INTERNAL REPORTS IN

SIMULATION, OPTIMIZATION AND CONTROL

No. SOC-17

FUNCTION OPTIMIZATION PACKAGE
VERSION FLOPT1

J.W. Bandler and W.Y. Chu

August 1973

FACULTY OF ENGINEERING
McMASTER UNIVERSITY
HAMILTON, ONTARIO, CANADA

FUNCTION OPTIMIZATION PACKAGE:

VERSION FLOPT1

J.W. Bandler and W.Y. Chu

Abstract The program developed solves unconstrained optimization problems by the Fletcher method. It is adaptable to solving constrained problems by using sequential unconstrained minimization techniques. The program is available for batch processing as well as on the time-sharing system INTERCOM of McMaster's CDC 6400.

I. INTRODUCTION

The package FLOPT1 is a modified and more concise version of GRADMIN [1] to be used for optimizing any well-behaved once continuously differentiable unconstrained objective function of the form $U = U(x_1, x_2, \ldots, x_n)$, where $x_1, x_2, \ldots, x_n$ are independent variables. By applying the Fiacco-McCormick method [2], objective functions subject to equality and/or inequality constraints can readily be optimized.

Three options are available for the package FLOPT1. The user can access the package for batch processing or he can use the package on the time-sharing system INTERCOM. Except for using the demonstration program, the user has to supply the main program and a subroutine called FUNCT which defines the objective function, the constraint functions (if applicable) and all first-order partial derivatives.

This work was supported by the National Research Council of Canada under grant A7239.

The authors are with the Group on Simulation, Optimization and Control, and Department of Electrical Engineering, McMaster University, Hamilton, Ontario, Canada.

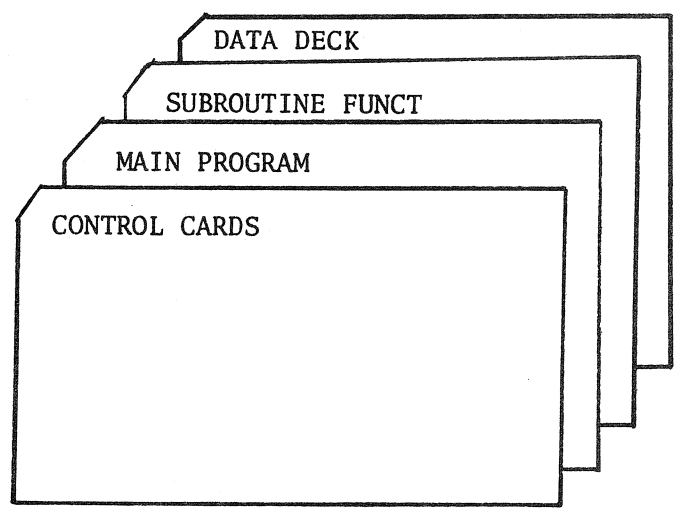
II. USAGE

The three options of the package FLOPT1 are stored on permanent files; each has its own filename. They are called:

1. FLOPT1B [for batch processing]
2. FLOPT1I [for INTERCOM]
3. FLOPT1D [for the demonstration program on INTERCOM]

A. Batch processing

1. Set up the input deck as follows:



2. CONTROL CARDS

A typical set of control cards is:

ABCD. USER NAME

ATTACH, ℓfn, FLOPT1B, ID = *********, MR = 1.†

FTN.

LOAD (ℓfn)

LGO.

END OF RECORD

PROGRAM TST(INPUT, OUTPUT, TAPE5=INPUT, TAPE6=OUTPUT)

3. MAIN PROGRAM

Write the main program as indicated below:

(a) Dimension the following arrays

X(N), G(N), EPS(N), H(K)

where N = The number of independent variables

K = N*(N+7)/2

(b) Supply the values of the following parameters.

N = The number of independent variables

K = N*(N+7)/2

M = 1 if data deck is to be read

= 0 if data deck is not to be read.

Always set M equal to 1 for the first call of FLOPT1.

(c) Call subroutine FLOPT1 as follows:

CALL FLOPT1(N, K, M, X, G, H, EPS)

(d) Add STOP and END cards.

---

† ℓfn stands for logical filename of no more than 7 characters. Appropriate identification parameter ID should be inserted in *********.

4. SUBROUTINE FUNCT

This subroutine defines the objective function, the constraints (where applicable) and all first-order partial derivatives. It is to be written as follows:

```
SUBROUTINE FUNCT(X, G, U)
DIMENSION X(N), G(N)
```

where N = The number of independent variables (to be entered explicitly in the dimension statement)

Define the objective function

$$U = U(x_1, x_2, \ldots, x_n)$$

Define the gradients of the objective function

$$G(1) = \frac{\partial U}{\partial x_1}$$

$$G(2) = \frac{\partial U}{\partial x_2}$$

$$\vdots$$

$$G(N) = \frac{\partial U}{\partial x_n}$$

Add RETURN and END cards.

If other statements are required for defining the function or the gradients, they may be added to this subroutine, e.g., function U may be defined in another subprogram which may then be called by subroutine FUNCT. Additional statements are also required if the problem is subject to constraints (See Appendix A).

