GRADMIN — **A PACKAGE FOR FUNCTION MINIMIZATION USING EFFICIENT GRADIENT METHODS**

PURPOSE

To optimize (minimize or maximize) any well behaved continuous unconstrained objective function of the form $F = f(x_1, x_2, \ldots, x_n)$, where $x_1, x_2, \ldots, x_n$ are independent variables. Objective functions subject to equality and/or inequality constraints may also be optimized by suitably transforming the constrained objective function into an artificial unconstrained objective function.

AUTHORS

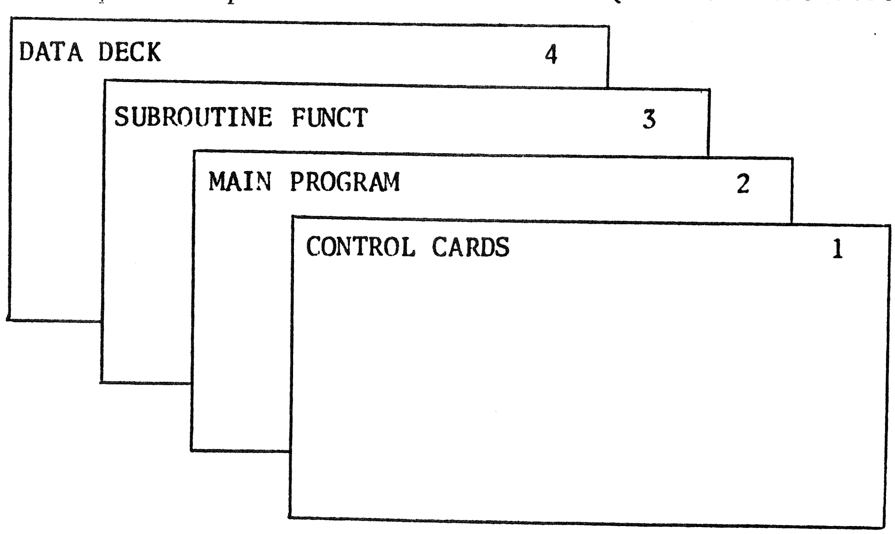
J.W. Bandler and V.K. Jha
Communications Research Laboratory
Department of Electrical Engineering
McMaster University
Hamilton, Ontario, Canada

PROGRAMMER

V.K. Jha

HOW TO USE

Set up the input deck as follows (McMaster CDC 6400)



1. <u>CONTROL CARDS</u> Use the following set of control cards.

```
AAAA                                              USER NAME
LOADER(PPLOADR)
ATTACH(TAPE, GRADMIN, ID=*********, MR=1)†
RUN(S)
SETINDF.
LOAD(TAPE)
REDUCE.
LGO.
END OF RECORD
PROGRAM TST(INPUT, OUTPUT, TAPE5 = INPUT, TAPE6 = OUTPUT)
```

2. <u>MAIN PROGRAM</u> Write the main program as indicated below

(a) Dimension the following arrays

X(NN), G(NN), X1(NN), X2(NN), G2(NN), ALFA(NN), P(NN,NN), Y(NN), PY(NN)

PE(NN), BIGV(NN), XSTRT(NN), DUM1(NN), DUM2(NN), EPS(N), H(K)

where N = The number of independent variables

NN = N + 2

K = N*(N+7)/2

(b) Supply the values of the following parameters

N = The number of independent variables

NN = N + 2

*KR = 1 if data deck is to be read

= 0 if data deck is not to be read always set it equal to 1 for the first call of GRADMIN.

---

*Parameter KR allows handling of problems with constraints, when the Fiacco-McCormick [2] method is used. Since many optimization runs are then necessary KR is set=0, after the first call of GRADMIN, so that for subsequent calls of GRADMIN data is not read and required again.

†Appropriate identification parameter ID should be inserted in *********.

(c) Call subroutine GRADMIN as follows

```
CALL GRADMIN(N, X, XSTRT, X2, X1, G, G2, ALFA, H, P, PY, PE, BIGV,
             EPS, NN, Y, DUM1, DUM2, KR)
```

(d) Add STOP and END cards.

3. SUBROUTINE FUNCT Write subroutine FUNCT as follows

```
SUBROUTINE FUNCT (N, X, F, G)
DIMENSION X(1), G(1)
```

$F = f(x_1, x_2, \ldots, x_n)$

$G(1) = \partial f/\partial x_1$

$G(2) = \partial f/\partial x_2$

$\vdots$

$G(N) = \partial f/\partial x_n$

```
RETURN
END
```

If other statements are required for defining the function or the gradients, they may be added to this subroutine, e.g., function F may be defined in another subprogram which may then be called by subroutine FUNCT.

4. DATA DECK Parameters to be supplied as data are defined below

MET1 First method to be called by GRADMIN

if MET1 = 1 New Fletcher method will be called

if MET1 = 2 Jacobson-Oksman method will be called

if MET1 = 3 Fletcher-Powell method will be called

if MET1 = 0 No first method will be called

MET2 Second method to be called by GRADMIN

if MET2 = 1 New Fletcher method will be called

if MET2 = 2 Jacobson-Oksman method will be called if MET2 = 3 Fletcher-Powell method will be called

if MET2 = 0 No second method will be called

MET 3	Third method to be called by GRADMIN if MET3 = 1 New Fletcher method will be called if MET3 = 2 Jacobson-Oksman method will be called if MET3 = 3 Fletcher-Powell method will be called if MET3 = 0 No third method will be called
M	A parameter to select the starting point; if M=1 same starting point is used by all the methods called; for any other value of M each method starts with the optimum left by the last method
MAX	Maximum number of permissible iterations
MODE	A parameter to choose the stopping criterion for Jacobson-Oksman method; if MODE=1 criterion will be $\Delta F \leq$ a small number if MODE=2 criterion will be $\|\|gradients\|\| \leq$ a small number
IPRINT	Intermediate output is printed out every IPRINT iterations; it should be set=0 if no intermediate output is desired.
IDATA	Input data is printed out if IDATA=1; it should be set=0 if input data is not to be printed out
EST	Minimum estimated value of the objective function
EPS1	Small test quantity used by the Fletcher-Powell method
ETA(I),I=1,4	Test quantities used by the Jacobson-Oksman method
EPS(I),I=1,N	Test quantities used by the New Fletcher method
XSTRT(I), I=1,NN	Starting values for variables x_1, x_2, ..., x_n and x_{n+1}, x_{n+2}. Two extra variables x_{n+1} and x_{n+2} are required by the Jacobson-Oksman method. Suggested starting values

for x_{n+1} and x_{n+2} are the estimated order of the objective function and the minimum estimated value of the function, respectively.

Recommended values for some of the parameters are

MAX = 100

EPS1 = 1.E-6

ETA(1) = 1.E-4

ETA(2) = 1.E-8

ETA(3) = ETA(4) = 1.E-16

EPS(I), I=1,N Each = 1.E-6

EST — A lower bound on the minimum value of the objective function may be obtained from physical reasons, if the true minimum is not known, e.g., for approximation problems 0.0 is convenient.

Setting up the data deck

Card No.	Format	Parameters
1	8I5	MET1, MET2, MET3, M, MAX, MODE, IPRINT, IDATA
2	5E16.8	EPS1, (ETA(I), I=1,4)
As many as required	5E16.8	EST, (EPS(I), I,N)
As many as required	5E16.8	(XSTRT(I), I=1,NN)

COMMENTS

As explained above, any number: one, two or all the three methods, and in any order may be called depending upon the values of MET1, MET2 and MET3 chosen by the user. Similarly, by choosing appropriate values of IPRINT and IDATA, the user may or may not printout the input data and intermediate output. Problems with the constraints may be solved using the Fiacco-McCormick technique as illustrated in Appendix B. All the subroutines have variable dimensioning and there is no limit on the maximum number of variables which can be handled. The numerical technique of perturbation may be used to find the gradients it it is not possible to calculate the gradients analytically. Suitable diagonastic messages are printed out whenever there is any unusual exit. To illustrate the use of this package, two problems have been solved and included in Appendix A. Appendix B shows the general structure of the package.

