INTERNAL REPORTS IN

SIMULATION, OPTIMIZATION AND CONTROL

No. SOC-84

FLOPT2- A PROGRAM FOR LEAST PTH OPTIMIZATION WITH EXTRAPOLATION TO MINIMAX SOLUTIONS

J.W. Bandler and W.Y. Chu

April 1975

FACULTY OF ENGINEERING
McMASTER UNIVERSITY
HAMILTON, ONTARIO, CANADA

FLOPT2- A PROGRAM FOR LEAST PTH OPTIMIZATION WITH EXTRAPOLATION TO MINIMAX SOLUTIONS

J. W. Bandler and W. Y. Chu

Abstract FLOPT2 is a package of subroutines primarily for solving least pth optimization problems. Its main features include Fletcher's quasi-Newton subroutine, a least pth objective formulation subroutine, an extrapolation procedure and a scheme for dropping inactive functions. With appropriate utilization of these features, the program can solve a wide variety of optimization problems. These may range from unconstrained problems, problems subject to inequality or equality constraints to nonlinear minimax approximation problems. In solving constrained problems, the user may, for example, use the Fiacco-McCormick method with extrapolation or the Bandler-Charalambous minimax formulation and least pth approximation, also with extrapolation. The program has been used on a CDC 6400 computer. Several examples of varying complexity are used to illustrate the versatility of the program. A FORTRAN IV listing is included.

This work was supported by the National Research Council of Canada under Grant A7239 and by the Defence Research Board of Canada under Grant 9931-39.

The authors are with the Group on Simulation, Optimization and Control and Department of Electrical Engineering, McMaster University, Hamilton, Canada, L8S 4L7.

I. INTRODUCTION

FLOPT2 is a package of subroutines primarily for solving least pth optimization problems. Its main features include the 1972 version of Fletcher's quasi-Newton subroutine [1], a least pth objective formulation subroutine, an extrapolation procedure and a scheme for dropping inactive functions [2]. With appropriate utilization of these features, the program can solve a wide variety of optimization problems. These may range from unconstrained problems, problems subject to inequality/equality constraints to minimax problems in general.

In solving constrained problems, the user may use the Fiacco-McCormick method with extrapolation [3] or use the Bandler-Charalambous minimax formulation [4] and least pth approximation. Using the p-algorithm [2], the program solves minimax problems that can be formulated with a least pth objective.

The program FLOPT2 is an improved version of the program FLOPT1 [5]. It may functionally replace the program FLNLP2 [6]. The program has been used on a CDC 6400 computer and is written in FORTRAN IV. It requires at least 4,630 octal words of computer memory. Several examples of varying complexity are used to illustrate the versatility of the program. Up to 150 functions can be currently handled.

II. ARGUMENT LIST

SUBROUTINE FLOPT2 (N, M, IGK, X, G, H, W, EPS, XE, IH, IK, FACTOR, XB, IFINIS, NR)

The arguments are as follows.

N — An integer to be set to the number of variables ($N \geq 2$).

M — An integer to be set to 1 if input data is to be read. Otherwise, set to zero.

IGK An integer to be set to 1 if a gradient check by perturbation is desired. Otherwise, set to any other value. Also, gradient check is not performed when input data is not read.

X A real array of N elements in which the current estimate of the solution is stored. An initial approximation must be set in X on entry. When the extrapolation procedure is used, an estimate of the next minimum in the sequence will be stored on exit of each cycle of optimization.

G A real array of N elements in which the gradient vector corresponding to X above will be returned. When the extrapolation procedure is used, the optimal solution of each cycle of optimization will be returned in G on exit.

H A real array of N*(N+1)/2 elements in which an estimate of the Hessian matrix is stored.

W A real array of 4*N elements used as working space.

EPS A real array of N elements to be set to the test quantities used in Fletcher's program.

XE A real array of NxIKx(JORDER+1) elements in which different orders of estimates of the minimax solution are stored when extrapolation is used.

IH An integer to be set to 1 if a single value of p is used. When a sequence of p values is used, IH should be set as the index of a DO loop that calls SUBROUTINE FLOPT2 IK times.

IK An integer to be set to the maximum number of cycles of optimization. It corresponds to the number of p values when extrapolation is used.

FACTOR To be defined in Section III.

XB A real array of N elements in which the best estimate of the minimax solution currently available is stored.

IFINIS An integer whose value will be equal to N when the convergence criterion for the estimates of the minimax solution is met.

NR An integer to be set to the total number of error functions. When the least pth objective formulation is NOT used, it should be set to 1.

III. INPUT DATA

Parameters to be supplied as input data are defined as follows.

MAX An integer to be set to the maximum number of iterations allowed.

IPT An integer controlling printing of intermediate output. Printing occurs every |IPT| iterations and also on exit except when IPT is set to zero in which case intermediate output is suppressed.

ID An integer to be set to 1 if input data is to be printed. Otherwise, set to zero.

IREDU An integer to be set to 1 if the scheme for dropping inactive functions is used; otherwise set to zero.

EST A real number to be set to the estimated minimum value of the objective function.

ETA A positive real number to be set to the multiplier tolerance for dropping inactive functions.

PO A real number to be set to the value of p used in the least pth formulation or the initial value of p when a sequence of p values is used.

FACTOR — A positive real number to be set to the multiplying factor for p when a sequence of p values is used together with extrapolation; otherwise set to 1. When extrapolation is used with the Fiacco-McCormick method, it is the factor by which the sequence of r is decreased.

X(I) I=1,N — Starting values for the variables $x_1, x_2, \ldots, x_n$ defined in Section II.

EPS(I) I=1,N — As defined in Section II.

IEX — An integer to be set to 1 if the extrapolation procedure is used; otherwise set to zero.

JORDER — An integer to be set to the highest order of estimates used in extrapolation (JORDER $\leq$ IK-1); otherwise set to zero.

JPRINT — An integer controlling printing of results of the extrapolation procedure and the reduction scheme. If JPRINT is set to:

- 0 nothing will be printed,
- 1 the estimates of the minimax solution and the error functions at the highest order estimate will be printed,
- 2 in addition to the above printout, multipliers and normalized errors at the next estimated least pth solution will be printed (except when IH=IK, which indicates the final optimization).

The input data is to be read in the following format:

CARD No.	FORMAT	PARAMETERS
1	4I5	MAX, IPT, ID, IREDU
2	5E16.8	EST, ETA, PO, FACTOR
As many as required	5E16.8	X(I), I = 1, N

CARD No.	FORMAT	PARAMETERS
As many as required	5E16.8	EPS(I), I = 1,N
Last	4I5	IEX, JORDER, JPRINT

IV. USER SUBROUTINES

The user has to supply the main program and a subroutine called FUNCT which defines the error functions and their first partial derivatives with respect to the variable parameters. If the least pth formulation is not used, the objective function also has to be defined.

In the main program, the user has to supply the values and proper dimensioning for the parameters in the argument list of subroutine FLOPT2. In using the extrapolation feature, the subroutine FLOPT2 has to be called a number of times. This may be done, for example, by

```
          .
          .
          IK = 5
          M = 1
          DO 1 IH = 1, IK
          CALL FLOPT2 (N, M, IGK, X, G, H, W, EPS, XE, IH, IK, FACTOR, XB,
     1    IFINIS, NR)
          M = 0
          .
          .
          .
1         CONTINUE
          .
          .
```

Depending on the objective formulations and options used, the subroutine FUNCT may assume different forms. Here, we present its form when a least pth objective formulation and the reduction scheme are used:

```
      SUBROUTINE FUNCT (X,G,U)
      DIMENSION X(N), G(N), ER(NR), GE(N,NR), ES(NR)
      COMMON/WY3/NA, JD(150)
```

where

N is the number of independent variables $\underset{\sim}{x}$,

NR is the total number of error functions,

NA is the number of active error functions (determined automatically by the program),

JD is an integer array used as an index set (set automatically by the program).

```
      DO 99 I = 1, NA
      K = JD(I)
      GO TO (1,2,...,NR), K
1     ER(1) = e1(x1,x2,...,xn)
      GE(1,1) = partial derivative of e1 w.r.t. x1
        .
        .
        .
      GE(N,1) = partial derivative of e1 w.r.t. xn
      GO TO 99
2     ER(2) = e2(x1,x2,...,xn)
      GE(1,2) = partial derivative of e2 w.r.t. x1
        .
        .
        .
```

```
          GE(N,2) = partial derivative of e_2 w.r.t. x_n
          GO TO 99
3         .
          .
          .
          .
          .
NR        ER(NR) = e_NR(x_1,x_2,...,x_n)
          GE(1,NR) = partial derivative of e_NR w.r.t. x_1
          .
          .
          .
          GE(N,NR) = partial derivative of e_NR w.r.t. x_n
99        CONTINUE
          CALL LEASTP (N, U, G, ER, GE, ES)
          RETURN
          END
```

The LEASTP subroutine will formulate the objective function U and its first partial derivatives (stored in array G). It should be noted that the error functions may be defined in another subprogram which is called by subroutine FUNCT (see Example 5 in Section VI). The minimal form of subroutine FUNCT can be found in Example 1.