5. DATA DECK

Parameters to be supplied as input data are defined below.

MAX	Maximum number of iterations allowed
IPT	Intermediate output is printed out every IPT iterations; it should be 0 if no intermediate output is desired
ID	Input data is printed out when ID=1; it should be 0 if input data is not to be printed
EST	Minimum estimated value of the objective function
X(I), I=1,N	Starting values for the variables $x_1, x_2, \ldots, x_n$
EPS(I), I=1,N	Test quantities used by Fletcher method

Recommended values for some of the parameters are

MAX = 100

EST = 0.0

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

6. ARRANGEMENT OF THE DATA DECK

CARD NO.	FORMAT	PARAMETERS
1	3I5	MAX, IPT, ID
2	E16.8	EST
As many as required	5E16.8	X(I), I=1,N
As many as required	5E16.8	EPS(I), I=1,N

B. Time-sharing system INTERCOM

On the time-sharing system INTERCOM, the user may either access the standard package or the demonstration program. To use the standard package:

1. The user should access the file by typing the command

   ATTACH, ℓfn, FLOPT1I, ID = *********[†].

2. The user then gets into the EDITOR mode to create the main program and subroutine FUNCT required by the package. This can be done by typing the commands

   EDITOR.

   CREATE

The main program and subroutine FUNCT should be written in the same manner as described in the previous section. However, there are some minor adjustments. In the main program, the dimension statement should be

DIMENSION X(N), G(N), EPS(N), H(K), XS(N)

and the CALL statement should be

CALL FLOPT1 (N, K, M, X, G, H, EPS, XS, JOPT)

The parameter JOPT, which specifies the mode of optimization, has to be defined. For unconstrained optimization, JOPT should be set to 0; for constrained optimization, JOPT has to be set to 1.

After typing the main program and subroutine FUNCT, the user may leave the EDITOR mode by typing the commands

SAVE, filename

BYE

3. The program just written has to be compiled. The user should type the command

   FTN, I = filename.

4. To execute the program with the package FLOPT1, the necessary commands are

   CONNECT, INPUT, OUTPUT.

   XEQ.

   LOAD = LGO, ℓfn.†

   EXECUTE

5. The input data required is the same as in batch processing. However, in the interactive mode, the user will be instructed to enter the required data. Input format is arbitrary. The user may separate each value by a comma, a blank, a few blanks or by typing the RETURN key. If a line cannot accommodate a specific input string, the user may go to the next line by typing the LINE FEED key. Unless specified, questions are expected to be answered by YES or NO. The user is also allowed to modify his data after he has entered all the data.

At the end of the run, the computer will await an instruction to stop execution or to rerun the program with different input data.

To use the demonstration program, the user simply types the following commands:

ATTACH, ℓfn, FLOPT1D, ID = *********.

CONNECT, INPUT, OUTPUT.

XEQ, ℓfn.

---

† ℓfn stands for logical filename of no more than 7 characters. It should be the same in the ATTACH command and in the LOAD command.

The main program and subroutine FUNCT for a specific problem will be printed out. The user is then asked to enter the data and the program is executed.

III. COMMENTS

The user will find much flexibility when using the package on INTERCOM. However, for a long subroutine FUNCT, it is advisable to store the created program on permanent file. The user can subsequently call back his program for execution.

In all the three options, by choosing appropriate values of IPT and ID, the user may or may not print out the input data and the intermediate output. Problems with constraints may be solved using the Fiacco-McCormick techniques as illustrated in Appendix C. In Appendix A, instructions for using the package to solve problems with constraints are given. Results for some of the problems solved using this pacakge have been included in Appendix B and C. Appendix D shows the general structure of the package.

REFERENCES

[1] J.W. Bandler and V.K. Jha, "GRADMIN - a package for function minimization using efficient gradient methods", Department of Electrical Engineering, McMaster University, Hamilton, Ontario, Canada, 1972, internal report.

[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, "A new approach to variable metric algorithms", Computer J., vol. 13, pp. 317-322, Aug. 1970.

APPENDIX A

The program package is primarily written to solve unconstrained optimization problems. By using the Fiacco-McCormick [2] techniques, a constrained problem can readily be transformed into an unconstrained one. The technique is basically to add the constraints as penalties to the unconstrained objective function. For example, to minimize with respect to $\underset{\sim}{\phi}$

$$U = U(\underset{\sim}{\phi}) \tag{A.1}$$

subject to inequality constraints

$$\underset{\sim}{g}(\underset{\sim}{\phi}) \geq \underset{\sim}{0} \tag{A.2}$$

and equality constraints

$$\underset{\sim}{h}(\underset{\sim}{\phi}) = \underset{\sim}{0} \tag{A.3}$$

A new objective function is defined

$$P(\underset{\sim}{\phi},r) = U(\underset{\sim}{\phi}) + r[\sum_{i=1}^{m} \frac{1}{g_i(\underset{\sim}{\phi})}] + \frac{1}{\sqrt{r}}[\sum_{j=1}^{s} h_j^2(\underset{\sim}{\phi})] \tag{A.4}$$

to be minimized instead of $U(\underset{\sim}{\phi})$. By reducing the parameter r, the effect of the penalty is reduced and under suitable conditions, a constrained minimum will be obtained.

