacobian is at most GTOL in absolute value. Therefore, GTOL measures the orthogonality desired between the function vector and the columns of the Jacobian. Section 4 contains more details about GTOL.

MAXFEV is a positive integer input variable. Termination occurs when the number of calls to FCN is at least MAXFEV by the end of an iteration.

EPSEFCN is an input variable used in determining a suitable step for the forward-difference approximation. This approximation assumes that the relative errors in the functions are of the order of EPSEFCN. If EPSEFCN is less than the machine precision, it is assumed that the relative errors in the functions are of the order of the machine precision.

DIAG is an array of length N. If MODE = 1 (see below), DIAG is internally set. If MODE = 2, DIAG must contain positive entries that serve as multiplicative scale factors for the variables.

MODE is an integer input variable. If MODE = 1, the variables will be scaled internally. If MODE = 2, the scaling is

specified by the input DIAG. Other values of MODE are equivalent to MODE = 1.

FACTOR is a positive input variable used in determining the initial step bound. This bound is set to the product of FACTOR and the Euclidean norm of  $DIAG \cdot X$  if nonzero, or else to FACTOR itself. In most cases FACTOR should lie in the interval (.1,100.). 100. is a generally recommended value.

NPRINT is an integer input variable that enables controlled printing of iterates if it is positive. In this case, FCN is called with IFLAG = 0 at the beginning of the first iteration and every NPRINT iterations thereafter and immediately prior to return, with X and FVEC available for printing. If NPRINT is not positive, no special calls of FCN with IFLAG = 0 are made.

INFO is an integer output variable. If the user has terminated execution, INFO is set to the (negative) value of IFLAG. See description of FCN. Otherwise, INFO is set as follows.

- INFO = 0 Improper input parameters.
- INFO = 1 Both actual and predicted relative reductions in the sum of squares are at most FTOL.
- INFO = 2 Relative error between two consecutive iterates is at most XTOL.
- INFO = 3 Conditions for INFO = 1 and INFO = 2 both hold.
- INFO = 4 The cosine of the angle between FVEC and any column of the Jacobian is at most GTOL in absolute value.
- INFO = 5 Number of calls to FCN has reached or exceeded MAXFEV.
- INFO = 6 FTOL is too small. No further reduction in the sum of squares is possible.
- INFO = 7 XTOL is too small. No further improvement in the approximate solution X is possible.
- INFO = 8 GTOL is too small. FVEC is orthogonal to the columns of the Jacobian to machine precision.

Sections 4 and 5 contain more details about INFO.

NEEV is an integer output variable set to the number of calls to FCN.

EJAC is an output M by N array. The upper N by N submatrix of EJAC contains an upper triangular matrix R with diagonal elements of nonincreasing magnitude such that

$$P^T (JAC^T JAC) P = R^T R,$$

where P is a permutation matrix and JAC is the final calculated Jacobian. Column j of P is column IPVT(j) (see below) of the identity matrix. The lower trapezoidal part of FJAC contains information generated during the computation of R.

LDEFJAC is a positive integer input variable not less than M which specifies the leading dimension of the array FJAC.

IPVT is an integer output array of length N. IPVT defines a permutation matrix P such that  $JAC^T P = Q^T R$ , where JAC is the final calculated Jacobian, Q is orthogonal (not stored), and R is upper triangular with diagonal elements of nonincreasing magnitude. Column j of P is column IPVT(j) of the identity matrix.

QTF is an output array of length N which contains the first N elements of the vector  $(Q^T \text{transpose})^T FVEC$ .

WA1, WA2, and WA3 are work arrays of length N.

WA4 is a work array of length M.

#### 4. Successful completion.

The accuracy of LMDIF is controlled by the convergence parameters FTOL, XTOL, and GTOL. These parameters are used in tests which make three types of comparisons between the approximation X and a solution XSOL. LMDIF terminates when any of the tests is satisfied. If any of the convergence parameters is less than the machine precision (as defined by the MINPACK function DPMPAR(1)), then LMDIF only attempts to satisfy the test defined by the machine precision. Further progress is not usually possible.

The tests assume that the functions are reasonably well behaved. If this condition is not satisfied, then LMDIF may incorrectly indicate convergence. The validity of the answer can be checked, for example, by rerunning LMDIF with tighter tolerances.

First convergence test. If ENORM(Z) denotes the Euclidean norm of a vector Z, then this test attempts to guarantee that

$$ENORM(FVEC) \leq (1+FTOL) * ENORM(FVECS),$$

where FVECS denotes the functions evaluated at XSOL. If this condition is satisfied with  $FTOL = 10^{(-K)}$ , then the final residual norm ENORM(FVEC) has K significant decimal digits and INFO is set to 1 (or to 3 if the second test is also satisfied). Unless high precision solutions are required, the

recommended value for FTOL is the square root of the machine precision.