REFERENCES

1. A.V. Fiacco and G.P. McCormick, "The sequential unconstrained minimization technique for nonlinear programming, a primal-dual method", Management Science, vol. 10, pp. 360-366, January 1964.

2. A.V. Fiacco and G.P. McCormick, "Computational algorithm for the sequential unconstrained minimization technique for nonlinear programming", Management Science, vol. 10, pp. 601-617, July 1964.

3. R. Fletcher and M.J.D. Powell, "A rapidly convergent descent method for minimization", Computer J., vol. 6 , pp. 163-168 , June 1963.

4. R. Fletcher, "A new approach to variable metric algorithms", Computer J., vol. 13, pp. 317-322, August 1970.

5. D.H. Jacobson and W. Oksman, "An algorithm that minimizes homogeneous functions of n variables in n+2 iterations and rapidly minimizes general functions", Division of Engineering and Applied Physics, Harvard University, Cambridge, Mass., Technical Report No. 618, October 1970.

ACKNOWLEDGEMENTS

The authors acknowledge the assistance of N.D. Markettos and C. Charalambous. Dr. R. Fletcher and Dr. D.H. Jacobson made available listings of their programs.

January 28, 1972

APPENDIX - A

Problem 1 - To minimize F

$$F = 100(x_1^2 - x_2)^2 + (1 - x_1)^2$$

A suitable listing of the input deck supplied is shown in Fig. 1. Since parameters IPRINT and IDATA have been set=1 here, the output of this program had a listing of the input data, and intermediate output was printed out after each iteration. Intermediate output provides the user information about the iteration number, number of function evaluations, time elapsed, values of $x_1, x_2, \ldots x_n$ for that iteration and the values of objective function and its gradients. The result given by the Package (New Fletcher Method) is

$$F = 1.01931839E-19$$

$$x_1 = 1.00000000E+00$$

$$x_2 = 1.00000000E+00$$

A suitable listing of the input deck, and the printout of input data and final results for problem 1 are given on Figs. 1-3.

Problem 2 - To minimize U

$$U = x_1^2 + 4x_2^2$$

subject to the following equality constraint

$$\psi = x_1 + 2x_2 - 1$$

The unconstrained function for this problem using Fiacco-McCormick is

$$F = U + \psi^2/\sqrt{r}$$

A suitable listing of the input deck is shown in Fig. 4. The result given by the package (New Fletcher Method) is

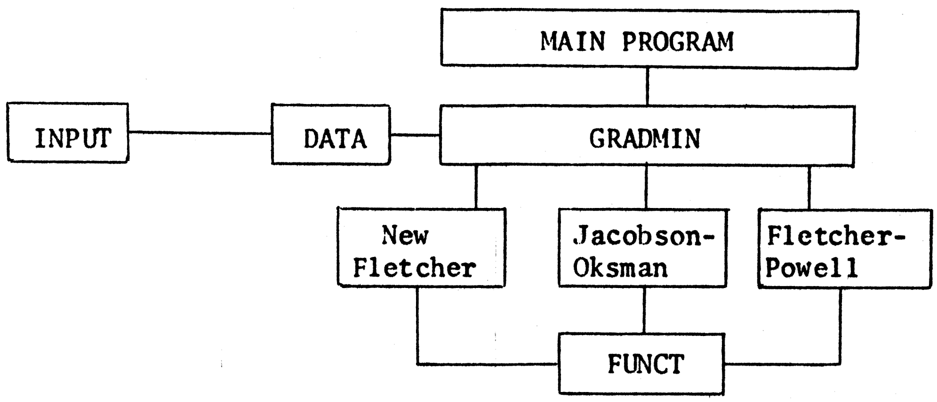
$$F = 4.99975001E-1$$

$$x_1 = 4.99975001E-1$$

$$x_2 = 2.49987501E-1$$

APPENDIX - B

The general structure of the package is



Attached is the complete fortran listing of the package.

```
HSRJ.                                                                          JHA V.
LOADER(PPLOADR)
ATTACH(TAPE,GRADMIN,ID=GABARNLER,MR=1)
RUN(S,,,,,,,,X)
SETINDF.
LOAD(TAPE)
REDUCE.
LGO.
'          6400 END OF RECORD
        PROGRAM TST (INPUT,OUTPUT,TAPE5=INPUT,TAPE6=OUTPUT)
        DIMENSION DUM1(4),DUM2(4),X(4),XSTRT(4),X2(4),X1(4),G(4),G2(4),
       1ALFA(4),H(9),P(4,4),Y(4),PY(4),PE(4),BIGV(4),EPS(2)
        N=2
        NN=N+2
        KR=1
        CALL GRADMIN(N,X,XSTRT,X2,X1,G,G2,ALFA,H,P,PY,PE,BIGV,EPS,NN,Y,
       1DUM1,DUM2,KR)
        STOP
        END
        SUBROUTINE FUNCT(N,X,F,G)
C       THIS SUBROUTINE CALCULATES THE VALUE OF THE OBJECTIVE FUNCTION AND
C       ITS PARTIAL DERIVATIVES AT ANY POINT DEFINED BY THE VECTOR X
        DIMENSION X(1),G(1)
C       FOLLOWING IS THE OBJECTIVE FUNCTION
        F=100.*(X(1)**2-X(2))**2+(1.-X(1))**2
C       GRADIENTS OF THE OBJECTIVE FUNCTION ARE GIVEN BELOW
        G(1)=400.*(X(1)**3-X(1)*X(2))-2.*(1.-X(1))
        G(2)=200.*(X(2)-X(1)**2)
        RETURN
        END
'          6400 END OF RECORD
     1     3     2     1   500     1     1     1
       1.E-        9         1.E-   9   1.E-          9     1.E-       16   1.E-        1
  0.                   1.E-         9   1.E-          9
  5.               5.                2.               0.
'              END OF FILE

                                   CD TOT    0036
```

Figure 1

```
NPUT DATA
---------

OLLOWING METHODS HAVE BEEN CALLED

EW FLETCHER METHOD
LETCHER-POWELL METHOD
ACOBSON-OKSMAN METHOD

UMBER OF INDEPENDENT VARIABLES.......................................N=    2

AXIMUM NUMBER OF ALLOWABLE ITERATIONS..............................MAX=  500

NTERMEDIATE OUTPUT TO BE PRINTED EVERY IPRINT ITERATIONS.......IPRINT=    1

TARTING POINT TO BE SAME FOR ALL THE METHODS IF M=1..................M=    1

TARTING VALUE FOR VECTOR X(I)............................XSTRT( 1)=  5.00000000E+0
                                                         XSTRT( 2)=  5.00000000E+0

EST QUANTITIES TO BE USED IN JACOBSON-OKSMAN METHOD..........ETA( 1)=  1.00000000E-0
                                                               ETA( 2)=  1.00000000E-0
                                                               ETA( 3)=  1.00000000E-1
                                                               ETA( 4)=  1.00000000E-1

EST QUANTITIES TO BE USED IN NEW FLETCHER METHOD.............EPS( 1)=  1.00000000E-0
                                                               EPS( 2)=  1.00000000E-0

EST QUANTITY TO BE USED IN FLETCHER-POWELL METHOD...............EPS1=  1.00000000E-0

STIMATE OF LOWER BOUND ON FUNCTION TO BE MINIMIZED...............EST=  0.
```

Figure 2

```
FOLLOWING IS THE OPTIMUM SOLUTION
---------------------------------

               F =  1.01931839E-19

           X( 1)=  1.00000000E+00
           X( 2)=  1.00000000E+00
NUMBER OF FUNCTION EVALUATIONS =   52

   EXECUTION TIME IN SECONDS=     .46600
```