A typical way of implementing the Fiacco-McCormick method is as follows

1. In subroutine FUNCT, define the penalty function as described above.

To avoid possible confusion, define the unconstrained objective function as F and the new objective function as U (function P in Eq. A.4). The constraint functions, their gradients and the gradients of the new objective function have to be defined. Appropriate adjustment in the dimension statement is expected.

2. In the main program, define the parameter r (the value of r should be available to the subroutine FUNCT and this can easily be done by using a COMMON statement) and put the subroutine FLOPT1 in a DO loop. In the batch processing option, the user has to pre-determine the number of calls of subroutine FLOPT1. In the interactive processing option, there is no such restriction.

Samples of the main program required when solving constrained problems are shown below:

For batch processing option FLOPT1B

```
      DIMENSION X(N), G(N), EPS(N), H(K)
      COMMON/SS/R
      M = 1
      R = 1.0
      DO 1 I = 1, 5
      CALL FLOPT1 (N, K, M, X, G, H, EPS)
      M = 0
      R = 0.01*R
1     CONTINUE
      STOP
      END
```

For interactive processing option FLOPT1I

```
      DIMENSION X(N), G(N), EPS(N), H(K), XS(N)
      COMMON/SS/R
      M = 1
      R = 1.0
      JOPT = 1
1     CALL FLOPT1 (N, K, M, X, G, H, EPS, XS, JOPT)
      M = 0
      R = 0.01*R
      GO TO 1
      END
```

APPENDIX B

To minimize

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

with the starting point:

$$x_1 = 0.0$$

$$x_2 = 0.0$$

A suitable listing of the input deck, a printout of the input data and some final results (using the batch-processing option) are given in Figs. 1-3.

CHU W

```
HSRO.
ATTACH,TAPE,FLOPT1B,ID=HSROBNLER,MR=1.
FTN.
LOAD(TAPE)
LGO.
'        6400 END OF RECORD
      PROGRAM TST (INPUT,OUTPUT,TAPE5=INPUT,TAPE6=OUTPUT)
      DIMENSION X(2),G(2),H(9),EPS(2)
      CALL FLOPT1(2,9,1,X,G,H,EPS)
      STOP
      END
C
      SUBROUTINE FUNCT(X,G,U)
      DIMENSION X(2),G(2)
      A=X(1)*X(1)
      B=A-X(2)
      C=1.0-X(1)
C     THE OBJECTIVE FUNCTION
      U=100.*B*B+C*C
C     GRADIENTS OF THE OBJECTIVE FUNCTION WITH RESPECT TO THE VARIABLE
C     PARAMETERS
      G(1)=400.0*(X(1)*A-X(1)*X(2))-2.0*C
      G(2)=-200.0*B
      RETURN
      END
'        6400 END OF RECORD
  100     1     1
0.0
0.0                 0.0
1.E-                61.E-               6
'              END OF FILE
```

Fig. 1 Main program and subroutine FUNCT for the example of Appendix B. Input data is also shown.

```
INPUT DATA
----------

NUMBER OF INDEPENDENT VARIABLES.........................N =    2

MAXIMUM NUMBER OF ALLOWABLE ITERATIONS...............MAX = 100

INTERMEDIATE PRINTOUT AT EVERY IPT ITERATIONS........IPT =    1

STARTING VALUE FOR VECTOR X(I)...................X( 1) =   0.
                                                 X( 2) =   0.

TEST QUANTITIES TO BE USED.....................EPS( 1) =    .100000E-05
                                               EPS( 2) =    .100000E-05

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

Fig. 2 Input data for the example of Appendix B.

```
IEXIT = 1
CRITERION FOR OPTIMUM (CHANGE IN VECTOR X .LT. EPS) HAS BEEN SATISFIED

OPTIMAL SOLUTION FOUND BY FLETCHER METHOD
-----------------------------------------

    U =    .534187E-13

X( 1) =    .100000E+01          G( 1) =    .856057E-05
X( 2) =    .100000E+01          G( 2) =  -.435743E-05

    NUMBER OF FUNCTION EVALUATIONS =    30
         EXECUTION TIME IN SECONDS =    .243
```

Fig. 3 Results for the example of Appendix B.

APPENDIX C

To minimize

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

subject to the following equality constraint

$$h = x_1 + 2x_2 - 1$$

The unconstrained function for this problem using the Fiacco-McCormick technique is

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

The starting point used is

$$x_1 = 0.0$$

$$x_2 = 0.0$$

Commands to enter the EDITOR mode, a listing of the main program and subroutine FUNCT, commands to compile the program, access the option FLOPT1I and execute the program, instructions for entering data, a record of the input data and some final results are given in Figs. 4-8.

```
MCMASTER UNIV INTERCOM 4.1
DATE  08/15/73
TIME  21.36.54.

PLEASE LOGIN
LOGIN.
ENTER USER NAME- HSBJ
[redacted] ENTER PASSWORD-




 08/15/73    LOGGED IN AT  21.38.09.
             WITH USER-ID BR
             EQUIP/PORT 70/12
COMMAND- EDITOR.
..CREATE
```

Fig. 4 Commands for entering the EDITOR mode.