Second convergence test. If D is the diagonal matrix whose entries are defined by the array DIAG, then this test attempts to guarantee that

$$\text{ENORM}(D*(X-XSOL)) \leq \text{XTOL} * \text{ENORM}(D*XSOL).$$

If this condition is satisfied with  $\text{XTOL} = 10^{*(-K)}$ , then the larger components of  $D*X$  have K significant decimal digits and INFO is set to 2 (or to 3 if the first test is also satisfied). There is a danger that the smaller components of  $D*X$  may have large relative errors, but if  $\text{MODE} = 1$ , then the accuracy of the components of X is usually related to their sensitivity. Unless high precision solutions are required, the recommended value for XTOL is the square root of the machine precision.

Third convergence test. This test is satisfied when the cosine of the angle between FVEC and any column of the Jacobian at X is at most GTOL in absolute value. There is no clear relationship between this test and the accuracy of LMDIF, and furthermore, the test is equally well satisfied at other critical points, namely maximizers and saddle points. Therefore, termination caused by this test ( $\text{INFO} = 4$ ) should be examined carefully. The recommended value for GTOL is zero.

## 5. Unsuccessful completion.

Unsuccessful termination of LMDIF can be due to improper input parameters, arithmetic interrupts, or an excessive number of function evaluations.

Improper input parameters. INFO is set to 0 if  $N \leq 0$ , or  $M \leq N$ , or  $\text{LDFJAC} \leq M$ , or  $\text{FTOL} \leq 0.00$ , or  $\text{XTOL} \leq 0.00$ , or  $\text{GTOL} \leq 0.00$ , or  $\text{MAXFEV} \leq 0$ , or  $\text{FACTOR} \leq 0.00$ .

Arithmetic interrupts. If these interrupts occur in the FCN subroutine during an early stage of the computation, they may be caused by an unacceptable choice of X by LMDIF. In this case, it may be possible to remedy the situation by rerunning LMDIF with a smaller value of FACTOR.

Excessive number of function evaluations. A reasonable value for MAXFEV is  $200*(N+1)$ . If the number of calls to FCN reaches MAXFEV, then this indicates that the routine is converging very slowly as measured by the progress of FVEC, and INFO is set to 5. In this case, it may be helpful to restart LMDIF with MODE set to 1.

## 6. Characteristics of the algorithm.

LMDIF is a modification of the Levenberg-Marquardt algorithm. Two of its main characteristics involve the proper use of implicitly scaled variables (if MODE = 1) and an optimal choice for the correction. The use of implicitly scaled variables achieves scale invariance of LMDIF and limits the size of the correction in any direction where the functions are changing rapidly. The optimal choice of the correction guarantees (under reasonable conditions) global convergence from starting points far from the solution and a fast rate of convergence for problems with small residuals.

**Timing.** The time required by LMDIF to solve a given problem depends on M and N, the behavior of the functions, the accuracy requested, and the starting point. The number of arithmetic operations needed by LMDIF is about  $N^3$  to process each evaluation of the functions (one call to FCN) and  $M(N^2)$  to process each approximation to the Jacobian (N calls to FCN). Unless FCN can be evaluated quickly, the timing of LMDIF will be strongly influenced by the time spent in FCN.

**Storage.** LMDIF requires  $M \cdot N + 2 \cdot M + 6 \cdot N$  double precision storage locations and N integer storage locations, in addition to the storage required by the program. There are no internally declared storage arrays.

## 7. Subprograms required.

USER-supplied ..... FCN

MINPACK-supplied ... DPMPAR, ENORM, EDJAC2, LMPAR, QRFAC, QRSOLV

FORTTRAN-supplied ... DABS, DMAX1, DMIN1, DSQRT, MOD

## 8. References.

Jorge J. More, The Levenberg-Marquardt Algorithm, Implementation and Theory. Numerical Analysis, G. A. Watson, editor. Lecture Notes in Mathematics 630, Springer-Verlag, 1977.