Figure 3

New Fletcher Method

```
HSBJ.                                                                      JHA V.
LOADER(PPLOADR)
ATTACH(TAPE,GRADMIN,ID=GABABNLER,MR=1)
RUN(S,,,,,,,,X)
SETINDF.
LOAD(TAPE)
REDUCE.
LGO.
'          6400 END OF RECORD
        PROGRAM TST (INPUT,OUTPUT,TAPE5=INPUT,TAPE6=OUTPUT)
        DIMENSION DUM1(4),DUM2(4),X(4),XSTRT(4),X2(4),X1(4),G(4),G2(4),
       1ALFA(4),H(9),P(4,4),Y(4),PY(4),PE(4),BIGV(4),EPS(2)
        COMMON/SPL/R
        N=2
        NN=N+2
        KR=1
        R=1.0
        DO 1 J=1,5
        CALL GRADMIN(N,X,XSTRT,X2,X1,G,G2,ALFA,H,P,PY,PE,BIGV,EPS,NN,Y,
       1DUM1,DUM2,KR)
        KR=0
        R=R/100.
 1      CONTINUE
        STOP
        END
        SUBROUTINE FUNCT(N,X,F,G)
C       THIS SUBROUTINE CALCULATES THE VALUE OF THE OBJECTIVE FUNCTION AND
C       ITS PARTIAL DERIVATIVES AT ANY POINT DEFINED BY THE VECTOR X
C       HERE THE PROBLEM IS TO MINIMIZE THE GIVEN FUNCTION SUBJECT TO AN
C       EQUALITY CONSTRAINT
        DIMENSION X(1),G(1),GU(2),GPSI(2)
        COMMON/SPL/R
C       FOLLOWING IS THE OBJECTIVE FUNCTION
        U=X(1)**2+4.*X(2)*X(2)
C       FOLLOWING 9S THE EQUALITY CONSTRAINT
        PSI=X(1)+2.*X(2)-1.
C       THE UNCONSTRAINED FUNCTION USING FIACCO-MCCORMICK TECHNIQUE IS
C       GIVEN BELOW
        F=U+(PSI**2)/SQRT(R)
C       GRADIENTS OF THE FUNCTION ARE GIVEN BELOW
        GU(1)=2.*X(1)
        GU(2)=8.*X(2)
        GPSI(1)=1.0
        GPSI(2)=2.0
        G(1)=GU(1)+2.*PSI*GPSI(1)/SQRT(R)
        G(2)=GU(2)+2.*PSI*GPSI(2)/SQRT(R)
        RETURN
        END
'          6400 END OF RECORD
     1     3      2      1   500     1      1    1
      1.E-       9          1.E-    9   1.E-           9     1.E-       16    1.E-         1
  0.                   1.E-         9   1.E-           9
  5.               5.                  2.                0.
'            END OF FILE

                                   CD TOT    0054
```