```
      PROGRAM TST(INPUT,OUTPUT,TAPE5=INPUT,TAPE6=OUTPUT)
      COMMON/SS/R
      DIMENSION X(2),G(2),H(9),EPS(2),XS(2)
      R=1.0
      M=1
1     CALL FLOPT1(2,9,M,X,G,H,EPS,XS,1)
      M=0
      R=R*0.01
      GO TO 1
      END
      SUBROUTINE FUNCT(X,G,U)
      COMMON/SS/R
      DIMENSION X(2),G(2),GF(2),GC(2)
      A=2.*X(2)
C   THE OBJECTIVE FUNCTION
      F=X(1)*X(1)+A*A
C   GRADIENTS OF THE OBJECTIVE FUNCTION
      GF(1)=2.*X(1)
      GF(2)=4.*A
C   THE EQUALITY CONSTRAINT
      C=X(1)+A-1.
C   GRADIENTS OF THE CONSTRAINT
      GC(1)=1.
      GC(2)=2.
C   THE UNCONSTRAINED OBJECTIVE FUNCTION USING FIACCO-MCCORMICK
C   TECHNIQUE
      U=F+(C*C)/SQRT(R)
      AB=2.*C/SQRT(R)
C   GRADIENTS OF THE OBJECTIVE FUNCTION U
      G(1)=GF(1)+AB*GC(1)
      G(2)=GF(2)+AB*GC(2)
      RETURN
      END
..SAVE,A
..BYE
COMMAND- FTN,I=A.
       .323 CP SECONDS COMPILATION TIME
COMMAND- ATTACH,XX,FLOPT1I,ID=HSBOBNLER.
 PF CYCLE NO. = 006
COMMAND- CONNECT,INPUT,OUTPUT.
COMMAND- XEQ.
OPTION=LOAD=LGO,XX.
OPTION=EXECUTE
```

Fig. 5 A listing of the main program and subroutine FUNCT of the example of Appendix C. Commands for compiling and executing the program with option FLOPT1I are also shown.

```
YOU ARE WELCOME TO USE THE PACKAGE " F L O P T 1 ".
PLEASE SUPPLY DATA WHEN ASKED FOR.  YOU CAN ENTER YOUR DATA IN ANY
FORMAT, HOWEVER, BE REASONABLE.  PLEASE SEPARATE EACH VALUE BY A
COMMA, A BLANK OR TYPING THE RETURN KEY.  THANK YOU.

SPECIFY THE MAXIMUM NUMBER OF ITERATIONS ALLOWED.
1←      100
ENTER AN INTEGER SO THAT INTERMEDIATE OUTPUT WILL BE PRINTED AFTER
EVERY SPECIFIED NUMBER OF ITERATIONS.  ENTER 0 IF YOU DON"T WANT
ANY INTERMEDIATE OUTPUT.
2←      0
ENTER 1 IF YOU WANT TO HAVE A RECORD OF YOUR INPUT DATA, OTHERWISE
ENTER 0.
3←      1
SPECIFY A MINIMUM ESTIMATED VALUE OF THE OBJECTIVE FUNCTION.
4←      0.0
ENTER STARTING VALUES FOR THE VARIABLE PARAMETERS.
5←      0.0  0.0
ENTER SMALL VALUES FOR TESTING CONVERGENCE.
6←      1.E-6,1.E-6

ANY MODIFICATION
#       YES
WHICH ENTRY
#       1
SPECIFY THE MAXIMUM NUMBER OF ITERATIONS ALLOWED.
1←      50

ANY MODIFICATION
#       NO


YOUR DATA IS NOW BEING PROCESSED.  IT MAY TAKE SOME TIME BEFORE
RESULTS ARE AVAILABLE.  PLEASE BE PATIENT.
```

Fig. 6 Instructions for entering data for the example of Appendix C.