V. OTHER SUBROUTINES

The following is a brief description of the subroutines called by FLOPT2.

LEASTP formulates a least pth objective function and the necessary gradients.

GRDCHK checks the gradient formulation by perturbation.

QUASIN minimizes a function using the Fletcher unconstrained minimization program by quasi-Newton methods.

RESULT outputs the optimal solution.

EXTRAP performs extrapolation.

The overall structure of the program is shown in Fig. 1.

VI. EXAMPLES

An unconstrained problem, two constrained problems, a minimax example and a microwave circuit example are used to illustrate the flexibility and power of the program. For each example, the main program, the subroutine FUNCT and the necessary input data are illustrated. The initial estimate of the Hessian matrix (required in Fletcher's program) was set to the unit matrix for the first optimization. In subsequent optimizations, the Hessian matrix estimated at the previous minimum was used. For all examples, the test quantities (EPS (I), I = 1,...,N) were 10^{-8}.

Example 1: Rosenbrock's function [7]

Minimize

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

The function has a minimum value of zero at $x_1 = x_2 = 1$. The starting point used was $x_1 = -1.2$, $x_2 = 1.0$.

A listing of the main program, subroutine FUNCT, input data and final results are given in Figs. 2-3. Note that the least pth formulation was not used and there was no extrapolation or reduction scheme (NR, IEX and IREDU were set to 1, 0 and 0, respectively). The subroutine FUNCT is shown in its simplest possible form.

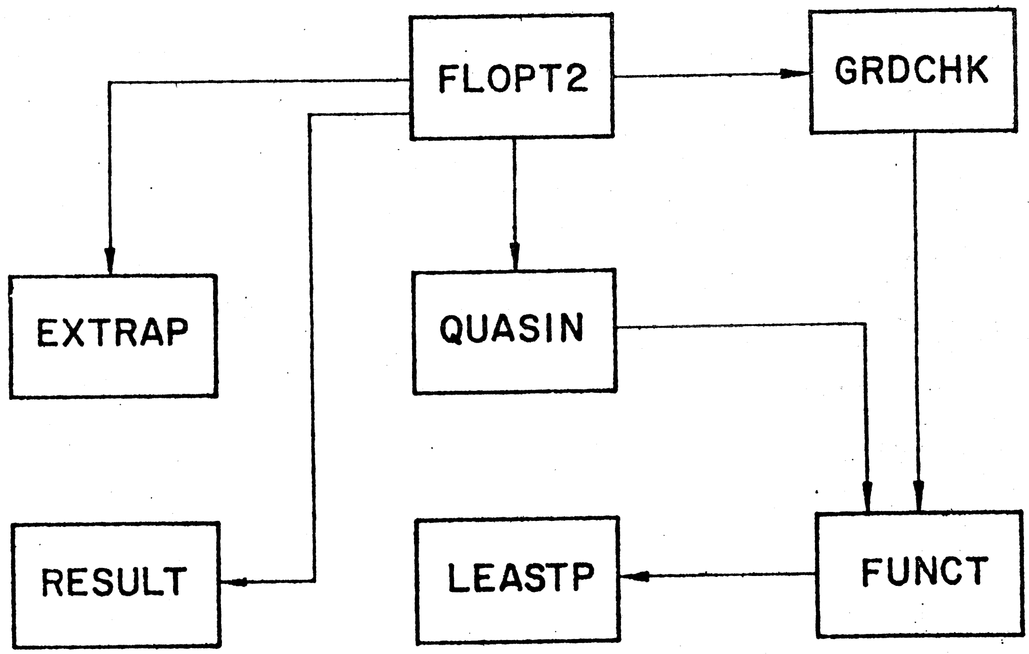

Fig. 1 Overall structure of FLOPT2.

```
      PROGRAM TST (INPUT,OUTPUT,TAPE5=INPUT,TAPE6=OUTPUT)
C     MAIN PROGRAM OF EXAMPLE 1
C
      DIMENSION X(2),G(2),H(3),W(8),EPS(2),XB(2),XF(2,1,1)
      N=2
      NP=1
      M=1
      IGK=1
      IH=1
      IK=1
      CALL FLOPT2(N,M,IGK,X,G,H,W,EPS,XF,IH,IK,FACTOR,XB,IFINIS,NP)
      STOP
      END


      SUBROUTINE FUNCT(X,G,U)
C     ROSENBROCK'S FUNCTION
C
      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
      G(1)=400.*X(1)*(A-X(2))-C-C
      G(2)=-200.*B
      RETURN
      END


C
C     INPUT DATA
C
  100     5     1     0
0.0               0.0               1.0               1.0
-1.2              1.0
1.E-             81.E-              8
     0     0     0
```

Fig. 2 Main program and subroutine FUNCT for Example 1.
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 =    5

STARTING VALUE FOR VECTOR X(I)......................X( 1) =  -.12000000E+0
                                                    X( 2) =   .10000000E+0

TEST QUANTITIES TO BE USED........................EPS( 1) =   .10000000E-0
                                                  EPS( 2) =   .10000000E-0

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




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

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

     U =    .25874638E-24

   X( 1) =    .10000000E+01      G( 1) =  -.76028073E-11
   X( 2) =    .10000000E+01      G( 2) =   .42632564E-11

    NUMBER OF FUNCTION EVALUATIONS =    47
        EXECUTION TIME IN SECONDS =    .121
```

Fig. 3 A record of the input data and final results for Example 1.

Example 2: Beale constrained function [8]

Minimize

$$f(x) = 9-8x_1-6x_2-4x_3+2x_1^2+2x_2^2+x_3^2+2x_1x_2+2x_1x_3$$

subject to

$$x_i \geq 0 \quad , \quad i = 1,2,3$$

$$3-x_1-x_2-2x_3 \geq 0.$$

The function has a minimum $f(\check{\underset{\sim}{x}}) = 1/9$ at $\check{\underset{\sim}{x}} = [4/3\ 7/9\ 4/9]^T$. The SUMT method of Fiacco and McCormick [3] was used to transform the constrained problem into an unconstrained problem by defining

$$U(\underset{\sim}{x},r) = f(\underset{\sim}{x}) - r \sum_{i=1}^{3} \ln g_i(\underset{\sim}{x}).$$

The objective function U was minimized w.r.t. $\underset{\sim}{x}$ for a strictly decreasing sequence of r values together with extrapolation. The starting point was $\underset{\sim}{x} = [1 \quad 2 \quad 1]^T$. Fig. 4 shows a listing of the main program, subroutine FUNCT and input data. A COMMON block named USER was used in the main program and subroutine FUNCT to transfer the value of the parameter r, a weighting factor WT and an indicator IGRAD. WT was used in the formulation of the unconstrained objective function only when the process got into the nonfeasible region. IGRAD is an indicator to control the printing of the original objective function and constraints at the extrapolated solution which is available from the argument list of subroutine FLOPT2. Note that the parameter r has to be decreased by the user in the main program. The factor was available from the argument list. The reduction scheme cannot be used in this example.

With the sequence of r values 10^{-2}, $2x10^{-3}$, $4x10^{-4}$, $8x10^{-5}$, $1.6x10^{-5}$ and 3rd order extrapolation, it took 40 function evaluations to reach the

```
      PROGRAM TST (INPUT,OUTPUT,TAPE5=INPUT,TAPE6=OUTPUT)
C     MAIN PROGRAM OF EXAMPLE 2
C
      DIMENSION X(3),G(3),H(6),W(12),EPS(3),XB(3),XE(3,6,4)
      COMMON/USER/R,WT,IGRAD
      N=3
      NR=1
      M=1
      IGK=1
      IK=6
      WT=1.E+16
      R=0.01
      DO 1 IH=1,IK
      IGRAD=1
      CALL FLOPT2(N,M,IGK,X,G,H,W,EPS,XE,IH,IK,FACTOR,XB,IFINIS,NR)
      M=0
      IGRAD=0
      CALL FUNCT(XB,G,U)
      IF(IFINIS.EQ.N) CALL EXIT
      R=R/FACTOR
1     CONTINUE
      STOP
      END