Figure 4

```
      SUBROUTINE DATA (N,NN,XSTRT,EST,EPS1,ETA,MET1,MET2,MET3,M,MAX,MODE
     1,IPRINT,EPS)
      DIMENSION XSTRT(1), EPS(1), ETA(1)
      READ (5,1) MET1,MET2,MET3,M,MAX,MODE,IPRINT,IDATA
      READ (5,2) EPS1,(ETA(I),I=1,4)
      READ (5,2) EST,(EPS(I),I=1,N)
      READ (5,2) (XSTRT(I),I=1,NN)
      IF (IDATA.EQ.0) RETURN
      CALL INPUT (MET1,MET2,MET3,M,MAX,MODE,N,IPRINT,IDATA,EPS1,ETA,EST,
     1EPS,XSTRT)
      RETURN
C
C
C
    1 FORMAT (8I5)
    2 FORMAT (5E16.8)
      END
```

```
      SUBROUTINE FINAL (X,F,N)
      DIMENSION X(1)
      COMMON /BLK/ KO
      COMMON /BLK1/ NUMF
      WRITE (6,4)
      IF (KO.EQ.0) GO TO 1
      WRITE (6,5)
      GO TO 2
1     WRITE (6,6)
2     CONTINUE
      WRITE (6,7) F
      WRITE (6,8) (I,X(I),I=1,N)
      WRITE (6,3) NUMF
      RETURN
C
C
C
3     FORMAT (1HO,/,19X,*NUMBER OF FUNCTION EVALUATIONS =*,I5)
4     FORMAT (1H1)
5     FORMAT (1HO,40X,*FOLLOWING IS THE OPTIMUM SOLUTION*,/,41X,*-------
     1---------------------------*)
6     FORMAT (1HO,44X,*RESULTS AT LAST ITERATION*/,45X,*----------------
     1---------*)
7     FORMAT (1HO,//,48X,*F =*,E16.8,/)
8     FORMAT (1HO,44X,*X(*,I2,*)=*,E16.8)
      END
```

```
      SUBROUTINE VMM01 (N,X,F,G,H,UNITH,FEST,EPS,MAXFN,IPRINT,IEXIT,PE)
      DIMENSION X(1), G(1), H(1), EPS(1)
      DIMENSION PE(1)
      COMMON /BLK1/ NFNS
      LOGICAL CONV,UNITH
      COMMON /BLK/ KO
      CALL SECOND (T3)
      KO=0
      CALL FUNCT (N,X,F,G)
      IF (F.LT.FEST) GO TO 24
      NFNS=1
      ITN=0
      STEP=1.
      IDX=N
      IDG=N+N
      IH=IDG+N
      IF (.NOT.UNITH) GO TO 2
      IJ=IH+1
      DO 1 I=1,N
      DO 1 J=I,N
      H(IJ)=0.
      IF (I.EQ.J) H(IJ)=1.0
1     IJ=IJ+1
2     CONV=.TRUE.
      GDX=0.
      DO 6 I=1,N
      Z=0.
      IJ=IH+I
      IF (I.EQ.1) GO TO 4
      II=I-1
      DO 3 J=1,II
      Z=Z-H(IJ)*G(J)
      IJ=IJ+N-J
3     CONTINUE
4     DO 5 J=I,N
      Z=Z-H(IJ)*G(J)
      IJ=IJ+1
5     CONTINUE
      IF (ABS(Z).GT.EPS(I)) CONV=.FALSE.
      H(IDX+I)=Z
      GDX=GDX+G(I)*Z
6     CONTINUE
      DO 7 I=1,N
      PE(I)=X(I)
7     CONTINUE
C
      IF (IPRINT.EQ.0) GO TO 8
      IF (MOD(ITN,IPRINT).NE.0) GO TO 8
      CALL SECOND (T4)
      TIME=T4-T3
      CALL WRITE2 (X,N,G,F,NFNS,ITN,TIME)
8     IEXIT=1
      IF (CONV) GO TO 25
      IEXIT=2
      IF (GDX.GE.0.) GO TO 25
      Z=1.
      IF (ITN.LT.N.AND.UNITH) Z=STEP
      W=2.*(FEST-F)/GDX
      IF (W.LT.Z) Z=W
      STEP=Z
9     GDX=GDX*Z
      DO 10 I=1,N
      H(IDX+I)=H(IDX+I)*Z
      X(I)=X(I)+H(IDX+I)
10    CONTINUE
      CALL FUNCT (N,X,FP,H)
      IF (FP.LT.FEST) GO TO 24
      NFNS=NFNS+1
      IEXIT=3
      IF (ITN.EQ.MAXFN) GO TO 25
      GPDX=0.
      DO 11 I=1,N
      H(IDG+I)=H(I)-G(I)
      GPDX=GPDX+H(I)*H(IDX+I)
11    CONTINUE
      DGDX=GPDX-GDX
```

```
      IF (F.GT.FP-.0001*GDX) GO TO 13
      IEXIT=4
      IF (GPDX.LT.0..AND.ITN.GT.N) GO TO 25
      Z=3.*(F-FP)+GPDX+GDX
      W=SQRT(1.-GDX/Z*GPDX/Z)*ABS(Z)
      Z=1.-(GPDX+W-Z)/(DGDX+2.*W)
      IF (Z.LT.0.1) Z=0.1
      DO 12 I=1,N
      X(I)=X(I)-H(IDX+I)
12    CONTINUE
      GO TO 15
13    F=FP
      DO 14 I=1,N
      G(I)=H(I)
14    CONTINUE
      IF (DGDX.GT.0.) GO TO 16
      GDX=GPDX
      Z=4.
15    STEP=Z*STEP
      GO TO 9
16    IF (GPDX.LT.0.5*GDX) STEP=2.*STEP
      DGHDG=0.
      DO 20 I=1,N
      Z=0.
      IJ=IH+I
      IF (I.EQ.1) GO TO 18
      II=I-1
      DO 17 J=1,II
      Z=Z+H(IJ)*H(IDG+J)
      IJ=IJ+N-J
17    CONTINUE
18    DO 19 J=I,N
      Z=Z+H(IJ)*H(IDG+J)
      IJ=IJ+1
19    CONTINUE
      DGHDG=DGHDG+Z*H(IDG+I)
      H(I)=Z
20    CONTINUE
      IF (DGHDG.LT.0.0) DGHDG=DGDX*0.01
```

```
      IF (DGDX.LT.DGHDG) GO TO 22
      W=1.0+DGHDG/DGDX
      DO 21 I=1,N
      H(IDX+I)=W*H(IDX+I)-H(I)
21    CONTINUE
      DGDX=DGDX+DGHDG
      DGHDG=DGDX
22    IJ=IH
      DO 23 I=1,N
      W=H(IDX+I)/DGDX
      Z=H(I)/DGHDG
      DO 23 J=I,N
      IJ=IJ+1
23    H(IJ)=H(IJ)+W*H(IDX+J)-Z*H(J)
      ITN=ITN+1
      GO TO 2
24    IEXIT=5
25    IF (IEXIT.EQ.1) KO=1
      IF (IEXIT.NE.4) GO TO 27
      DO 26 I=1,N
      X(I)=PE(I)
26    CONTINUE
27    CONTINUE
      IF (IPRINT.EQ.0) RETURN
      GO TO (28,29,30,29,31), IEXIT
28    WRITE (6,33) IEXIT
      GO TO 32
29    WRITE (6,34) IEXIT
      GO TO 32
30    WRITE (6,35) IEXIT
      GO TO 32
31    WRITE (6,36) IEXIT
32    CONTINUE
      RETURN
C
C
C
33    FORMAT (/,1HO,*IEXIT=*,I2,*CRITERION FOR OPTIMUM (CHANGE IN VECTOR
     1 X .LT.EPS) HAS BEEN SATISFIED*)
34    FORMAT (/,1HO,*IEXIT=*,I2,*EITHER OF THE FOLLOWING THINGS HAS HAPP
     1ENED*,/,9X,*1. EPS CHOSEN IS TOO SMALL*,/,9X,*2. GRADIENTS ARE NOT
     2 CORRECT*,/,9X,*3. MATRIX H GOES SINGULAR*)
35    FORMAT (/,1HO,*IEXIT=*,I2,*MAXIMUM NUMBER OF ALLOWABLE ITERATION H
     1AS BEEN EXCEEDED*)
36    FORMAT (/,1HO,*IEXIT=*,I2,*FUNCTION VALUE LESS THAN MINIMUM ESTIMA
     1TED HAS BEEN DETECTED*)
      END
```

```
      SUBROUTINE THETA (X,NDIM,ETA,EST,MAX,MODE,X2,X1,G1,G2,ALFA,H,P,Y,P
     1Y,PE,BIGV,EPS1,NN,IPRINT,F2)
      DIMENSION X(1), X1(1), ETA(1), X2(1), G1(1), G2(1), ALFA(1), H(1),
     1 P(NN,1), Y(1), PY(1), PE(1), BIGV(1)
      COMMON /BLK/ KO
      COMMON /BLK1/ NUMF
      KO=0
      CALL SECOND (T3)
      IFLAG=0
      KK=0
      M=0
      N2=NDIM+1
      N1=NDIM+2
      NUMF=0
      IER=0
      DO 1 I=1,N1
      X1(I)=X(I)
    1 CONTINUE
      CALL FUNCT (NDIM,X1,F1,G1)
      NUMF=NUMF+1
      CALL SECOND (T4)
      TIME=T4-T3
      IF (IPRINT.EQ.0) GO TO 2
      CALL WRITE2 (X1,NDIM,G1,F1,NUMF,M,TIME)
    2 CONTINUE
      DO 3 I=1,NDIM
      X2(I)=X1(I)
      G2(I)=G1(I)
      H(I)=-G1(I)
    3 CONTINUE