```
INPUT DATA
----------

NUMBER OF INDEPENDENT VARIABLES.......................N =   2
MAXIMUM NUMBER OF ALLOWABLE ITERATIONS...............MAX =  50
INTERMEDIATE PRINTOUT AT EVERY IPT ITERATIONS........IPT =   0
STARTING VALUE FOR VECTOR X(I)
     X( 1) =  0.
     X( 2) =  0.
TEST QUANTITIES TO BE USED
   EPS( 1) =    .100000E-05
   EPS( 2) =    .100000E-05
ESTIMATE OF LOWER BOUND OF FUNCTION TO BE MINIMIZED..EST =  0.
```

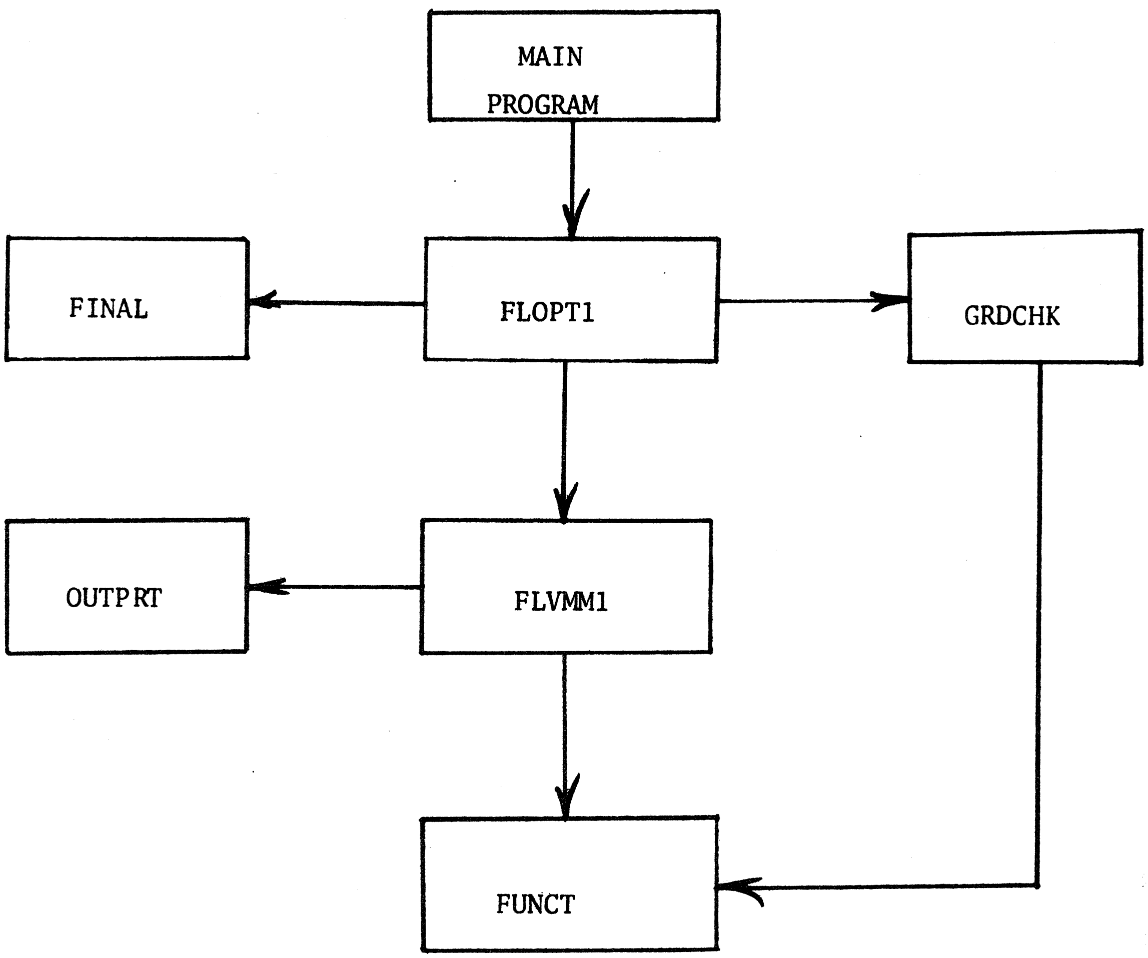
Fig. 7 Input data for the example of Appendix C.

```
IEXIT = 1
CRITERION FOR OPTIMUM (CHANGE IN VECTOR X .LT. EPS) HAS BEEN SATISFIED


OPTIMAL SOLUTION FOUND BY FLETCHER METHOD
-----------------------------------------

    U =    .499975E+00

X( 1) =    .499975E+00          G( 1) =  -.335586E-09
X( 2) =    .249988E+00          G( 2) =  -.672216E-09

    NUMBER OF FUNCTION EVALUATIONS =     7

         EXECUTION TIME IN SECONDS =   .014

DO YOU WANT TO TERMINATE THE PROGRAM
#      YES

 TOTAL NO. OF FUNCTION EVALUATIONS =   33

   TOTAL EXECUTION TIME IN SECONDS =   .072

   EXIT
```

Fig. 8 Results for the example of Appendix C.

APPENDIX D

The general structure of the package is



Attached is the complete FORTRAN listing of the package - option FLOPT1I.

```
      SUBROUTINE ELOPT1 (N,K,M,X,G,H,EPS,XS,JOPT)
      COMMON /BLK/ KO
      COMMON /BLK1/ NE
      LOGICAL UNITH
      DIMENSION X(N), G(N), EPS(N), H(K), XS(N)
      DATA D/2HNO/
      UNITH=.TRUE.
      IF (M.EQ.0) GO TO 9
      TIMET=0.0
      NET=0
      WRITE (6,24)
      WRITE (6,25)
      READ (5,*) MAX
      WRITE (6,26)
      READ (5,*) IPT
      WRITE (6,27)
      READ (5,*) ID
      WRITE (6,28)
      READ (5,*) EST
      WRITE (6,29)
      READ (5,*) X
      WRITE (6,30)
      READ (5,*) EPS
1     WRITE (6,17)
      READ (5,18) ANS
      IF (ANS.EQ.D) GO TO 8
      WRITE (6,19)
      READ (5,*) NN
      GO TO (2,3,4,5,6,7), NN
2     WRITE (6,25)
      READ (5,*) MAX
      GO TO 1
3     WRITE (6,26)
      READ (5,*) IPT
      GO TO 1
4     WRITE (6,27)
      READ (5,*) ID
      GO TO 1
5     WRITE (6,28)
      READ (5,*) EST
      GO TO 1
6     WRITE (6,29)
      READ (5,*) X
      GO TO 1
7     WRITE (6,30)
      READ (5,*) EPS
      GO TO 1
8     WRITE (6,31)
      IF (ID.EQ.0) GO TO 9
      WRITE (6,32)
      WRITE (6,33) N
      WRITE (6,34) MAX
      WRITE (6,35) IPT
      WRITE (6,36)
      WRITE (6,37) (I,X(I),I=1,N)
      WRITE (6,38)
      WRITE (6,39) (I,EPS(I),I=1,N)
      WRITE (6,40) EST
9     CONTINUE
```