## 9. Example.

The problem is to determine the values of  $x(1)$ ,  $x(2)$ , and  $x(3)$  which provide the best fit (in the least squares sense) of

$$x(1) + u(i)/(v(i) \cdot x(2) + w(i) \cdot x(3)), \quad i = 1, 15$$

to the data

$$y = (0.14, 0.18, 0.22, 0.25, 0.29, 0.32, 0.35, 0.39, \\ 0.37, 0.58, 0.73, 0.96, 1.34, 2.10, 4.39),$$

where  $u(i) = i$ ,  $v(i) = 16 - i$ , and  $w(i) = \min(u(i), v(i))$ . The  $i$ -th component of FVEC is thus defined by

$$y(i) - (x(1) + u(i)/(v(i)*x(2) + w(i)*x(3))).$$

```

C *****
C
C DRIVER FOR LMDIF EXAMPLE.
C DOUBLE PRECISION VERSION
C
C *****
C INTEGER J, M, N, MAXFEV, MODE, NPRINT, INFO, NFEV, LDFJAC, NWRITE
C INTEGER IPVT(3)
C DOUBLE PRECISION FTOL, XTOL, GTOL, EPSFCN, FACTOR, FNORM
C DOUBLE PRECISION X(3), FVEC(15), DIAG(3), FJAC(15, 3), QTF(3),
* WA1(3), WA2(3), WA3(3), WA4(15)
C DOUBLE PRECISION ENORM, DPMPAR
C EXTERNAL FCN
C
C LOGICAL OUTPUT UNIT IS ASSUMED TO BE NUMBER 6.
C
C DATA NWRITE /6/
C
C M = 15
C N = 3
C
C THE FOLLOWING STARTING VALUES PROVIDE A ROUGH FIT.
C
C X(1) = 1.D0
C X(2) = 1.D0
C X(3) = 1.D0
C
C LDFJAC = 15
C
C SET FTOL AND XTOL TO THE SQUARE ROOT OF THE MACHINE PRECISION
C AND GTOL TO ZERO. UNLESS HIGH PRECISION SOLUTIONS ARE
C REQUIRED, THESE ARE THE RECOMMENDED SETTINGS.
C
C FTOL = DSQRT(DPMPAR(1))
C XTOL = DSQRT(DPMPAR(1))
C GTOL = 0.D0
C
C MAXFEV = 800
C EPSFCN = 0.D0
C MODE = 1
C FACTOR = 1.D2
C NPRINT = 0
C
C CALL LMDIF(FCN, M, N, X, FVEC, FTOL, XTOL, GTOL, MAXFEV, EPSFCN,
* DIAG, MODE, FACTOR, NPRINT, INFO, NFEV, FJAC, LDFJAC,
* IPVT, QTF, WA1, WA2, WA3, WA4)

```



0.8241057D-01 0.1133037D+01 0.2343695D+01



## Documentation for MINPACK subroutine CHKDER ..

Double precision version

Argonne National Laboratory

Burton S. Garbow, Kenneth E. Hillstom, Jorge J. More

March 1980

## 1. Purpose.

The purpose of CHKDER is to check the gradients of M nonlinear functions in N variables, evaluated at a point X, for consistency with the functions themselves. The user must call CHKDER twice, first with MODE = 1 and then with MODE = 2.

## 2. Subroutine and type statements.

```

SUBROUTINE CHKDER(M,N,X,FVEC,FJAC,LDFJAC,XP,FVECP,MODE,ERR)
INTEGER M,N,LDFJAC,MODE
DOUBLE PRECISION X(N),FVEC(M),FJAC(LDFJAC,N),XP(N),FVECP(M),
*           ERR(M)

```

## 3. Parameters.

Parameters designated as input parameters must be specified on entry to CHKDER and are not changed on exit, while parameters designated as output parameters need not be specified on entry and are set to appropriate values on exit from CHKDER.

M is a positive integer input variable set to the number of functions.

N is a positive integer input variable set to the number of variables.

X is an input array of length N.

FVEC is an array of length M. On input when MODE = 2, FVEC must contain the functions evaluated at X.

FJAC is an M by N array. On input when MODE = 2, the rows of FJAC must contain the gradients of the respective functions evaluated at X.

LDFJAC is a positive integer input variable not less than M which specifies the leading dimension of the array FJAC.

XP is an array of length N. On output when MODE = 1, XP is set to a neighboring point of X.

FVECP is an array of length M. On input when MODE = 2, FVECP must contain the functions evaluated at XP.

MODE is an integer input variable set to 1 on the first call and 2 on the second. Other values of MODE are equivalent to MODE = 1.

ERR is an array of length M. On output when MODE = 2, ERR contains measures of correctness of the respective gradients. If there is no severe loss of significance, then if ERR(I) is 1.0 the I-th gradient is correct, while if ERR(I) is 0.0 the I-th gradient is incorrect. For values of ERR between 0.0 and 1.0, the categorization is less certain. In general, a value of ERR(I) greater than 0.5 indicates that the I-th gradient is probably correct, while a value of ERR(I) less than 0.5 indicates that the I-th gradient is probably incorrect.