      SUBROUTINE FUNCT(X,G,U)
C     BEALE FUNCTION
C
      DIMENSION X(3),G(3),C(4),GF(3),GC(3,4)
      COMMON/USER/R,WT,IGRAD
      B=X(1)+X(1)
      D=X(2)+X(2)
      E=X(3)+X(3)
      F=9.+B*(X(1)+X(2)+X(3)-4.)+D*(X(2)-3.)+X(3)*X(3)-E-E
      C(1)=X(1)
      C(2)=X(2)
      C(3)=X(3)
      C(4)=3.-X(1)-X(2)-E
      IF(IGRAD.EQ.0) GO TO 1
      GF(1)=-8.+B+B+D+E
      GF(2)=-6.+D+D+B
      GF(3)=-4.+B+E
      GC(1,1)=1.
      GC(2,1)=0.
      GC(3,1)=0.
      GC(1,2)=0.
      GC(2,2)=1.
      GC(3,2)=0.
      GC(1,3)=0.
      GC(2,3)=0.
      GC(3,3)=1.
      GC(1,4)=-1.
      GC(2,4)=-1.
      GC(3,4)=-2.
      C1=0.
      C2=0.
      DO 13 I=1,4
      IF (C(I).LT.1.E-6) GO TO 14
```

```
      S1=S1-ALOG(C(I))
      GO TO 13
14    S2=S2+WT*C(I)*C(I)
13    CONTINUE
      U=F+R*S1+S2
      DO 12 J=1,3
      S3=0.
      S4=0.
      DO 11 I=1,4
      IF (C(I).LT.1.E-6) GO TO 15
      S3=S3-GC(J,I)/C(I)
      GO TO 11
15    S4=S4+(WT+WT)*C(I)*GC(J,I)
11    CONTINUE
      G(J)=GF(J)+S3*R+S4
12    CONTINUE
      RETURN
1     PRINT 2,(X(I),I=1,3)
2     FORMAT(//1H0,*SOLUTION*,7X,3E16.8)
      PRINT 3,F
3     FORMAT(*0OBJECTIVE FUNCTION      *,E16.8)
      PRINT 4,(C(I),I=1,4)
4     FORMAT(*0CONSTRAINTS      *,4E16.8)
      RETURN
      END

C
C     INPUT DATA
C
  100     5     1     0
0.0               0.0               0.0               5.0
1.0               2.0               1.0
1.E-             81.E-             81.E-              8
    1     3     1
```

Fig. 4 Main program and subroutine FUNCT for Example 2. Input data is also shown.

following solution

$$f(\underset{\sim}{x}) = 0.1111111$$

$$x_1 = 1.3333333$$

$$x_2 = 0.7777778$$

$$x_3 = 0.4444445.$$

There was only one active constraint and its value was of the order of 10^{-14}. Fig. 5 shows the results obtained at the end of the 5th optimization.

Example 3: A minimax example [9]

Minimize the maximum of the following three functions

$$e_1(\underset{\sim}{x}) = x_1^2+x_2^4$$

$$e_2(\underset{\sim}{x}) = (2-x_1)^2+(2-x_2)^2$$

$$e_3(\underset{\sim}{x}) = 2\exp(-x_1+x_2).$$

The minimax solution is defined by the functions e_1 and e_2 at the point $x_1 = 1.13904$, $x_2 = 0.89956$ where $e_1 = e_2 = 1.95222$ and $e_3 = 1.57408$. Using the p-algorithm with p = 4, 16, 64, 256, 1024, 46 function evaluations yielded $x_1 = 1.1390346$, $x_2 = 0.8995623$. All the three functions were used in the initial objective formulation. The reduction scheme reduced to the two active functions at the end of the process. Fig. 6 shows the main program, the subroutine FUNCT and input data. Note that in using the reduction scheme, the user has to supply a statement defining the COMMON block WY3 in the subroutine FUNCT. A printout of the input data is shown in Fig. 7. Fig. 8 shows the results obtained at the end of the first optimization. Information to be used by the reduction scheme is also shown. Intermediate and final estimates of the minimax solution are shown in Fig. 9.

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

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

     U =    .11112350E+00

 X( 1) =    .13333299E+01       G( 1) =  -.22224279E+00
 X( 2) =    .77777948E+00       G( 2) =  -.22224284E+00
 X( 3) =    .44444531E+00       G( 3) =  -.44448557E+00

    NUMBER OF FUNCTION EVALUATIONS =     5
         EXECUTION TIME IN SECONDS =    .018

ESTIMATES OF THE MINIMAX SOLUTION BY EXTRAPOLATION
--------------------------------------------------

ORDER  1
X( 1) =   .13333333E+01
X( 2) =   .77777777E+00
X( 3) =   .44444445E+00
ORDER  2
X( 1) =   .13333333E+01
X( 2) =   .77777776E+00
X( 3) =   .44444445E+00
ORDER  3
X( 1) =   .13333333E+01
X( 2) =   .77777776E+00
X( 3) =   .44444445E+00

SOLUTION              .13333333E+01   .77777776E+00   .44444445E+00
OBJECTIVE FUNCTION          .11111111E+00
CONSTRAINTS           .13333333E+01   .77777776E+00   .44444445E+00   .78159701E-1
```

Fig. 5 Results for the Beale problem.

```
      PROGRAM TST (INPUT,OUTPUT,TAPE5=INPUT,TAPE6=OUTPUT)
C     MAIN PROGRAM OF EXAMPLE 3
C
      DIMENSION X(2),G(2),H(3),W(8),EPS(2),XB(2),XE(2,5,4)
      N=2
      NR=3
      M=1
      IGK=0
      IK=5
      DO 1 IH=1,IK
      CALL FLOPT2(N,M,IGK,X,G,H,W,EPS,XE,IH,IK,FACTOR,XB,IFINIS,NR)
      IF(IFINIS.EQ.N) CALL EXIT
      M=0
1     CONTINUE
      STOP
      END


      SUBROUTINE FUNCT(X,G,U)
C     A MINIMAX EXAMPLE
C
      DIMENSION X(2),G(2),FR(3),GF(2,3),FS(3)
      COMMON/WY3/NA,JD(150)
      N=2
      Y1=X(1)*X(1)
      Y2=X(2)*X(2)
      Y3=X(1)+X(1)
      Y4=X(2)+X(2)
      DO 12 I=1,NA
      K=JD(I)
      GO TO (1,2,3) ,K
1     FR(1)=Y1+Y2*Y2
      GF(1,1)=Y3
      GF(2,1)=(Y2+Y2)*Y4
      GO TO 12
2     FR(2)=8.-4.*(X(1)+X(2))+Y1+Y2
      GF(1,2)=-4.+Y3
      GF(2,2)=-4.+Y4
      GO TO 12
3     FR(3)=2.*EXP(-X(1)+X(2))
      GF(1,3)=-FR(3)
      GF(2,3)=FR(3)
12    CONTINUE
      CALL LEASTP(N,U,G,FR,GF,FS)
      RETURN
      END


C
C     INPUT DATA
C
  100     10      1      1
 [illegible].0              0.0005             4.0              4.0
 2.0              2.0
 1.E-            81.E-             8
    1       2       2
```

Fig. 6 Main program and subroutine FUNCT for Example 3. 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 =  10

STARTING VALUE FOR VECTOR X(I).....................X( 1) =   .20000000E+01
                                                   X( 2) =   .20000000E+01

TEST QUANTITIES TO BE USED.......................EPS( 1) =   .10000000E-07
                                                 EPS( 2) =   .10000000E-07

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

HIGHEST ORDER OF ESTIMATES USED IN EXTRAPOLATION..JORDER =   3

MULTIPLIER TOLERANCE FOR DROPPING FUNCTIONS..........ETA =   .50000000E-03
```

Fig. 7 Input data for the minimax example.

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

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

      U =    .24033042E+01

     EM =    .20164297E+01
 EN( 1) =    .94621380E+00
 EN( 2) =    .10000000E+01
 EN( 3) =    .68198002E+00

  X( 1) =    .12008090E+01      G( 1) =   -.24628389E-07
  X( 2) =    .82623537E+00      G( 2) =    .54009946E-07

    NUMBER OF FUNCTION EVALUATIONS =   13

          VALUE OF THE PARAMETER P =   .40000000E+01
          EXECUTION TIME IN SECONDS =   .058

MULTIPLIERS AND NORMALIZED ERRORS AT THE ABOVE SOLUTION
     EM =   .20164297E+01
 MU( 1) =   .39724138          EN( 1) =    .94621380
 MU( 2) =   .49556128          EN( 2) =   1.00000000
 MU( 3) =   .10719733          EN( 3) =    .68198003

NUMBER OF ERROR FUNCTIONS FOR THE NEXT OPTIMIZATION =   3
```

Fig. 8 Results at the end of the first optimization .

```
ORDER  1
X( 1) =   .11361326E+01
X( 2) =   .90182476E+00
ORDER  2
X( 1) =   .11370098E+01
X( 2) =   .90117757E+00
CORRESPONDING NORMALIZED ERRORS
    EM =   .19523318E+01
EN( 1) =  1.00000000
EN( 2) =   .99991343
EN( 3) =   .80919973

MULTIPLIERS AND NORMALIZED ERRORS FOR THE NEXT ESTIMATED
LEAST PTH SOLUTION
    EM =   .19531062E+01
MU( 1) =   .48334550        EN( 1) =   .99895927
MU( 2) =   .51665387        EN( 2) =  1.00000000
MU( 3) =   .00000063        EN( 3) =   .80833931

NUMBER OF ERROR FUNCTIONS FOR THE NEXT OPTIMIZATION =   2


ESTIMATES OF THE MINIMAX SOLUTION BY EXTRAPOLATION

ORDER  1
X( 1) =   .11390378E+01
X( 2) =   .89955997E+00
ORDER  2
X( 1) =   .11390377E+01
X( 2) =   .89955990E+00
ORDER  3
X( 1) =   .11390346E+01
X( 2) =   .89956228E+00
CORRESPONDING NORMALIZED ERRORS
    EM =   .19522246E+01
EN( 1) =   .99999986
EN( 2) =  1.00000000
```

Fig. 9 Intermediate and final solutions for the minimax example.