```
      IF (JOPT.EQ.1) GO TO 11
      DO 10 I=1,N
      XS(I)=X(I)
10    CONTINUE
11    NF=1
      CALL GRDCHK (N,X,G)
      IF (IPT.EQ.0) GO TO 12
      WRITE (6,41)
      WRITE (6,42)
      WRITE (6,43)
12    CALL SECOND (T1)
      CALL FLVMM1 (N,K,X,U,G,H,UNITH,EST,EPS,MAX,IPT)
      CALL SECOND (T2)
      CALL FINAL (N,X,U,G)
      T=T2-T1
      WRITE (6,44) T
      TIMET=TIMET+T
      NFT=NFT+NF
      WRITE (6,20)
      READ (5,18) ANS
      IF (JOPT.EQ.1) GO TO 13
      GO TO 14
1     IF (ANS.EQ.D) GO TO 16
      WRITE (6,21) NFT
      WRITE (6,22) TIMET
      CALL EXIT
14    IF (ANS.NE.D) GO TO 16
      WRITE (6,23)
      READ (5,18) ANS
      IF (ANS.EQ.D) GO TO 1
      DO 15 I=1,N
      X(I)=XS(I)
15    CONTINUE
      GO TO 1
16    RETURN
C
C
17    FORMAT (1H0,*ANY MODIFICATION*/* , *)
18    FORMAT (A10)
19    FORMAT (1H ,*WHICH ENTRY*/* , *)
20    FORMAT (1H0/1H ,*DO YOU WANT TO TERMINATE THE PROGRAM*/* , *)
21    FORMAT (1H0,1X,*TOTAL NO. OF FUNCTION EVALUATIONS =*,I5)
22    FORMAT (1H0,3X,*TOTAL EXECUTION TIME IN SECONDS =*,F7.3/)
23    FORMAT (1H ,*SHALL I RETAIN THE STARTING VALUES OF THE VARIABLE PA
     1RAMETERS*/* , *)
24    FORMAT (1H1,*YOU ARE WELCOME TO USE THE PACKAGE ' F L O P T 1 '.*/
     1* PLEASE SUPPLY DATA WHEN ASKED FOR.  YOU CAN ENTER YOUR DATA IN A
     2NY*/* FORMAT, HOWEVER, BE REASONABLE.  PLEASE SEPARATE EACH VALUE
     3BY A*/* COMMA, A BLANK OR TYPING THE RETURN KEY.  THANK YOU.*/)
25    FORMAT (1H ,*SPECIFY THE MAXIMUM NUMBER OF ITERATIONS ALLOWED.*/*
     11( *)
26    FORMAT (1H ,*ENTER AN INTEGER SO THAT INTERMEDIATE OUTPUT WILL BE
     1PRINTED AFTER*/* EVERY SPECIFIED NUMBER OF ITERATIONS.  ENTER 0 IF
     2 YOU DON'T WANT*/* ANY INTERMEDIATE OUTPUT.*/* 2( *)
27    FORMAT (1H ,*ENTER 1 IF YOU WANT TO HAVE A RECORD OF YOUR INPUT DA
     1TA, OTHERWISE*/* ENTER 0.*/* 3( *)
28    FORMAT (1H ,*SPECIFY A MINIMUM ESTIMATED VALUE OF THE OBJECTIVE FU
     1NCTION.*/* 4( *)
29    FORMAT (1H ,*ENTER STARTING VALUES FOR THE VARIABLE PARAMETERS.*/*
```