#### 4. Successful completion.

CHKDER usually guarantees that if ERR(I) is 1.0, then the I-th gradient at X is consistent with the I-th function. This suggests that the input X be such that consistency of the gradient at X implies consistency of the gradient at all points of interest. If all the components of X are distinct and the fractional part of each one has two nonzero digits, then X is likely to be a satisfactory choice.

If ERR(I) is not 1.0 but is greater than 0.5, then the I-th gradient is probably consistent with the I-th function (the more so the larger ERR(I) is), but the conditions for ERR(I) to be 1.0 have not been completely satisfied. In this case, it is recommended that CHKDER be rerun with other input values of X. If ERR(I) is always greater than 0.5, then the I-th gradient is consistent with the I-th function.

#### 5. Unsuccessful completion.

CHKDER does not perform reliably if cancellation or rounding errors cause a severe loss of significance in the evaluation of a function. Therefore, none of the components of X should be unusually small (in particular, zero) or any other value which may cause loss of significance. The relative differences between corresponding elements of FVECP and FVEC should be at least two orders of magnitude greater than the machine precision (as defined by the MINPACK function DPMPAR(1)). If there is a severe loss of significance in the evaluation of the I-th function, then ERR(I) may be 0.0 and yet the I-th gradient could be correct.

If ERR(I) is not 0.0 but is less than 0.5, then the I-th gradient is probably not consistent with the I-th function (the more so the smaller ERR(I) is), but the conditions for ERR(I) to

be 0.0 have not been completely satisfied. In this case, it is recommended that CHKDER be rerun with other input values of X. If ERR(I) is always less than 0.5 and if there is no severe loss of significance, then the I-th gradient is not consistent with the I-th function.

## 6. Characteristics of the algorithm.

CHKDER checks the I-th gradient for consistency with the I-th function by computing a forward-difference approximation along a suitably chosen direction and comparing this approximation with the user-supplied gradient along the same direction. The principal characteristic of CHKDER is its invariance to changes in scale of the variables or functions.

**Timing.** The time required by CHKDER depends only on M and N. The number of arithmetic operations needed by CHKDER is about N when MODE = 1 and M\*N when MODE = 2.

**Storage.** CHKDER requires  $M*N + 3*M + 2*N$  double precision storage locations, in addition to the storage required by the program. There are no internally declared storage arrays.

## 7. Subprograms required.

MINPACK-supplied ... DPMPAR

FORTRAN-supplied ... DABS, DLOG10, DSQRT

## 8. References.

None.

## 9. Example.

This example checks the Jacobian matrix for the problem that determines the values of  $x(1)$ ,  $x(2)$ , and  $x(3)$  which provide the best fit (in the least squares sense) of

$$x(1) + u(i)/(v(i)*x(2) + w(i)*x(3)), \quad i = 1, 15$$

to the data

$$y = (0.14, 0.18, 0.22, 0.25, 0.29, 0.32, 0.35, 0.39, \\ 0.37, 0.58, 0.73, 0.96, 1.34, 2.10, 4.39),$$

where  $u(i) = i$ ,  $v(i) = 16 - i$ , and  $w(i) = \min(u(i), v(i))$ . The i-th component of FVEC is thus defined by