Example 4: Rosen-Suzuki function [8]

Minimize

$$f(\underset{\sim}{x}) = x_1^2+x_2^2+2x_3^2+x_4^2-5x_1-5x_2-21x_3+7x_4$$

subject to

$$-x_1^2-x_2^2-x_3^2-x_4^2-x_1+x_2-x_3+x_4+8 \geq 0$$
$$-x_1^2-2x_2^2-x_3^2-2x_4^2+x_1+x_4+10 \geq 0$$
$$-2x_1^2-x_2^2-x_3^2-2x_1+x_2+x_4+5 \geq 0.$$

The function has a minimum $f(\check{\underset{\sim}{x}}) = -44$ at $\check{\underset{\sim}{x}} = [0 \ \ 1 \ \ 2 \ \ -1]^T$. The Bandler-Charalambous technique was used to transform the nonlinear programming problem into an unconstrained minimax problem. The value of the parameter α was 10. Using the p-algorithm with p = 4, 12, 36, 108, 324, 972, 76 function evaluations yielded $x_1 = 0.0000000$, $x_2 = 1.0000001$, $x_3 = 1.9999999$, $x_4 = -1.0000003$. With the reduction scheme, only active constraints were considered at the later stages of the process. A listing of the main program, the subroutine FUNCT and input data is shown in Fig. 10. As in the Beale problem, statements were added (in subroutine FUNCT) to allow printing of the constraints at the extrapolated solutions. Fig. 11 shows a printout of the input data. Fig. 12 shows the final solution of the problem.

Example 5: A microwave circuit example

The design of a three-section 100-percent relative bandwidth 10:1 transmission-line transformer [10] is considered. In this case, we let the error functions e_i be the modulus of the reflection coefficient sampled at the 11 normalized frequencies (w.r.t. 1 GHz)

{0.5, 0.6, 0.7, 0.77, 0.9, 1.0, 1.1, 1.23, 1.3, 1.4, 1.5}.

Gradient vectors with respect to section lengths and characteristic impedances

```
      PROGRAM TST (INPUT,OUTPUT,TAPE5=INPUT,TAPE6=OUTPUT)
C     MAIN PROGRAM OF EXAMPLE 4
C
      DIMENSION X(4),G(4),H(10),W(16),EPS(4),XB(4),XE(4,6,4)
      COMMON/USER/IGRAD
      N=4
      NR=4
      M=1
      IGK=1
      IK=6
      DO 1 IH=1,IK
      IGRAD=1
      CALL FLOPT2(N,M,IGK,X,G,H,W,EPS,XE,IH,IK,FACTOR,XB,IFINIS,NR)
      M=0
      IGRAD=0
      CALL FUNCT(XB,G,U)
      IF(IFINIS.EQ.N) CALL EXIT
1     CONTINUE
      STOP
      END


      SUBROUTINE FUNCT(X,G,U)
C     ROSEN-SUZUKI FUNCTION
C
      DIMENSION X(4),G(4),C(3),GF(4),GC(4,3),ER(4),GE(4,4),ES(4)
      COMMON/USER/IGRAD
      COMMON/WY3/NA,JD(150)
      DATA ALFA/10.0/
      N=4
      B=X(1)*X(1)
      R=X(2)*X(2)
      D=X(3)*X(3)
      E=X(4)*X(4)
      BB=X(1)+X(1)
      RR=X(2)+X(2)
      DD=X(3)+X(3)
      EE=X(4)+X(4)
      F=B+R+D+D+E-5.*(X(1)+X(2))-21.*X(3)+7.*X(4)
      IF(IGRAD.EQ.0) GO TO 5
      GF(1)=BB-5.
      GF(2)=RR-5.
      GF(3)=DD+DD-21.
      GF(4)=EE+7.
      DO 9 I=1,NA
      K=JD(I)
      GO TO (1,2,3,4), K
1     C(1)=-B-R-D-E-X(1)+X(2)-X(3)+X(4)+8.
      ER(1)=F-ALFA*C(1)
      GC(1,1)=-BB-1.
      GC(2,1)=-RR+1.
      GC(3,1)=-DD-1.
      GC(4,1)=-EE+1.
      DO 11 J=1,4
      GE(J,1)=GF(J)-ALFA*GC(J,1)
11    CONTINUE
      GO TO 9
2     C(2)=-B-R-R-D-E-E+X(1)+X(4)+10.
```

```
      FR(2)=F-ALFA*C(2)
      GC(1,2)=GC(1,1)+2.
      GC(2,2)=-RR-RR
      GC(3,2)=GC(3,1)+1.
      GC(4,2)=-FF-FF+1.
      DO 22 J=1,4
      GF(J,2)=GF(J)-ALFA*GC(J,2)
22    CONTINUE
      GO TO 9
3     C(3)=-B-B-R-D-BB+X(2)+X(4)+5.
      FR(3)=F-ALFA*C(3)
      GC(1,3)=GC(1,1)+GC(1,1)
      GC(2,3)=GC(2,1)
      GC(3,3)=GC(3,1)+1.
      GC(4,3)=1.
      DO 33 J=1,4
      GF(J,3)=GF(J)-ALFA*GC(J,3)
33    CONTINUE
      GO TO 9
4     FR(4)=F
      DO 44 J=1,4
      GF(J,4)=GF(J)
44    CONTINUE
9     CONTINUE
      CALL LEASTP(N,U,G,FR,GF,FS)
      RETURN
5     C(1)=-B-R-D-F-X(1)+X(2)-X(3)+X(4)+8.
      C(2)=-B-R-R-D-F-F+X(1)+X(4)+10.
      C(3)=-B-B-R-D-BB+X(2)+X(4)+5.
      PRINT 6,(X(I),I=1,4)
6     FORMAT(//1H0,*SOLUTION*,7X,4E16.8)
      PRINT 7,F
7     FORMAT(*0OBJECTIVE FUNCTION     *,E16.8)
      PRINT 8,(C(I),I=1,3)
8     FORMAT(*0CONSTRAINTS     *,3E16.8)
      RETURN
      END


C
C     INPUT DATA
C
  100    10     1     1
-100.0          0.0001          4.0          3.0
0.0             0.0             0.0          0.0
1.E-           81.E-           81.E-        81.E-        8
    1    3    2
```

Fig. 10 Main program and subroutine FUNCT for Example 4. Input data is also shown.

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

NUMBER OF INDEPENDENT VARIABLES........................N =   4

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

INTERMEDIATE PRINTOUT AT EVERY IPT ITERATIONS.........IPT =  10

STARTING VALUE FOR VECTOR X(I)...................X( 1) =  0.
                                                 X( 2) =  0.
                                                 X( 3) =  0.
                                                 X( 4) =  0.

TEST QUANTITIES TO BE USED.....................EPS( 1) =   .10000000E-07
                                               EPS( 2) =   .10000000E-07
                                               EPS( 3) =   .10000000E-07
                                               EPS( 4) =   .10000000E-07

ESTIMATE OF LOWER BOUND OF FUNCTION TO BE MINIMIZED..EST =  -.10000000E+03

HIGHEST ORDER OF ESTIMATES USED IN EXTRAPOLATION..JORDER =   3

MULTIPLIER TOLERANCE FOR DROPPING FUNCTIONS..........ETA =   .10000000E-03
```

Fig. 11   Input data for the Rosen-Suzuki problem.