```
     1 5( *)
30    FORMAT (1H ,*ENTER SMALL VALUES FOR TESTING CONVERGENCE.*/* 6( *)
31    FORMAT (//,* YOUR DATA IS NOW BEING PROCESSED.  IT MAY TAKE SOME T
     1IME BEFORE*/* RESULTS ARE AVAILABLE.  PLEASE BE PATIENT.*)
32    FORMAT (1H1,*INPUT DATA*/,1H ,10(*-*))
33    FORMAT (1H0,*NUMBER OF INDEPENDENT VARIABLES*,24(*.*),*N =*,I4)
34    FORMAT (1H ,*MAXIMUM NUMBER OF ALLOWABLE ITERATIONS*,15(*.*),*MAX
     1=*,I4)
35    FORMAT (1H ,*INTERMEDIATE PRINTOUT AT EVERY IPT ITERATIONS*,8(*.*)
     1,*IPT =*,I4)
36    FORMAT (1H ,*STARTING VALUE FOR VECTOR X(I)*)
37    FORMAT (1H ,5X,*X(*,I2,*) =*,E14.6)
38    FORMAT (1H ,*TEST QUANTITIES TO BE USED*)
39    FORMAT (1H ,3X,*EPS(*,I2,*) =*,E14.6)
40    FORMAT (1H ,*ESTIMATE OF LOWER BOUND OF FUNCTION TO BE MINIMIZED*,
     12(*.*),*EST =*,E14.6)
41    FORMAT (1H1)
42    FORMAT (1H ,*OPTIMIZATION BY FLETCHER METHOD*/,1H ,31(*-*))
43    FORMAT (1H0,*ITER.*,2X,*FUNCT.*,5X,*TIME(SEC)*,5X,*OBJECTIVE*,6X,*
     1VARIABLE*,7X,*GRADIENT*/1H ,1X,*NO.*,3X,*EVALU.*,6X,*ELAPSED*,6X,*
     2FUNCTION*,6X,*VECTOR X(I)*,4X,*VECTOR G(I)*,/)
44    FORMAT (1H0,9X,*EXECUTION TIME IN SECONDS =*,F7.3)
      END
```

```
      SUBROUTINE ELVMM1 (N,K,X,U,G,H,UNITH,EST,EPS,MAX,IPT)
      DIMENSION X(N), G(N), EPS(N), H(K)
      COMMON /BLK1/ NF
      COMMON /BLK/ KO
      LOGICAL CONV,UNITH
      CALL SECOND (T3)
      KO=0
      CALL FUNCT (X,G,U)
      IF (U.LT.EST) GO TO 23
      ITN=1
      STEP=1.0
      ID=N+N
      IH=ID+N
      IF (.NOT.UNITH) GO TO 2
      IJ=IH+1
      DO 1 I=1,N
      DO 1 J=I,N
      H(IJ)=0.0
      IF (I.EQ.J) H(IJ)=1.0
    1 IJ=IJ+1
    2 CONV=.TRUE.
      GDX=0.0
      DO 6 I=1,N
      Z=0.0
      IJ=IH+I
      IF (I.EQ.1) GO TO 4
      L=I-1
      DO 3 J=1,L
      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(N+I)=Z
      GDX=GDX+G(I)*Z
    6 CONTINUE
      IF (IPT.EQ.0) GO TO 7
      IF (MOD(ITN,IPT).NE.0) GO TO 7
      CALL SECOND (T4)
      TIME=T4-T3
      CALL OUTPRT (X,N,G,U,NF,ITN,TIME)
    7 IEXIT=1
      IF (CONV) GO TO 24
      IEXIT=2
      IF (GDX.GE.0.0) GO TO 24
      Z=1.0
      IF (ITN.LT.N.AND.UNITH) Z=STEP
      W=2.*(EST-U)/GDX
      IF (W.LT.Z) Z=W
      STEP=Z
    8 GDX=GDX*Z
      DO 9 I=1,N
      H(N+I)=H(N+I)*Z
      X(I)=X(I)+H(N+I)
    9 CONTINUE
      NF=NF+1
```

```
      CALL FUNCT (X,H,FP)
      IF (FP.LT.EST) GO TO 23
      IEXIT=3
      IF (ITN.EQ.MAX) GO TO 24
      GPDX=0.0
      DO 10 I=1,N
      H(ID+I)=H(I)-G(I)
      GPDX=GPDX+H(I)*H(N+I)
10    CONTINUE
      DGDX=GPDX-GDX
      IF (U.GT.FP-0.0001*GDX) GO TO 12
      IEXIT=4
      IF (GPDX.LT.0.0.AND.ITN.GT.N) GO TO 24
      Z=3.*(U-FP)+GPDX+GDX
      W=SQRT(1.0-GDX/Z*GPDX/Z)*ABS(Z)
      Z=1.0-(GPDX+W-Z)/(DGDX+2.0*W)
      IF (Z.LT.0.1) Z=0.1
      DO 11 I=1,N
      X(I)=X(I)-H(N+I)
11    CONTINUE
      GO TO 14
12    U=FP
      DO 13 I=1,N
      G(I)=H(I)
13    CONTINUE
      IF (DGDX.GT.0.0) GO TO 15
      GDX=GPDX
      Z=4.0
14    STEP=Z*STEP
      GO TO 8
15    IF (GPDX.LT.0.5*GDX) STEP=2.0*STEP
      DGHDG=0.0
      DO 19 I=1,N
      Z=0.0
      IJ=IH+I
      IF (I.EQ.1) GO TO 17
      L=I-1
      DO 16 J=1,L
      Z=Z+H(IJ)*H(ID+J)
      IJ=IJ+N-J
16    CONTINUE
17    DO 18 J=I,N
      Z=Z+H(IJ)*H(ID+J)
      IJ=IJ+1
18    CONTINUE
      DGHDG=DGHDG+Z*H(ID+I)
      H(I)=Z
19    CONTINUE
      IF (DGHDG.LT.0.0) DGHDG=DGDX*0.01
      IF (DGDX.LT.DGHDG) GO TO 21
      W=1.0+DGHDG/DGDX
      DO 20 I=1,N
      H(N+I)=W*H(N+I)-H(I)
20    CONTINUE
      DGDX=DGDX+DGHDG
      DGHDG=DGDX
21    IJ=IH
      DO 22 I=1,N
      W=H(N+I)/DGDX
```