      F2=F1
      X2(N2)=X1(N2)
      X2(N1)=X1(N1)
    4 CONTINUE
      KOUNT=0
      EPS=EPS1
      CALL MIN1D (FUNCT,X2,H,RO,NDIM,F2,G2,NUMF,IER,EPS,EST)
      IF (IER.NE.0) GO TO 30
      DO 5 I=1,N1
      BIGV(I)=X2(I)
      ALFA(I)=X2(I)
    5 CONTINUE
      RO=-RO
      GG=0.
      DO 6 I=1,NDIM
      GG=GG+G2(I)*G2(I)
    6 CONTINUE
      GG=SQRT(GG)
      IF (IPRINT.EQ.0) GO TO 7
      IF (KK.NE.IPRINT) GO TO 7
      CALL SECOND (T4)
      TIME=T4-T3
      KK=0
      CALL WRITE2 (X2,NDIM,G2,F2,NUMF,M,TIME)
    7 CONTINUE
      DO 9 I=1,N1
      DO 8 J=1,N1
      P(I,J)=0.
    8 CONTINUE
      P(I,I)=1.
    9 CONTINUE
   10 CONTINUE
      KOUNT=0
      KOUNT=KOUNT+1
   11 DO 12 I=1,NDIM
      Y(I)=G2(I)
   12 CONTINUE
      Y(N2)=F2
      Y(N1)=ETA(1)
      V=0.
      DO 13 I=1,NDIM
      V=V+X2(I)*G2(I)
   13 CONTINUE
      YA=0.
      DO 14 I=1,N1
      YA=YA+Y(I)*ALFA(I)
```

```
14    CONTINUE
      VYA=V-YA
      BIGV(KOUNT)=V
      DO 15 I=1,N1
      PY(I)=0.
      PE(I)=P(I,KOUNT)
      DO 15 J=1,N1
15    PY(I)=PY(I)+P(J,I)*Y(J)
      EPY=PY(KOUNT)
      IF (ABS(EPY).LT.ETA(3)) GO TO 31
      PY(KOUNT)=PY(KOUNT)-1.
      DO 16 I=1,N1
      DO 16 J=1,N1
16    P(I,J)=P(I,J)-PE(I)*PY(J)/EPY
      DO 17 I=1,N1
      ALFA(I)=0.
      DO 17 J=1,N1
17    ALFA(I)=ALFA(I)+P(I,J)*BIGV(J)
      DEL=0.
      DO 18 I=1,NDIM
      DEL=DEL+G2(I)*(X2(I)-ALFA(I))
18    CONTINUE
      IF (ABS(DEL).GT.ETA(4)) GO TO 19
      IF (IFLAG.EQ.1) GO TO 29
      IFLAG=1
      GO TO 31
19    IFLAG=0
      DO 20 I=1,N1
      H(I)=X2(I)-ALFA(I)
      IF (DEL.GT.0) H(I)=-H(I)
20    CONTINUE
      DO 21 I=1,NDIM
      X1(I)=X2(I)
      G1(I)=G2(I)
21    CONTINUE
      F1=F2
      X1(N2)=X2(N2)
      X1(N1)=X2(N1)
      X2(N2)=ALFA(N2)
      X2(N1)=ALFA(N1)
      CALL MIN1D (FUNCT,X2,H,RO,NDIM,F2,G2,NUMF,IER,EPS,EST)
      IF (IER.NE.0) GO TO 30
      IF (DEL.GT.0) RO=-RO
      GG=0.
      DO 22 I=1,NDIM
      GG=GG+G2(I)*G2(I)
22    CONTINUE
      GG=SQRT(GG)
      KOUNT=KOUNT+1
      M=M+1
      KK=KK+1
      IF (KK.NE.IPRINT) GO TO 23
      KK=0
      CALL SECOND (T4)
      TIME=T4-T3
      CALL WRITE2 (X2,NDIM,G2,F2,NUMF,M,TIME)
23    CONTINUE
      IF (M.GT.MAX) GO TO 26
      IF (MODE.EQ.2) GO TO 24
      IF (((F1-F2).LE.ETA(2))) GO TO 27
      GO TO 25
24    IF ((GG.LT.ETA(1))) GO TO 28
25    CONTINUE
      IF (KOUNT.LE.N1) GO TO 11
      GO TO 10
26    WRITE (6,35)
      GO TO 30
27    KO=1
      IF (IPRINT.EQ.0) RETURN
      WRITE (6,36)
      GO TO 30
28    KO=1
      IF (IPRINT.EQ.0) RETURN
      WRITE (6,37)
      GO TO 30
```

```
29    WRITE (6,34)
      KO=1
30    RETURN
31    IF (IPRINT.EQ.0) GO TO 32
      PRINT 38
32    CONTINUE
      DO 33 I=1,NDIM
      X1(I)=X2(I)
      G1(I)=G2(I)
      H(I)=-G1(I)
33    CONTINUE
      F1=F2
      X1(N2)=X(N2)
      X1(N1)=X(N1)
      X2(N2)=X(N2)
      X2(N1)=X(N1)
      GO TO 4


34    FORMAT (1HO,*RESTART COULD NOT YIELD SIGNIFICANT IMPROVEMENT,OPTIM
     1UM HAS BEEN REACHED*)
35    FORMAT (1HO,*MAXIMUM NUMBER OF ALLOWABLE ITERATIONS HAS BEEN EXCEE
     1DED*)
36    FORMAT (1HO,*CRITERION FOR OPTIMUM (FUNCTION VALUE DOES NOT CHANGE
     1  SIGNIFICANTLY ,MODE=1) HAS BEEN SATISFIED*)
37    FORMAT (1HO,*CRITERION FOR OPTIMUM (GRADIENTS HAVE BECOME TOO SMAL
     1L MODE=2) HAS BEEN SATISFIED*)
38    FORMAT (///20X,*A RESTART HAS OCCURRED*///)
      END
```

```
      SUBROUTINE INPUT (MET1,MET2,MET3,M,MAX,MODE,N,IPRINT,IDATA,EPS1,ET
     1A,EST,EPS,XSTRT)
      DIMENSION XSTRT(1), ETA(1), EPS(1)
      WRITE (6,7)
      IF (MET1.EQ.0) MET1=4
      IF (MET2.EQ.0) MET2=4
      IF (MET3.EQ.0) MET3=4
      INDEX=0
      GO TO (3,4,5,6), MET1
1     GO TO (3,4,5,6), MET2
2     GO TO (3,4,5,6), MET3
3     WRITE (6,8)
      GO TO 6
4     WRITE (6,9)
      GO TO 6
5     WRITE (6,10)
6     CONTINUE
      INDEX=INDEX+1
      IF (INDEX.EQ.1) GO TO 1
      IF (INDEX.EQ.2) GO TO 2
      WRITE (6,11) N
      WRITE (6,12) MAX
      WRITE (6,13) IPRINT
      WRITE (6,14) M
      WRITE (6,15) XSTRT(1)
      WRITE (6,16) (I,XSTRT(I),I=2,N)
      WRITE (6,17) ETA(1),ETA(2),ETA(3),ETA(4)
      WRITE (6,18) EPS(1)
      WRITE (6,19) (I,EPS(I),I=2,N)
      WRITE (6,20) EPS1
      WRITE (6,21) EST
      RETURN
C
C
7     FORMAT (1H0,*INPUT DATA*,/,1X,10(*-*),//,1X,*FOLLOWING METHODS HAV
     1E BEEN CALLED*,/)
8     FORMAT (1H0,*NEW FLETCHER METHOD*)
9     FORMAT (1H0,*JACOBSON-OKSMAN METHOD*)
10    FORMAT (1H0,*FLETCHER-POWELL METHOD*)
11    FORMAT (1H0,/1X,*NUMBER OF INDEPENDENT VARIABLES*,36(*.*),*N=*,I5,
     1/)
12    FORMAT (1H0,*MAXIMUM NUMBER OF ALLOWABLE ITERATIONS*,27(*.*),*MAX=
     1*,I5,/)
13    FORMAT (1H0,*INTERMEDIATE OUTPUT TO BE PRINTED EVERY IPRINT ITERAT
     1IONS*,5(*.*),*IPRINT=*,I5,/)
14    FORMAT (1H0,*STARTING POINT TO BE SAME FOR ALL THE METHODS IF M=1*
     1,15(*.*),*M=*,I5,/)
15    FORMAT (1H0,*STARTING VALUE FOR VECTOR X(I)*,29(*.*),*XSTRT( 1)=*,
     1E16.8)
16    FORMAT (1H0,59X,*XSTRT(*,I2,*)=*,E16.8)
17    FORMAT (1H0,/,1H0,*TEST QUANTITIES TO BE USED IN JACOBSON-OKSMAN M
     1ETHOD*,9(*.*),*ETA( 1)=*,E16.8,/,62X,*ETA( 2)=*,E16.8,/,62X,*ETA(
     23)=*,E16.8,/62X,*ETA( 4)=*,E16.8)
18    FORMAT (1H0,/,1H0,*TEST QUANTITIES TO BE USED IN NEW FLETCHER METH
     1OD*,12(*.*),*EPS( 1)=*,E16.8)
19    FORMAT (1H0,61X,*EPS(*,I2,*)=*,E16.8)
20    FORMAT (1H0,/,1H0,*TEST QUANTITY TO BE USED IN FLETCHER-POWELL MET
     1HOD*,14(*.*),*EPS1=*,E16.8)
21    FORMAT (1H0,/,1H0,*ESTIMATE OF LOWER BOUND ON FUNCTION TO BE MINIM
     1IZED*,14(*.*),*EST=*,E16.8)
      END
```

```
      SUBROUTINE MIN1D (FUNCT,X,H,AMBDA,N,F,G,NUMF,IER,EPS,EST)