```
ESTIMATES OF THE MINIMAX SOLUTION BY EXTRAPOLATION
--------------------------------------------------

ORDER  1
X( 1) =   -.99715414E-06
X( 2) =    .99999870E+00
X( 3) =    .19999970E+01
X( 4) =   -.99999632E+00
ORDER  2
X( 1) =    .58563435E-07
X( 2) =    .99999962E+00
X( 3) =    .20000000E+01
X( 4) =   -.10000001E+01
ORDER  3
X( 1) =    .14318896E-07
X( 2) =    .10000001E+01
X( 3) =    .19999999E+01
X( 4) =   -.10000003E+01
CORRESPONDING NORMALIZED ERRORS
     EM =   -.43999997E+02
EN( 1) =   1.00000000
EN( 3) =   1.00000009
EN( 4) =   1.00000007


SOLUTION              .14318896E-07   .10000001E+01   .19999999E+01   -.10000003E+01
OBJECTIVE FUNCTION        -.44000000E+02
CONSTRAINTS          -.30339277E-06   .99999878E+00   .91383583E-07
```

Fig. 12   Final results for the Rosen-Suzuki problem.

are obtained using the adjoint network method. Using 3rd order extrapolation and the reduction scheme with p = 8, 48, 288, 1728, we get a reflection coefficient magnitude of 0.19729 (optimal to 5 figures). The effort required is summarized in Table 1. A total of 557 network analyses were required, which was about 30% less than what would be required if the reduction scheme was not used. A listing of the main program, subroutine FUNCT and the input data is shown in Fig. 13. Note that the sample points are read from the main program and passed to the subroutine FUNCT via a COMMON block named USER. At the end of each optimization, the responses of the transformer at the local solution and the extrapolated solution are printed. In subroutine FUNCT, the error functions and their gradients are obtained from the subroutine NET which defines the reflection coefficient of the transformer. Fig. 14 shows the input data for this example. Fig. 15 shows the parameter values and error functions at the solution for p = 1728. A final estimate of the minimax solution and the corresponding errors are shown in Fig. 16. In Fig. 17, the 2nd column gives the modulus of the reflection coefficient at the solution for p = 1728, while the 3rd column gives that of the extrapolated minimax solution. Only the crucial frequency points are used, which appear in column 1.

VII. COMMENTS

The package is so organised that pertinent information of the optimization process can be obtained from the argument list of the subroutine FLOPT2. This allows the user to do some useful things in the main program, expecially when using extrapolation. Some suggestions are:

(i) In using the extrapolation procedure, we usually do not know how

Parameter p	Function evaluations	x	Number of error functions	=	Number of network analyses
8	27		11		297
48	20		8		160
288	14		4		56
1728	11		4		44
Total	72			total	557

Table 1 Computational effort for the transformer problem.

```
      PROGRAM TST (INPUT,OUTPUT,TAPE5=INPUT,TAPE6=OUTPUT)
C     MAIN PROGRAM OF EXAMPLE 5
C
      DIMENSION X(6),G(6),H(21),W(24),EPS(6),XB(6),XE(6,5,4)
      DIMENSION GRAD(6)
      COMMON/USER/WN(11)
      COMMON/WY3/NA,JD(150)
      N=6
      NR=11
      READ 1,(WN(I),I=1,NR)
1     FORMAT(8F10.0)
      M=1
      IGK=0
      IK=5
      DO 2 IH=1,IK
      CALL FLOPT2(N,M,IGK,X,G,H,W,EPS,XE,IH,IK,FACTOR,XB,IFINIS,NR)
      PRINT 22
22    FORMAT(1H1,8X,*RESPONSES OF THE TRANSFORMER*)
      PRINT 24
24    FORMAT(1H0,11X,*FREQUENCY*,8X,*REFLECTION COEF.*,10X,*BEST*)
      DO 3 I=1,NA
      K=JD(I)
      S=WN(K)
      CALL NET(G,S,ARHO,ATNG,GRAD,0)
      CALL NET(XB,S,ARHO,ATNB,GRAD,0)
      PRINT 26,S,ATNG,ATNB
26    FORMAT(1H0,3F20.8)
3     CONTINUE
      IF(IFINIS.EQ.N) CALL EXIT
      M=0
2     CONTINUE
      STOP
      END


      SUBROUTINE FUNCT(X,G,U)
C     A MICROWAVE CIRCUIT EXAMPLE
C
      DIMENSION X(6),G(6),FR(11),GE(6,11),ES(11)
      DIMENSION GRAD(6)
      COMMON/USER/WN(11)
      COMMON/WY3/NA,JD(150)
      N=6
      DO 1 I=1,NA
      K=JD(I)
      CALL NET(X,WN(K),ARHO,ATN,GRAD,1)
      FR(K)=ATN
      DO 2 J=1,N
      GE(J,K)=GRAD(J)
2     CONTINUE
1     CONTINUE
      CALL LEASTP(N,U,G,FR,GE,ES)
      RETURN
      END


C
C     INPUT DATA
```

```
C
.5          .6          .7          .77         .9          1.0         1.1         1.23
1.3         1.4         1.5
  100     10      1     1
0.0                 0.0005              8.0                 6.0
0.3                 1.5                 1.2                 3.0                 0.8
6.0
1.E-                81.E-               81.E-               81.E-               81.E-               8
1.E-                8
     1     3     2
```

Fig. 13 Main program and subroutine FUNCT for the transformer example. Input data is also shown.

INPUT DATA

```
NUMBER OF INDEPENDENT VARIABLES..........................N =   6

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

INTERMEDIATE PRINTOUT AT EVERY IPT ITERATIONS........IPT =  10

STARTING VALUE FOR VECTOR X(I)....................X( 1) =   .80000000E+00
                                                  X( 2) =   .15000000E+01
                                                  X( 3) =   .12000000E+01
                                                  X( 4) =   .30000000E+01
                                                  X( 5) =   .80000000E+00
                                                  X( 6) =   .60000000E+01

TEST QUANTITIES TO BE USED.......................EPS( 1) =   .10000000E-07
                                                 EPS( 2) =   .10000000E-07
                                                 EPS( 3) =   .10000000E-07
                                                 EPS( 4) =   .10000000E-07
                                                 EPS( 5) =   .10000000E-07
                                                 EPS( 6) =   .10000000E-07

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

HIGHEST ORDER OF ESTIMATES USED IN EXTRAPOLATION..JORDER =   3

MULTIPLIER TOLERANCE FOR DROPPING FUNCTIONS..........ETA =   .50000000E-03
```

Fig. 14 Input data for the transformer example.

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

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

      U =   .19744011E+00

     EM =   .19732020E+00
 EN( 1) =   .10000000E+01
 EN( 4) =   .99996250E+00
 EN( 8) =   .99969126E+00
 EN(11) =   .99936393E+00

  X( 1) =   .99995486E+00      G( 1) =  -.66649713E-10
  X( 2) =   .16347092E+01      G( 2) =  -.19447769E-10
  X( 3) =   .99999743E+00      G( 3) =  -.19477199E-09
  X( 4) =   .31622777E+01      G( 4) =  -.50507778E-12
  X( 5) =   .99995486E+00      G( 5) =  -.66084774E-10
  X( 6) =   .61172959E+01      G( 6) =   .44413492E-11

     NUMBER OF FUNCTION EVALUATIONS =   11
           VALUE OF THE PARAMETER P =   .17280000E+04
          EXECUTION TIME IN SECONDS =   .294
```

Fig. 15 Results for the transformer example.

```
ESTIMATES OF THE MINIMAX SOLUTION BY EXTRAPOLATION
--------------------------------------------------

ORDER  3
X( 1) =    .10000000E+01
X( 2) =    .16347073E+01
X( 3) =    .99999999E+00
X( 4) =    .31622777E+01
X( 5) =    .10000000E+01
X( 6) =    .61173032E+01
CORRESPONDING NORMALIZED ERRORS
     EM =    .19729074E+00
EN( 1) =    .99999681
EN( 4) =    .99999994
EN( 8) =   1.00000000
EN(11) =    .99999875
```

Fig. 16. Final estimate of the minimax solution.

```
RESPONSES OF THE TRANSFORMER
  FREQUENCY          REFLECTION COEF.          BEST
   .50000000            .19732020            .19729051
   .77000000            .19731280            .19729073
  1.23000000            .19725928            .19729074
  1.50000000            .19719469            .19729049
```

Fig. 17 Responses of the 3-section transformer.

many cycles of optimization are required and the parameter IK may be set too large. The value of the parameter IFINIS may be used as a stopping criterion. When the accuracy in each element of XB (estimates of the minimax solution) is less than one hundred times of the accuracy in each element of X, the value of IFINIS becomes n. A statement as

```
IF (IFINIS.EQ.N) CALL EXIT
```

put inside the DO loop will serve the purpose. See Examples 2-5.

(ii) Responses (function values or constraints) at the end of each optimization or at the estimated minimax solution may be evaluated in the main program by calling subroutine FUNCT. See Examples 2, 4, 5. When extrapolation is used, but not the least pth formulation (as in the Fiacco-McCormick method), the sequence of the controlling parameter r can be updated in the main program. See Example 2.