```
      Z=H(I)/DGHDG
      DO 22 J=I,N
      IJ=IJ+1
22    H(IJ)=H(IJ)+W*H(N+J)-Z*H(J)
      ITN=ITN+1
      GO TO 2
23    IEXIT=5
24    IF (IEXIT.EQ.1) KO=1
      GO TO (25,26,27,26,28), IEXIT
25    WRITE (6,30) IEXIT
      GO TO 29
26    WRITE (6,31) IEXIT
      GO TO 29
27    WRITE (6,32) IEXIT
      GO TO 29
28    WRITE (6,33) IEXIT
29    CONTINUE
      RETURN
C
C
30     FORMAT (1H1,*IEXIT =*,I2,/1H ,*CRITERION FOR OPTIMUM (CHANGE IN VE
     1CTOR X .LT. EPS) HAS BEEN SATISFIED*)
31     FORMAT (1H1,*IEXIT =*,I2,/1H ,*EITHER OF THE FOLLONIMG THINGS HAS
     1HAPPENED*,/10X,*1. EPS CHOSEN IS TOO SMALL*,/10X,*2. GRADIENTS ARE
     2 NOT CORRECT*,/10X,*3. MATRIX H GOES SINGULAR*)
32     FORMAT (1H1,*IEXIT =*,I2,/1H ,*MAXIMUM NUMBER OF ALLOWABLE ITERATI
     1ON HAS BEEN EXCEEDED*)
33     FORMAT (1H1,*IEXIT =*,I2,/1H ,*FUNCTION VALUE LESS THAN MINIMUM ES
     1TIMATED HAS BEEN DETECTED*)
       END
```

```
      SUBROUTINE GRDCHK (N,X,G)
      DIMENSION X(N), G(N), T(100)
      IC=0
      CALL FUNCT (X,G,U)
      WRITE (6,3)
      WRITE (6,4)
      DO 1 I=1,N
      DX=1.E-4*X(I)
      IF (ABS(DX).LT.1.E-10) DX=1.E-10
      X(I)=X(I)+DX
      CALL FUNCT (X,T,F2)
      X(I)=X(I)-2.0*DX
      CALL FUNCT (X,T,F1)
      Y=0.5*(F2-F1)/DX
      X(I)=X(I)+DX
      IF (ABS(Y).LT.1.E-14) Y=1.E-14
      IF (ABS(G(I)).LT.1.E-14) G(I)=1.E-14
      YP=ABS((Y-G(I))/Y)*100.0
      WRITE (6,5) G(I),Y,YP
      IF (YP.GT.10.0) IC=1
1     CONTINUE
      IF (IC.EQ.1) GO TO 2
      WRITE (6,6)
      RETURN
2     WRITE (6,7)
      CALL EXIT
C
C
3     FORMAT (1H1,*GRADIENT CHECK AT STARTING POINT*/1H ,32(*-*))
4     FORMAT (1H0,5X,*ANALYTICAL GRADIENTS*,5X,*NUMERICAL GRADIENTS*,5X,
     1*PERCENTAGE ERROR*)
5     FORMAT (1H ,8X,F14.6,10X,F14.6,8X,F14.6)
6     FORMAT (1H0,*GRADIENTS ARE O.K.*)
7     FORMAT (1H0,*YOUR PROGRAM HAS BEEN TERMINATED BECAUSE GRADIENTS AR
     1E INCORRECT*/1H ,*PLEASE CHECK IT AGAIN*)
      END
```

```
      SUBROUTINE OUTPRT (X,N,G,U,NF,ITN,TIME)
      DIMENSION X(N), G(N)
      WRITE (6,1) ITN,NF,TIME,U,((X(I),G(I)),I=1,N)
      RETURN
C
C
1     FORMAT (1H ,I3,5X,I3,6X,F10.3,1X,F14.6,1X,80(F14.6,1X,F14.6,/,44X)
     1)
      END
```

```
      SUBROUTINE FINAL (N,X,U,G)
      DIMENSION X(N), G(N)
      COMMON /BLK/ KO
      COMMON /BLK1/ NF
      IF (KO.EQ.0) GO TO 1
      WRITE (6,3)
      GO TO 2
1     WRITE (6,4)
2     CONTINUE
      WRITE (6,5) U
      WRITE (6,6) (I,X(I),I,G(I),I=1,N)
      WRITE (6,7) NF
      RETURN
C
C
3     FORMAT (1H0,/1H0,*OPTIMAL SOLUTION FOUND BY FLETCHER METHOD*/1H ,4
     11(*-*))
4     FORMAT (1H0,/1H0,*RESULTS FOUND BY FLETCHER METHOD AT LAST ITERATI
     1ON*/1H ,50(*-*))
5     FORMAT (1H0,4X,*U =*,F14.6/)
6     FORMAT (1H ,*X(*,I2,*) =*,F14.6,8X,*G(*,I2,*) =*,F14.6)
7     FORMAT (1H0,4X,*NUMBER OF FUNCTION EVALUATIONS =*,I5)
      END
```