      DIMENSION H(1), X(1), G(1)
      IER=0
      DY=0.
      HNRM=0.
      GNRM=0.
      DO 1 J=1,N
      HNRM=HNRM+ABS(H(J))
      GNRM=GNRM+ABS(G(J))
      DY=DY+H(J)*G(J)
1     CONTINUE
      IF (DY) 2,31,31
2     IF (HNRM/GNRM-EPS) 31,31,3
3     FY=F
      ALFA=2.*(EST-F)/DY
      IF (X(N+1).GT.0.) ALFA=X(N+1)*ALFA/2.
      AMBDA=1.
      IF (ALFA) 6,6,4
4     IF (ALFA-AMBDA) 5,6,6
5     AMBDA=ALFA
6     ALFA=0.
7     FX=FY
      DX=DY
      DO 8 I=1,N
      X(I)=X(I)+AMBDA*H(I)
8     CONTINUE
      CALL FUNCT (N,X,F,G)
      NUMF=NUMF+1
      IF (F.LT.FX) RETURN
      FY=F
      DY=0.
      DO 9 I=1,N
      DY=DY+G(I)*H(I)
9     CONTINUE
      IF (DY) 10,30,13
10    IF (FY-FX) 11,13,13
11    AMBDA=AMBDA+ALFA
      ALFA=AMBDA
      IF (HNRM*AMBDA-1.E10) 7,7,12
12    IER=2
      GO TO 31
13    T=0.
14    IF (AMBDA) 15,30,15
15    Z=3.*(FX-FY)/AMBDA+DX+DY
      ALFA=AMAX1(ABS(Z),ABS(DX),ABS(DY))
      DALFA=Z/ALFA
      DALFA=DALFA*DALFA-DX/ALFA*DY/ALFA
      IF (DALFA) 31,16,16
16    W=ALFA*SQRT(DALFA)
      ALFA=DY-DX+W+W
      IF (ALFA) 17,18,17
17    ALFA=(DY-Z+W)/ALFA
      GO TO 19
18    ALFA=(Z+DY-W)/(Z+DX+Z+DY)
19    ALFA=ALFA*AMBDA
      DO 20 I=1,N
      X(I)=X(I)+(T-ALFA)*H(I)
20    CONTINUE
      CALL FUNCT (N,X,F,G)
      NUMF=NUMF+1
      IF (F.LT.FX) GO TO 30
      IF (F-FX) 21,21,22
21    IF (F-FY) 30,30,22
22    DALFA=0.
      DO 23 I=1,N
      DALFA=DALFA+G(I)*H(I)
23    CONTINUE
      IF (DALFA) 24,27,27
24    IF (F-FX) 26,25,27
25    IF (DX-DALFA) 26,30,26
26    FX=F
      DX=DALFA
      T=ALFA
      AMBDA=ALFA
      GO TO 14
27    IF (FY-F) 29,28,29
```

```
28    IF (DY-DALFA) 29,30,29
29    FY=F
      DY=DALFA
      AMBDA=AMBDA-ALFA
      GO TO 13
30    AMBDA=AMBDA-ALFA
      RETURN
31    CONTINUE
      IF (DY.GE.0.) IER=-2
      IF (HNRM/GNRM.LE.EPS) IER=-3
      IF (DALFA.LT.0.) IER=-1
      II=IABS(IER)
      GO TO (32,33,34), II
32    WRITE (6,36) IER
      GO TO 35
33    WRITE (6,37) IER
      GO TO 35
34    WRITE (6,38) IER
35    RETURN
C
C
C
36    FORMAT (1HO,*IER=*,I2,*THERE IS AN ERROR IN GRADIENTS CALCULATION*
     1)
37    FORMAT (1HO,*IER=*,I2,*ERROR HAS OCCURED, SEARCH DIRECTION IS NOT
     1A DESCENT DIRECTION*)
38    FORMAT (1HO,*IER=*,I2,*ERROR HAS OCCURED, SEARCH DIRECTION VECTOR
     1 IS TOO SMALL IN COMPARISON TO GRADIENT VECTOR*)
      END
```

```
      SUBROUTINE WRITE1 (N)
      WRITE (6,5)
      GO TO (1,2,3), N
1     WRITE (6,6)
      GO TO 4
2     WRITE (6,7)
      GO TO 4
3     WRITE (6,8)
4     CONTINUE
      WRITE (6,9)
      RETURN
C
C
C
5     FORMAT (1H1)
6     FORMAT (1HO,*OPTIMIZATION BY NEW FLETCHER METHOD*,/,1HO,*---------
     1--------------------------*)
7     FORMAT (1HO,*OPTIMIZATION BY JACOBSON-OKSMAN METHOD*,/,1HO,*------
     1--------------------------------*)
8     FORMAT (1HO,*OPTIMIZATION BY FLETCHER-POWELL METHOD*,/,1HO,*------
     1--------------------------------*)
9     FORMAT (1HO,*ITERATION*,2X,*FUNCTION*,6X,*TIME ELAPSED*,8X,*OBJECT
     1IVE *,13X,*VARIABLE VECTOR X(I)*,9X,*GRADIENT VECTOR G(I)*,/1HO,*N
     2UMBER*,5X,*EVALUATIONS*,3X,*(SECONDS)*,11X,*FUNCTION*,/)
      END
```

```
      SUBROUTINE WRITE2 (X,N,G,F,NUMF,ITER,TIME)
      DIMENSION X(1), G(1)
      WRITE (6,1) ITER,NUMF,TIME,F,((X(I),G(I)),I=1,N)
      RETURN
C
C
C
1     FORMAT (1HO,I5,7X,I5,5X,E16.8,3X,E16.8,12X,95(E16.8,13X,E16.8,/,70
     1X))
      END
```

```
      SUBROUTINE FMFP (FUNCT,N,X,F,G,EST,EPS,LIMIT,IER,H,IPRINT)
      COMMON /BLK/ KO
      COMMON /BLK1/ NUMF
C        SUBROUTINE FMFP
C
C        PURPOSE
C           TO FIND A LOCAL MINIMUM OF A FUNCTION OF SEVERAL VARIABLES
C           BY THE METHOD OF FLETCHER AND POWELL
C
C        USAGE
C           CALL FMFP(FUNCT,N,X,F,G,EST,EPS,LIMIT,IER,H)
C
C        DESCRIPTION OF PARAMETERS
C           FUNCT  - USER-WRITTEN SUBROUTINE CONCERNING THE FUNCTION TO
C                    BE MINIMIZED. IT MUST BE OF THE FORM
C                    SUBROUTINE FUNCT(N,ARG,VAL,GRAD)
C                    AND MUST SERVE THE FOLLOWING PURPOSE
C                    FOR EACH N-DIMENSIONAL ARGUMENT VECTOR  ARG,
C                    FUNCTION VALUE AND GRADIENT VECTOR MUST BE COMPUTED
C                    AND, ON RETURN, STORED IN VAL AND GRAD RESPECTIVELY
C           N      - NUMBER OF VARIABLES
C           X      - VECTOR OF DIMENSION N CONTAINING THE INITIAL
C                    ARGUMENT WHERE THE ITERATION STARTS. ON RETURN,
C                    X HOLDS THE ARGUMENT CORRESPONDING TO THE
C                    COMPUTED MINIMUM FUNCTION VALUE
C           F      - SINGLE VARIABLE CONTAINING THE MINIMUM FUNCTION
C                    VALUE ON RETURN, I.E. F=F(X).
C           G      - VECTOR OF DIMENSION N CONTAINING THE GRADIENT
C                    VECTOR CORRESPONDING TO THE MINIMUM ON RETURN,
C                    I.E. G=G(X).
C           EST    - IS AN ESTIMATE OF THE MINIMUM FUNCTION VALUE.
C           EPS    - TESTVALUE REPRESENTING THE EXPECTED ABSOLUTE ERROR.
C                    A REASONABLE CHOICE IS 10**(-6), I.E.
C                    SOMEWHAT GREATER THAN 10**(-D), WHERE D IS THE
C                    NUMBER OF SIGNIFICANT DIGITS IN FLOATING POINT
C                    REPRESENTATION.
C           LIMIT  - MAXIMUM NUMBER OF ITERATIONS.
C           IER    - ERROR PARAMETER
C                    IER = 0 MEANS CONVERGENCE WAS OBTAINED
C                    IER = 1 MEANS NO CONVERGENCE IN LIMIT ITERATIONS
C                    IER =-1 MEANS ERRORS IN GRADIENT CALCULATION
C                    IER = 2 MEANS LINEAR SEARCH TECHNIQUE INDICATES
C                    IT IS LIKELY THAT THERE EXISTS NO MINIMUM.