(iii) System failure or time-limit may sometimes occur before the execution of the program is complete. Most of the information will be lost if it has not been saved. As a precaution, the user may at the end of each optimization cycle save the value of the array XE and the starting value of the next optimization cycle (which is stored in the array X) as punched output. Should restarting be necessary, the user simply reads in the value of the array XE obtained before the interruption, the starting value of $\underset{\sim}{x}$, some required input data and sets the value of IH to the appropriate cycle number. The process should then proceed as if nothing had happened.

Suppose time-limit occurred during execution of the fourth optimization cycle and we had saved relevant information of the previous three cycles. To restart the optimization process at the fourth cycle, the main program may contain the following statements:

```
          .
          .
          READ (5,2)     (XE(I,1,1), I = 1, N)
          READ (5,2)     (XE(I,2,1), I = 1, N)
          READ (5,2)     (XE(I,2,2), I = 1, N)
          READ (5,2)     (XE(I,3,1), I = 1, N)
          READ (5,2)     (XE(I,3,2), I = 1, N)
          READ (5,2)     (XE(I,3,3), I = 1, N)
2         FORMAT (5E16.8)
          M = 1
          IGK = 0
          DO 1   IH = 4, IK
          CALL FLOPT2 (N, M, IGK, X, G, H, W, EPS, XE, IH, IK, FACTOR,
     1    XB, IFINIS, NR)
          IF  (IFINIS .EQ. N) CALL EXIT
          M = 0
1         CONTINUE
          .
          .
```

The purpose of each statement should be self-explanatory.

VIII. CONCLUSIONS

A package of subroutines, called FLOPT2, for solving least pth optimization problems has been presented. Its features, which include Fletcher's quasi-Newton subroutine, a least pth objective formulation subroutine, an extrapolation procedure and a scheme for dropping inactive functions, make it capable of solving unconstrained problems, constrained problems or nonlinear minimax approximation problems. Several examples have been presented to illustrate the versatility of the program. The mathematical background for the extrapolation procedure to minimax solutions (or the p-algorithm) has been omitted, but is readily available [2], [6], [11].

REFERENCES

[1] R. Fletcher, "FORTRAN subroutines for minimization by quasi-Newton methods", Atomic Energy Research Establishment, Harwell, Berkshire, England, Report AERE-R7125, 1972.

[2] J.W. Bandler, C. Charalambous, J.H.K. Chen and W.Y. Chu, "New results in the least pth approach to minimax design", Faculty of Engineering, McMaster University, Hamilton, Canada, Mar. 1975, Report SOC-80.

[3] A.V. Fiacco and G.P. McCormick, Nonlinear Programming: Sequential Unconstrained Minimization Techniques. New York: Wiley, 1968.

[4] J.W. Bandler and C. Charalambous, "Nonlinear programming using minimax techniques", J. Optimization Theory and Applications, vol. 13, June 1974, pp. 607-619.

[5] J.W. Bandler and W.Y. Chu, "Function optimization package version FLOPT1", Faculty of Engineering, McMaster University, Hamilton, Canada, Aug. 1973, Report SOC-17.

[6] W.Y. Chu, "Extrapolation in least pth approximation and nonlinear programming", Faculty of Engineering, McMaster University, Hamilton, Canada, Dec. 1974, Report SOC-71.

[7] H.H. Rosenbrock, "An automatic method for finding the greatest or least value of a function", Comp. J., vol. 3, Oct. 1960,pp. 175-184.

[8] J. Kowalik and M.R. Osborne, Methods for Unconstrained Optimization Problems. New York: Elsevier, 1968.

[9] C. Charalambous and J.W. Bandler, "Nonlinear minimax optimization as a sequence of least pth optimization with finite values of p", Faculty of Engineering, McMaster University, Hamilton, Canada, June 1973, Report SOC-3 (Revised Mar. 1974).

[10] J.W. Bandler, T.V. Srinivasan and C. Charalambous, "Minimax optimization of networks by grazor search", IEEE Trans. Microwave Theory Tech., vol. MTT-20, Sept. 1972, pp. 596-604.

[11] J.W. Bandler and W.Y. Chu, "Nonlinear programming using least pth optimization with extrapolation", Faculty of Engineering, McMaster University, Hamilton, Canada, Mar. 1975, Report SOC-78.