$$y(i) - (x(1) + u(i)/(v(i)*x(2) + w(i)*x(3))).$$

```

C      *****
C
C      DRIVER FOR CHKDER EXAMPLE.
C      DOUBLE PRECISION VERSION
C
C      *****
C      INTEGER I,M,N,LDFJAC,MODE,NWRITE
C      DOUBLE PRECISION X(3),FVEC(15),FJAC(15,3),XP(3),FVECP(15),
*      ERR(15)
C
C      LOGICAL OUTPUT UNIT IS ASSUMED TO BE NUMBER 6.
C
C      DATA NWRITE /6/
C
C      M = 15
C      N = 3
C
C      THE FOLLOWING VALUES SHOULD BE SUITABLE FOR
C      CHECKING THE JACOBIAN MATRIX.
C
C      X(1) = 9.2D-1
C      X(2) = 1.3D-1
C      X(3) = 5.4D-1
C
C      LDFJAC = 15
C
C      MODE = 1
C      CALL CHKDER(M,N,X,FVEC,FJAC,LDFJAC,XP,FVECP,MODE,ERR)
C      MODE = 2
C      CALL FCN(M,N,X,FVEC,FJAC,LDFJAC,1)
C      CALL FCN(M,N,X,FVEC,FJAC,LDFJAC,2)
C      CALL FCN(M,N,XP,FVECP,FJAC,LDFJAC,1)
C      CALL CHKDER(M,N,X,FVEC,FJAC,LDFJAC,XP,FVECP,MODE,ERR)
C
C      DO 10 I = 1, M
C          FVECP(I) = FVECP(I) - FVEC(I)
10      CONTINUE
C          WRITE (NWRITE,1000) (FVEC(I),I=1,M)
C          WRITE (NWRITE,2000) (FVECP(I),I=1,M)
C          WRITE (NWRITE,3000) (ERR(I),I=1,M)
C          STOP
1000  FORMAT (/5X,5H FVEC // (5X,3D15.7))
2000  FORMAT (/5X,13H FVECP - FVEC // (5X,3D15.7))
3000  FORMAT (/5X,4H ERR // (5X,3D15.7))
C
C      LAST CARD OF DRIVER FOR CHKDER EXAMPLE.
C
C      END
C      SUBROUTINE FCN(M,N,X,FVEC,FJAC,LDFJAC,IFLAG)
C      INTEGER M,N,LDFJAC,IFLAG
C      DOUBLE PRECISION X(N),FVEC(M),FJAC(LDFJAC,N)
C
C      SUBROUTINE FCN FOR CHKDER EXAMPLE.
C

```

```

INTEGER I
DOUBLE PRECISION TMP1,TMP2,TMP3,TMP4
DOUBLE PRECISION Y(15)
DATA Y(1),Y(2),Y(3),Y(4),Y(5),Y(6),Y(7),Y(8),
*     Y(9),Y(10),Y(11),Y(12),Y(13),Y(14),Y(15)
*     /1.4D-1,1.8D-1,2.2D-1,2.5D-1,2.9D-1,3.2D-1,3.5D-1,3.9D-1,
*     3.7D-1,5.8D-1,7.3D-1,9.6D-1,1.34D0,2.1D0,4.39D0/

```

C

```

IF (IFLAG .EQ. 2) GO TO 20
DO 10 I = 1, 15
  TMP1 = I
  TMP2 = 16 - I
  TMP3 = TMP1
  IF (I .GT. 8) TMP3 = TMP2
  FVEC(I) = Y(I) - (X(1) + TMP1/(X(2)*TMP2 + X(3)*TMP3))
10  CONTINUE
GO TO 40
20 CONTINUE
DO 30 I = 1, 15
  TMP1 = I
  TMP2 = 16 - I

```

C

C

C

C

```

ERROR INTRODUCED INTO NEXT STATEMENT FOR ILLUSTRATION.
CORRECTED STATEMENT SHOULD READ      TMP3 = TMP1 .

```

```

  TMP3 = TMP2
  IF (I .GT. 8) TMP3 = TMP2
  TMP4 = (X(2)*TMP2 + X(3)*TMP3)**2
  FJAC(I,1) = -1.D0
  FJAC(I,2) = TMP1*TMP2/TMP4
  FJAC(I,3) = TMP1*TMP3/TMP4
30  CONTINUE
40 CONTINUE
RETURN

```

C

C

C

```

LAST CARD OF SUBROUTINE FCN.

```

```

END

```

Results obtained with different compilers or machines may be different. In particular, the differences FVECP - FVEC are machine dependent.

```

FVEC

```

```

-0.1181606D+01 -0.1429655D+01 -0.1606344D+01
-0.1745269D+01 -0.1840654D+01 -0.1921586D+01
-0.1984141D+01 -0.2022537D+01 -0.2468977D+01
-0.2827562D+01 -0.3473582D+01 -0.4437612D+01
-0.6047662D+01 -0.9267761D+01 -0.1891806D+02

```

```

FVECP - FVEC

```

```

-0.7724666D-08 -0.3432405D-08 -0.2034843D-09

```

0.2313685D-08	0.4331078D-08	0.5984096D-08
0.7363281D-08	0.8531470D-08	0.1488591D-07
0.2335850D-07	0.3522012D-07	0.5301255D-07
0.8266660D-07	0.1419747D-06	0.3198990D-06

ERR

0.1141397D+00	0.9943516D-01	0.9674474D-01
0.9980447D-01	0.1073116D+00	0.1220445D+00
0.1526814D+00	0.1000000D+01	0.1000000D+01
0.1000000D+01	0.1000000D+01	0.1000000D+01
0.1000000D+01	0.1000000D+01	0.1000000D+01