C           H      - WORKING STORAGE OF DIMENSION N*(N+7)/2.
C
C        REMARKS
C            I) THE SUBROUTINE NAME REPLACING THE DUMMY ARGUMENT  FUNCT
C               MUST BE DECLARED AS EXTERNAL IN THE CALLING PROGRAM.
C           II) IER IS SET TO 2 IF , STEPPING IN ONE OF THE COMPUTED
C               DIRECTIONS, THE FUNCTION WILL NEVER INCREASE WITHIN
C               A TOLERABLE RANGE OF ARGUMENT.
C               IER = 2 MAY OCCUR ALSO IF THE INTERVAL WHERE F
C               INCREASES IS SMALL AND THE INITIAL ARGUMENT WAS
C               RELATIVELY FAR AWAY FROM THE MINIMUM SUCH THAT THE
C               MINIMUM WAS OVERLEAPED. THIS IS DUE TO THE SEARCH
C               TECHNIQUE WHICH DOUBLES THE STEPSIZE UNTIL A POINT
C               IS FOUND WHERE THE FUNCTION INCREASES.
C
C        SUBROUTINES AND FUNCTION SUBPROGRAMS REQUIRED
C           FUNCT
C
C        METHOD
C           THE METHOD IS DESCRIBED IN THE FOLLOWING ARTICLE
C           R. FLETCHER AND M.J.D. POWELL, A RAPID DESCENT METHOD FOR
C           MINIMIZATION,
C           COMPUTER JOURNAL VOL.6, ISS. 2, 1963, PP.163-168.
C
C     ..................................................................
C
C
C        DIMENSIONED DUMMY VARIABLES
      DIMENSION H(1), X(1), G(1)
      KQ=0
      KAML=0
C
C        COMPUTE FUNCTION VALUE AND GRADIENT VECTOR FOR INITIAL ARGUMENT
```

```
      KO=0
      CALL SECOND (T3)
      CALL FUNCT (N,X,F,G)
      KOUNT=0
      NUMF=1
      CALL SECOND (T4)
      TIME=T4-T3
      IF (IPRINT.EQ.0) GO TO 1
      CALL WRITE2 (X,N,G,F,NUMF,KOUNT,TIME)
1     CONTINUE
C
C        RESET ITERATION COUNTER AND GENERATE IDENTITY MATRIX
      IER=0
      KK=0
      N2=N+N
      N3=N2+N
      N31=N3+1
2     K=N31
      DO 5 J=1,N
      H(K)=1.
      NJ=N-J
      IF (NJ) 6,6,3
3     DO 4 L=1,NJ
      KL=K+L
      H(KL)=0.
4     CONTINUE
      K=KL+1
5     CONTINUE
C
C        START ITERATION LOOP
6     IF (KOUNT.EQ.0) GO TO 7
      IF (KK.NE.IPRINT) GO TO 7
      KK=0
      CALL SECOND (T4)
      TIME=T4-T3
      CALL WRITE2 (X,N,G,F,NUMF,KOUNT,TIME)
7     CONTINUE
      IF (KAML.EQ.0) GO TO 10
      FTE=ABS(F-OLDF)
      FQE=EPS*ABS(F)
      IF (FTE.LT.FQE) GO TO 8
      KQ=0
      GO TO 9
8     KQ=KQ+1
9     CONTINUE
10    KAML=KAML+1
      IF (KQ.EQ.4) GO TO 70
      KOUNT=KOUNT+1
      KK=KK+1
C
C        SAVE FUNCTION VALUE, ARGUMENT VECTOR AND GRADIENT VECTOR
      OLDF=F
      DO 14 J=1,N
      K=N+J
      H(K)=G(J)
      K=K+N
      H(K)=X(J)
C
C        DETERMINE DIRECTION VECTOR H
      K=J+N3
      T=0.
      DO 13 L=1,N
      T=T-G(L)*H(K)
      IF (L-J) 11,12,12
11    K=K+N-L
      GO TO 13
12    K=K+1
13    CONTINUE
      H(J)=T
14    CONTINUE
C
C        CHECK WHETHER FUNCTION WILL DECREASE STEPPING ALONG H.
      DY=0.
      HNRM=0.
      GNRM=0.
```

```
C
C        CALCULATE DIRECTIONAL DERIVATIVE AND TESTVALUES FOR DIRECTION
C        VECTOR H AND GRADIENT VECTOR G.
      DO 15 J=1,N
      HNRM=HNRM+ABS(H(J))
      GNRM=GNRM+ABS(G(J))
      DY=DY+H(J)*G(J)
   15 CONTINUE
C
C        REPEAT SEARCH IN DIRECTION OF STEEPEST DESCENT IF DIRECTIONAL
C        DERIVATIVE APPEARS TO BE POSITIVE OR ZERO.
      IF (DY) 16,60,60
C
C        REPEAT SEARCH IN DIRECTION OF STEEPEST DESCENT IF DIRECTION
C        VECTOR H IS SMALL COMPARED TO GRADIENT VECTOR G.
   16 IF (HNRM/GNRM-EPS) 60,60,17
C
C        SEARCH MINIMUM ALONG DIRECTION H
C
C        SEARCH ALONG H FOR POSITIVE DIRECTIONAL DERIVATIVE
   17 FY=F
      ALFA=2.*(EST-F)/DY
      AMBDA=1.
C
C        USE ESTIMATE FOR STEPSIZE ONLY IF IT IS POSITIVE AND LESS THAN
C        1. OTHERWISE TAKE 1. AS STEPSIZE
      IF (ALFA) 20,20,18
   18 IF (ALFA-AMBDA) 19,20,20
   19 AMBDA=ALFA
   20 ALFA=0.
C
C        SAVE FUNCTION AND DERIVATIVE VALUES FOR OLD ARGUMENT
   21 FX=FY
      DX=DY
C
C        STEP ARGUMENT ALONG H
      DO 22 I=1,N
      X(I)=X(I)+AMBDA*H(I)
   22 CONTINUE
C
C        COMPUTE FUNCTION VALUE AND GRADIENT FOR NEW ARGUMENT
      CALL FUNCT (N,X,F,G)
      NUMF=NUMF+1
      FY=F
C
C        COMPUTE DIRECTIONAL DERIVATIVE DY FOR NEW ARGUMENT.  TERMINATE
C        SEARCH, IF DY IS POSITIVE. IF DY IS ZERO THE MINIMUM IS FOUND
      DY=0.
      DO 23 I=1,N
      DY=DY+G(I)*H(I)
   23 CONTINUE
      IF (DY) 24,44,27
C
C        TERMINATE SEARCH ALSO IF THE FUNCTION VALUE INDICATES THAT
C        A MINIMUM HAS BEEN PASSED
   24 IF (FY-FX) 25,27,27
C
C        REPEAT SEARCH AND DOUBLE STEPSIZE FOR FURTHER SEARCHES
   25 AMBDA=AMBDA+ALFA
      ALFA=AMBDA
C        END OF SEARCH LOOP
C
C        TERMINATE IF THE CHANGE IN ARGUMENT GETS VERY LARGE
      IF (HNRM*AMBDA-1.E10) 21,21,26
C
C        LINEAR SEARCH TECHNIQUE INDICATES THAT NO MINIMUM EXISTS
   26 IER=2
      GO TO 65
C
C        INTERPOLATE CUBICALLY IN THE INTERVAL DEFINED BY THE SEARCH
C        ABOVE AND COMPUTE THE ARGUMENT X FOR WHICH THE INTERPOLATION
C        POLYNOMIAL IS MINIMIZED
   27 T=0.
   28 IF (AMBDA) 29,44,29
```

```
29    Z=3.*(FX-FY)/AMBDA+DX+DY
      ALFA=AMAX1(ABS(Z),ABS(DX),ABS(DY))
      DALFA=Z/ALFA
      DALFA=DALFA*DALFA-DX/ALFA*DY/ALFA
      IF (DALFA) 60,30,30
30    W=ALFA*SQRT(DALFA)
      ALFA=DY-DX+W+W
      IF (ALFA) 31,32,31
31    ALFA=(DY-Z+W)/ALFA
      GO TO 33
32    ALFA=(Z+DY-W)/(Z+DX+Z+DY)
33    ALFA=ALFA*AMBDA
      DO 34 I=1,N
      X(I)=X(I)+(T-ALFA)*H(I)
34    CONTINUE
C
C        TERMINATE, IF THE VALUE OF THE ACTUAL FUNCTION AT X IS LESS
C        THAN THE FUNCTION VALUES AT THE INTERVAL ENDS. OTHERWISE REDUCE
C        THE INTERVAL BY CHOOSING ONE END-POINT EQUAL TO X AND REPEAT
C        THE INTERPOLATION.  WHICH END-POINT IS CHOOSEN DEPENDS ON THE
C        VALUE OF THE FUNCTION AND ITS GRADIENT AT X