FORTRAN Listing for FLOPT2

```
      SUBROUTINE FLOPT2 (N,M,IGK,X,G,H,W,EPS,XE,IH,IK,FACTOR,XB,IFINIS,N
     1R)
C
C     THIS SUBROUTINE CO-ORDINATES INPUT, OUTPUT, MINIMIZATION AND DROP-
C     PING FUNCTIONS
C
      DIMENSION X(1), G(1), H(1), W(1), EPS(1), XE(N,IK,1), XB(1)
      COMMON /WY1/ IFN,KO
      COMMON /WY2/ P,JV,V(150),EM,EN(150)
      COMMON /WY3/ NA,JD(150)
      IFINIS=0
      JV=0
      IF (M.EQ.0) GO TO 3
      READ (5,23) MAX,IPT,ID,IREDU
      READ (5,24) EST,ETA,PO,FACTOR
      READ (5,24) (X(I),I=1,N)
      READ (5,24) (EPS(I),I=1,N)
      READ (5,23) IEX,JORDER,JPRINT
      MODE=1
      P=PO
      IF (ID.EQ.0) GO TO 1
      WRITE (6,33)
      WRITE (6,34) N
      WRITE (6,35) MAX
      WRITE (6,36) IPT
      WRITE (6,37) X(1)
      WRITE (6,38) (I,X(I),I=2,N)
      WRITE (6,39) EPS(1)
      WRITE (6,40) (I,EPS(I),I=2,N)
      WRITE (6,41) EST
      IF (IEX.EQ.1) WRITE (6,46) JORDER
      IF (IREDU.EQ.1) WRITE (6,25) ETA
    1 CONTINUE
      NA=NR
      DO 2 I=1,NA
      JD(I)=I
    2 CONTINUE
      IF (IGK.EQ.1) CALL GRDCHK (N,X,G,W)
    3 P=PO
      IF (IPT.EQ.0) GO TO 4
      WRITE (6,42)
      WRITE (6,43)
      WRITE (6,44)
    4 CONTINUE
      CALL SECOND (T1)
      CALL QUASIN (N,X,U,G,H,W,EST,EPS,MODE,MAX,IPT,IEXIT)
      CALL SECOND (T2)
      T=T2-T1
      CALL RESULT (N,X,U,G,NR)
      IF (NR.GT.1) WRITE (6,26) P
      WRITE (6,45) T
      MODE=3
      JV=1
      IF (IEX.EQ.0) GO TO 13
      DO 5 I=1,N
      G(I)=X(I)
    5 CONTINUE
      CALL EXTRAP (N,X,XE,IH,IK,FACTOR,XB,JORDER)
```

```
      J1=JORDER+1
      IHH=IH-1
      IF (JPRINT.EQ.0 .OR.IH.EQ.1) GO TO 9
      IJ=J1
      IF (IH.LE.J1) IJ=IH
      WRITE (6,27)
      DO 7 L=2,IJ
      L1=L-1
      WRITE (6,28) L1
      DO 6 J=1,N
      WRITE (6,29) J,XE(J,IH,L)
6     CONTINUE
7     CONTINUE
      IF (NR.LE.1) RETURN
      CALL FUNCT (XB,G,U)
      WRITE (6,19)
      WRITE (6,20) EM
      DO 8 I=1,NA
      J=JD(I)
      WRITE (6,21) J,EN(J)
8     CONTINUE
9     IF (IH.LT.3) GO TO 13
      IF (IH.GT.J1) GO TO 11
      DO 10 I=1,N
      IF (ABS(XE(I,IH,IH)-XE(I,IHH,IHH)).LT.100.*EPS(I)) IFINIS=IFINIS+1
10    CONTINUE
      GO TO 13
11    DO 12 I=1,N
      IF (ABS(XE(I,IH,J1)-XE(I,IHH,J1)).LT.100.*EPS(I)) IFINIS=IFINIS+1
12    CONTINUE
13    CONTINUE
      IF (NR.LE.1.OR.IH.EQ.IK) RETURN
      PO=P*FACTOR
      IF (IH.EQ.1) GO TO 14
      WRITE (6,30)
      GO TO 15
14    P=PO/FACTOR
      WRITE (6,31)
15    CALL FUNCT (X,G,U)
      WRITE (6,20) EM
      K=0
      DO 18 I=1,NA
      J=JD(I)
      IF (JPRINT.LT.2) GO TO 16
      WRITE (6,22) J,V(J),J,EN(J)
16    IF (IREDU.EQ.0) GO TO 17
      IF (V(J).LT.ETA) GO TO 18
17    K=K+1
      JD(K)=J
18    CONTINUE
      NA=K
      IF (IH.EQ.1) PO=P*FACTOR
      WRITE (6,32) NA
      RETURN
C
C
C
19    FORMAT (1H0,*CORRESPONDING NORMALIZED ERRORS*)
```

```
20    FORMAT (1H0,4X,*EM =*,E16.8)
21    FORMAT (1H0,*EN(*,I2,*) =*,F12.8)
22    FORMAT (*0MU(*,I2,*) =*,F12.8,8X,*EN(*,I2,*) =*,F12.8)
23    FORMAT (4I5)
24    FORMAT (5E16.8)
25    FORMAT (1H0/* MULTIPLIER TOLERANCE FOR DROPPING FUNCTIONS*,10(*.*)
     1,*ETA =*,E16.8)
26    FORMAT (1H0,10X,*VALUE OF THE PARAMETER P =*,E16.8)
27    FORMAT (1H0/1H0,*ESTIMATES OF THE MINIMAX SOLUTION BY EXTRAPOLATIO
     1N*/1H ,50(*-*)/)
28    FORMAT (1H0,*ORDER*,I3)
29    FORMAT (1H0,*X(*,I2,*) =*,E16.8)
30    FORMAT (1H0/*0MULTIPLIERS AND NORMALIZED ERRORS FOR THE NEXT ESTIM
     1ATED*/*0LEAST PTH SOLUTION*)
31    FORMAT (1H0/*0MULTIPLIERS AND NORMALIZED ERRORS AT THE ABOVE SOLUT
     1ION*)
32    FORMAT (1H0/*0NUMBER OF ERROR FUNCTIONS FOR THE NEXT OPTIMIZATION
     1=*,I4)
33    FORMAT (1H1,*INPUT DATA*/1H ,10(*-*)//)
34    FORMAT (1H0,*NUMBER OF INDEPENDENT VARIABLES*,24(*.*),*N =*,I4/)
35    FORMAT (1H0,*MAXIMUM NUMBER OF ALLOWABLE ITERATIONS*,15(*.*),*MAX
     1=*,I4/)
36    FORMAT (1H0,*INTERMEDIATE PRINTOUT AT EVERY IPT ITERATIONS*,8(*.*)
     1,*IPT =*,I4/)
37    FORMAT (1H0,*STARTING VALUE FOR VECTOR X(I)*,21(*.*),*X( 1) =*,E16
     1.8)
38    FORMAT (1H0,51X,*X(*,I2,*) =*,E16.8)
39    FORMAT (1H0,/,1H ,*TEST QUANTITIES TO BE USED*,23(*.*),*EPS( 1) =*
     1,E16.8)
40    FORMAT (1H0,49X,*EPS(*,I2,*) =*,E16.8)
41    FORMAT (1H0,/1H ,*ESTIMATE OF LOWER BOUND OF FUNCTION TO BE MINIMI
     1ZED*,2(*.*),*EST =*,E16.8)
42    FORMAT (1H1)
43    FORMAT (1H0,*OPTIMIZATION BY FLETCHER METHOD*/1H ,31(*-*)/)
44    FORMAT (1H0,*ITER.*,2X,*FUNCT.*,6X,*OBJECTIVE*,7X,*VARIABLE*,9X,*G
     1RADIENT*/1H0,1X,*NO.*,3X,*EVALU.*,6X,*FUNCTION*,7X,*VECTOR X(I)*,6
     2X,*VECTOR G(I)*,/)
45    FORMAT (1H0,9X,*EXECUTION TIME IN SECONDS =*,F7.3)
46    FORMAT (1H0/1H ,*HIGHEST ORDER OF ESTIMATES USED IN EXTRAPOLATION*
     1,*..JORDER =*,I4)
      END
```

```
      SUBROUTINE LEASTP (N,U,G,ER,GE,ES)
C
C     THIS SUBROUTINE PERFORMS LEAST PTH OBJECTIVE FORMULATION
C
      DIMENSION G(1), ER(1), GE(N,1), ES(1)
      COMMON /WY2/ P,JV,V(150),EM,EN(150)
      COMMON /WY3/ NA,JD(150)
      J=JD(1)
      EM=ER(J)
      DO 1 I=2,NA
      J=JD(I)
      EM=AMAX1(EM,ER(J))
    1 CONTINUE
      IF (EM.NE.0.) GO TO 3
      DO 2 I=1,NA
      J=JD(I)
      ER(J)=ER(J)-1.E-10
    2 CONTINUE
      EM=EM-1.E-10
    3 Q=SIGN(P,EM)
      S1=0.
      DO 5 I=1,NA
      J=JD(I)
      IF (EM.LT.0.) GO TO 4
      IF (ER(J).LE.0.) GO TO 5
    4 EN(J)=ER(J)/EM
      ES(J)=EN(J)**Q
      S1=S1+ES(J)
    5 CONTINUE
      ST=S1**(1./Q)
      U=EM*ST
      IF (JV.EQ.1) GO TO 9
      ST=ST/S1
      DO 8 I=1,N
      S2=0.
      DO 7 J=1,NA
      K=JD(J)
      IF (EM.LT.0.) GO TO 6
      IF (ER(K).LE.0.) GO TO 7
    6 S2=S2+ES(K)*GE(I,K)/EN(K)
      CONTINUE
      G(I)=ST*S2
    8 CONTINUE
      RETURN
    9 DO 11 J=1,NA
      K=JD(J)
      V(K)=0.
      IF (EM.LT.0.) GO TO 10
      IF (ER(K).LE.0.) GO TO 11
   10 V(K)=ES(K)/S1
   11 CONTINUE
      RETURN
      END
```

```
      SUBROUTINE GRDCHK (N,X,G,W)
C
C     THIS SUBROUTINE PERFORMS GRADIENT CHECK
C
      DIMENSION X(N), G(N), W(N)
      IC=0
      CALL FUNCT (X,G,U)
      WRITE (6,3)
      WRITE (6,4)
      DO 1 I=1,N
      Z=X(I)
      DX=1.E-6*X(I)
      IF (ABS(DX).LT.1.E-10) DX=1.E-10
      X(I)=Z+DX
      CALL FUNCT (X,W,U2)
      X(I)=Z-DX
      CALL FUNCT (X,W,U1)
      Y=0.5*(U2-U1)/DX
      X(I)=Z
      IF (ABS(Y).LT.1.E-20) Y=1.E-20
      IF (ABS(G(I)).LT.1.E-20) G(I)=1.E-20
      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
C