C
      NUMF=NUMF+1
      CALL FUNCT (N,X,F,G)
      IF (F-FX) 35,35,36
35    IF (F-FY) 44,44,36
36    DALFA=0.
      DO 37 I=1,N
      DALFA=DALFA+G(I)*H(I)
37    CONTINUE
      IF (DALFA) 38,41,41
38    IF (F-FX) 40,39,41
39    IF (DX-DALFA) 40,44,40
40    FX=F
      DX=DALFA
      T=ALFA
      AMBDA=ALFA
      GO TO 28
41    IF (FY-F) 43,42,43
42    IF (DY-DALFA) 43,44,43
43    FY=F
      DY=DALFA
      AMBDA=AMBDA-ALFA
      GO TO 27
C
C        TERMINATE, IF FUNCTION HAS NOT DECREASED DURING LAST ITERATION
44    IF (OLDF-F+EPS) 60,45,45
C
C        COMPUTE DIFFERENCE VECTORS OF ARGUMENT AND GRADIENT FROM
C        TWO CONSECUTIVE ITERATIONS
45    DO 46 J=1,N
      K=N+J
      H(K)=G(J)-H(K)
      K=N+K
      H(K)=X(J)-H(K)
46    CONTINUE
C
C        TEST LENGTH OF ARGUMENT DIFFERENCE VECTOR AND DIRECTION VECTOR
C        IF AT LEAST N ITERATIONS HAVE BEEN EXECUTED. TERMINATE, IF
C        BOTH ARE LESS THAN  EPS
      IER=0
      IF (KOUNT-N) 50,47,47
47    T=0.
      Z=0.
      DO 48 J=1,N
      K=N+J
      W=H(K)
      K=K+N
      T=T+ABS(H(K))
      Z=Z+W*H(K)
48    CONTINUE
      IF (HNRM-EPS) 49,49,50
49    IF (T-EPS) 65,65,50
C
```

```
C          TERMINATE, IF NUMBER OF ITERATIONS WOULD EXCEED  LIMIT
 50     IF (KOUNT-LIMIT) 51,58,58
C
C          PREPARE UPDATING OF MATRIX H
 51     ALFA=0.
        DO 55 J=1,N
        K=J+N3
        W=0.
        DO 54 L=1,N
        KL=N+L
        W=W+H(KL)*H(K)
        IF (L-J) 52,53,53
 52     K=K+N-L
        GO TO 54
 53     K=K+1
 54     CONTINUE
        K=N+J
        ALFA=ALFA+W*H(K)
        H(J)=W
 55     CONTINUE
C
C          REPEAT SEARCH IN DIRECTION OF STEEPEST DESCENT IF RESULTS
C          ARE NOT SATISFACTORY
        IF (Z*ALFA) 56,2,56
C
C          UPDATE MATRIX H
 56     K=N31
        DO 57 L=1,N
        KL=N2+L
        DO 57 J=L,N
        NJ=N2+J
        H(K)=H(K)+H(KL)*H(NJ)/Z-H(L)*H(J)/ALFA
 57     K=K+1
        GO TO 6
C          END OF ITERATION LOOP
C
C          NO CONVERGENCE AFTER  LIMIT  ITERATIONS
 58     IER=1
        IF (KK.NE.IPRINT) GO TO 59
        CALL WRITE2 (X,N,G,F,NUMF,KOUNT)
 59     CONTINUE
        GO TO 65
C
C          RESTORE OLD VALUES OF FUNCTION AND ARGUMENTS
 60     DO 61 J=1,N
        K=N2+J
        X(J)=H(K)
 61     CONTINUE
        CALL FUNCT (N,X,F,G)
        NUMF=NUMF+1
C
C          REPEAT SEARCH IN DIRECTION OF STEEPEST DESCENT IF DERIVATIVE
C          FAILS TO BE SUFFICIENTLY SMALL
        IF (GNRM-EPS) 64,64,62
C
C          TEST FOR REPEATED FAILURE OF ITERATION
 62     IF (IER) 65,63,63
 63     IER=-1
        GO TO 2
 64     IER=0
 65     II=IER+2
        IF (II.EQ.2) KO=1
        IF (IPRINT.EQ.0) RETURN
        GO TO (66,67,68,69), II
 66     WRITE (6,73) IER
        GO TO 71
 67     WRITE (6,74) IER
        GO TO 71
 68     WRITE (6,75) IER
        GO TO 71
 69     WRITE (6,76) IER
 70     WRITE (6,72)
        KO=1
 71     RETURN
```

```
C
C
C
72    FORMAT (1HO,*THERE IS NO SIGNIFICANT DECREASE IN THE FUNCTION VALU
     1E OPTIMUM IS ASSUMED TO HAVE BEEN REACHED*)
73    FORMAT (1HO,*IER=*,I2,* ERROR IN GRADIENTS CALCULATIONS*)
74    FORMAT (1HO,*IER=*,I2,* CRITERION FOR OPTIMUM HAS BEEN SATISFIED*)
75    FORMAT (1HO,*IER=*,I2,* MAXIMUM NUMBER OF ALLOWABLE ITERATIONS HAS
     1 BEEN EXCEEDED*)
76    FORMAT (1HO,*IER=*,I2,* CHANGE IN ARGUMENTS GETS TOO LARGE, LINEAR
     1 SEARCH INDICATES THAT NO MINIMUM EXISTS*)
      END
```

```
      SUBROUTINE GRADMIN (N,X,XSTRT,X2,X1,G,G2,ALFA,H,P,PY,PE,BIGV,EPS,N
     1N,Y,DUM1,DUM2,KR)
      COMMON /BLK/ KO
C
      EXTERNAL FUNCT
      LOGICAL CONV,UNITH
      DIMENSION DUM1(1), DUM2(1)
      DIMENSION X(1), XSTRT(1), X2(1), X1(1), G(1), G2(1), ALFA(1), H(1)
     1, P(NN,1), Y(1), PY(1), PE(1), BIGV(1), EPS(1), ETA(4)
      UNITH=.TRUE.
      IF (KR.EQ.0) GO TO 1
      CALL DATA (N,NN,XSTRT,EST,EPS1,ETA,MET1,MET2,MET3,M,MAX,MODE,IPRIN
     1T,EPS)
 1    DO 2 I=1,NN
      X(I)=XSTRT(I)
 2    CONTINUE
      IF (MET1.EQ.0) MET1=4
      IF (MET2.EQ.0) MET2=4
      IF (MET3.EQ.0) MET3=4
      INDEX=0
      GO TO (5,10,15,20), MET1
 3    GO TO (5,10,15,20), MET2
 4    GO TO (5,10,15,20), MET3
 5    IF (IPRINT.EQ.0) GO TO 6
      CALL WRITE1 (1)
 6    CONTINUE
      CALL SECOND (T1)
      IF (KR.NE.0) GO TO 8
      DO 7 I=1,NN
      X(I)=DUM1(I)
 7    CONTINUE
 8    CONTINUE
      CALL VMMO1 (N,X,F,G,H,UNITH,EST,EPS,MAX,IPRINT,IEXIT,PE)
      DO 9 I=1,NN
      DUM1(I)=X(I)
 9    CONTINUE
      CALL SECOND (T2)
      CALL FINAL (X,F,N)
      T=T2-T1
      WRITE (6,25) T
      GO TO 20
 10   IF (IPRINT.EQ.0) GO TO 11
      CALL WRITE1 (2)
 11   CONTINUE
      CALL SECOND (T1)
      IF (KR.NE.0) GO TO 13
      DO 12 I=1,NN
      X(I)=X2(I)
 12   CONTINUE
 13   CONTINUE
      CALL THETA (X,N,ETA,EST,MAX,MODE,X2,X1,G,G2,ALFA,H,P,Y,PY,PE,BIGV,
     1EPS1,NN,IPRINT,F2)
      DO 14 I=1,NN
      X(I)=X2(I)
 14   CONTINUE
      CALL SECOND (T2)
      CALL FINAL (X2,F2,N)
      T=T2-T1
      WRITE (6,25) T
      GO TO 20
 15   IF (IPRINT.EQ.0) GO TO 16
      CALL WRITE1 (3)
 16   CONTINUE
      CALL SECOND (T1)
      IF (KR.NE.0) GO TO 18
      DO 17 I=1,NN
      X(I)=DUM2(I)
 17   CONTINUE
 18   CONTINUE
      CALL FMFP (FUNCT,N,X,F,G,EST,EPS1,MAX,IER,H,IPRINT)
      DO 19 I=1,NN
      DUM2(I)=X(I)
 19   CONTINUE
      CALL SECOND (T2)
      CALL FINAL (X,F,N)
      T=T2-T1
```

```
      WRITE (6,25) T
20    INDEX=INDEX+1
      IF (M.EQ.1) GO TO 21
      GO TO 23
21    DO 22 I=1,NN
      X(I)=XSTRT(I)
22    CONTINUE
23    CONTINUE
      GO TO (3,4,24), INDEX
24    CONTINUE
      RETURN
C
C
C
25    FORMAT (1HO,//25X,*EXECUTION TIME IN SECONDS=*,F10.5)
      END
```