    3 FORMAT (1H1,/1H ,*GRADIENT CHECK AT STARTING POINT*/1H ,32(*-*))
    4 FORMAT (///,1H0,5X,*ANALYTICAL GRADIENTS*,5X,*NUMERICAL GRADIENTS*
     1,5X,*PERCENTAGE ERROR*)
    5 FORMAT (1H0,6X,E16.8,8X,E16.8,7X,F16.8)
    6 FORMAT (1H0,///1H ,*GRADIENTS ARE O.K.*)
    7 FORMAT (1H0,///1H ,*YOUR PROGRAM HAS BEEN TERMINATED BECAUSE GRADI
     1ENTS ARE INCORRECT*/1H0,*PLEASE CHECK IT AGAIN*)
      END
```

```
      SUBROUTINE QUASIN (N,X,U,G,H,W,EST,EPS,MODE,MAX,IPT,IEXIT)
C
C     THIS SUBROUTINE IS THE 1972 VERSION OF FLETCHER METHOD
C
      DIMENSION X(1), G(1), H(1), W(1), EPS(1)
      COMMON /WY1/ IFN,KO
      KO=0
      NP=N+1
      N1=N-1
      NN=N*NP/2
      IS=N
      IU=N
      IV=N+N
      IB=IV+N
      IEXIT=0
      IF (MODE.EQ.3) GO TO 6
      IF (MODE.EQ.2) GO TO 3
      IJ=NN+1
      DO 2 I=1,N
      DO 1 J=1,I
      IJ=IJ-1
      H(IJ)=0.
1     CONTINUE
      H(IJ)=1.
2     CONTINUE
      GO TO 6
3     CONTINUE
      IJ=1
      DO 5 I=2,N
      Z=H(IJ)
      IF (Z.LE.0.) RETURN
      IJ=IJ+1
      I1=IJ
      DO 5 J=I,N
      ZZ=H(IJ)
      H(IJ)=H(IJ)/Z
      JK=IJ
      IK=I1
      DO 4 K=I,J
      JK=JK+NP-K
      H(JK)=H(JK)-H(IK)*ZZ
      IK=IK+1
4     CONTINUE
5     IJ=IJ+1
      IF (H(IJ).LE.0.) RETURN
6     CONTINUE
      IJ=NP
      DMIN=H(1)
      DO 7 I=2,N
      IF (H(IJ).GE.DMIN) GO TO 7
      DMIN=H(IJ)
7     IJ=IJ+NP-I
      IF (DMIN.LE.0.) RETURN
      Z=EST
      ITN=0
      CALL FUNCT (X,G,U)
      IFN=1
```

```
      DF=U-EST
      IF (DF.LE.0.0) DF=1.0
 8    CONTINUE
      IF (IPT.EQ.0) GO TO 9
      IF (MOD(ITN,IPT).NE.0) GO TO 9
      PRINT 44, ITN,IFN,U,((X(I),G(I)),I=1,N)
 9    CONTINUE
      ITN=ITN+1
      W(1)=-G(1)
      DO 11 I=2,N
      IJ=I
      I1=I-1
      Z=-G(I)
      DO 10 J=1,I1
      Z=Z-H(IJ)*W(J)
      IJ=IJ+N-J
 10   CONTINUE
      W(I)=Z
 11   CONTINUE
      W(IS+N)=W(N)/H(NN)
      IJ=NN
      DO 13 I=1,N1
      IJ=IJ-1
      Z=0.
      DO 12 J=1,I
      Z=Z+H(IJ)*W(IS+NP-J)
      IJ=IJ-1
 12   CONTINUE
      W(IS+N-I)=W(N-I)/H(IJ)-Z
 13   CONTINUE
      GS=0.
      DO 14 I=1,N
      GS=GS+W(IS+I)*G(I)
 14   CONTINUE
      IEXIT=2
      IF (GS.GE.0.) GO TO 36
      GS0=GS
      ALPHA=-2.*DF/GS
      IF (ALPHA.GT.1.) ALPHA=1.
      DF=U
      TOT=0.
 15   CONTINUE
      IEXIT=3
      IF (ITN.EQ.MAX) GO TO 36
      ICON=0
      IEXIT=1
      DO 16 I=1,N
      Z=ALPHA*W(IS+I)
      IF (ABS(Z).GE.EPS(I)) ICON=1
      X(I)=X(I)+Z
 16   CONTINUE
      CALL FUNCT (X,W,FY)
      IFN=IFN+1
      GYS=0.
      DO 17 I=1,N
      GYS=GYS+W(I)*W(IS+I)
 17   CONTINUE
      IF (FY.GE.U) GO TO 18
```

```
      IF (ABS(GYS/GS0).LE..9) GO TO 20
      IF (GYS.GT.0.) GO TO 18
      TOT=TOT+ALPHA
      Z=10.
      IF (GS.LT.GYS) Z=GYS/(GS-GYS)
      IF (Z.GT.10.) Z=10.
      ALPHA=ALPHA*Z
      U=FY
      GS=GYS
      GO TO 15
18    CONTINUE
      DO 19 I=1,N
      X(I)=X(I)-ALPHA*W(IS+I)
19    CONTINUE
      IF (ICON.EQ.0) GO TO 36
      Z=3.*(U-FY)/ALPHA+GYS+GS
      ZZ=SQRT(Z*Z-GS*GYS)
      GZ=GYS+ZZ
      Z=1.-(GZ-Z)/(ZZ+GZ-GS)
      ALPHA=ALPHA*Z
      GO TO 15
20    CONTINUE
      ALPHA=TOT+ALPHA
      U=FY
      IF (ICON.EQ.0) GO TO 34
      DF=DF-U
      DGS=GYS-GS0
      LINK=1
      IF (DGS+ALPHA*GS0.GT.0.) GO TO 22
      DO 21 I=1,N
      W(IU+I)=W(I)-G(I)
21    CONTINUE
      SIG=1./(ALPHA*DGS)
      GO TO 29
22    CONTINUE
      ZZ=ALPHA/(DGS-ALPHA*GS0)
      Z=DGS*ZZ-1.
      DO 23 I=1,N
      W(IU+I)=Z*G(I)+W(I)
23    CONTINUE
      SIG=1./(ZZ*DGS*DGS)
      GO TO 29
24    CONTINUE
      LINK=2
      DO 25 I=1,N
      W(IU+I)=G(I)
25    CONTINUE
      IF (DGS+ALPHA*GS0.GT.0.) GO TO 26
      SIG=1./GS0
      GO TO 29
26    CONTINUE
      SIG=-ZZ
      GO TO 29
27    CONTINUE
      DO 28 I=1,N
      G(I)=W(I)
28    CONTINUE
      GO TO 8
```

```
29    CONTINUE
      W(IV+1)=W(IU+1)
      DO 31 I=2,N
      IJ=I
      I1=I-1
      Z=W(IU+I)
      DO 30 J=1,I1
      Z=Z-H(IJ)*W(IV+J)
      IJ=IJ+N-J
30    CONTINUE
      W(IV+I)=Z
31    CONTINUE
      IJ=1
      DO 32 I=1,N
      IVI=IV+I
      IBI=IB+I
      Z=H(IJ)+SIG*W(IVI)*W(IVI)
      IF (Z.LE.0.) Z=DMIN
      IF (Z.LT.DMIN) DMIN=Z
      H(IJ)=Z
      W(IBI)=W(IVI)*SIG/Z
      SIG=SIG-W(IBI)*W(IBI)*Z
      IJ=IJ+NP-I
32    CONTINUE
      IJ=1
      DO 33 I=1,N1
      IJ=IJ+1
      I1=I+1
      DO 33 J=I1,N
      W(IU+J)=W(IU+J)-H(IJ)*W(IV+I)
      H(IJ)=H(IJ)+W(IB+I)*W(IU+J)
33    IJ=IJ+1
      GO TO (24,27), LINK
34    CONTINUE
      DO 35 I=1,N
      G(I)=W(I)
35    CONTINUE
36    CONTINUE
      IF (IEXIT.EQ.1) KO=1
      IF (IPT.EQ.0) GO TO 37
      PRINT 44, ITN,IFN,U,((X(I),G(I)),I=1,N)
37    IF (IEXIT.EQ.0) GO TO 38
      GO TO 39
38    PRINT 45, IEXIT
      GO TO 43
39    GO TO (40,41,42), IEXIT
40    PRINT 46, IEXIT
      GO TO 43
41    PRINT 47, IEXIT
      GO TO 43
42    PRINT 48, IEXIT
43    CONTINUE
      RETURN
C
C
44    FORMAT (1H ,I3,5X,I3,2X,E16.8,80(E16.8,2X,E16.8,/30X))
45    FORMAT (1H1,*IEXIT =*,I2/1H0,*THE ESTIMATE OF THE HESSIAN MATRIX I
     1S NOT POSITIVE DEFINITE*)
```

```
46     FORMAT (1H1,*IEXIT =*,I2/1H0,*CRITERION FOR OPTIMUM (CHANGE IN VEC
      1TOR X .LT. EPS) HAS BEEN SATISFIED*)
47     FORMAT (1H1,*IEXIT =*,I2/1H0,*EPS CHOSEN IS TOO SMALL*)
48     FORMAT (1H1,*IEXIT =*,I2/1H0,*MAXIMUM NUMBER OF ALLOWABLE ITERATIO
      1NS HAS BEEN REACHED*)
       END
```

```
      SUBROUTINE EXTRAP (N,X,XE,IH,IK,FACTOR,XB,JORDER)
C
C     THIS SUBROUTINE PERFORMS EXTRAPOLATION
C
      DIMENSION X(1), XE(N,IK,1), XB(1)
      I=IH
      II=I+1
      DO 1 J=1,N
      XE(J,I,1)=X(J)
1     CONTINUE
      IF (I.LT.2) GO TO 11
      IF (I.GT.JORDER) GO TO 2
      IJ=I
      GO TO 3
2     IJ=JORDER+1
3     DO 5 L=2,IJ
      LL=L-1
      S=FACTOR**LL
      DO 4 J=1,N
      XE(J,I,L)=(S*XE(J,I,LL)-XE(J,I-1,LL))/(S-1.0)
4     CONTINUE
5     CONTINUE
      DO 6 J=1,N
      XB(J)=XE(J,I,IJ)
6     CONTINUE
      DO 7 J=1,N
      XE(J,II,IJ)=XE(J,I,IJ)
7     CONTINUE
      DO 9 K=2,IJ
      L=IJ+1-K
      SS=FACTOR**L
      DO 8 J=1,N
      XE(J,II,L)=((SS-1.0)*XE(J,II,L+1)+XE(J,I,L))/SS
8     CONTINUE
9     CONTINUE
      DO 10 J=1,N
      X(J)=XE(J,II,1)
10    CONTINUE
      RETURN
11    DO 12 J=1,N
      XB(J)=XE(J,I,1)
12    CONTINUE
      RETURN
      END
```

```
      SUBROUTINE RESULT (N,X,U,G,NR)
C
C     THIS SUBROUTINE PRINTS RESULTS
C
      COMMON /WY1/ IFN,KO
      COMMON /WY2/ P,JV,V(150),EM,EN(150)
      COMMON /WY3/ NA,JD(150)
      DIMENSION X(N), G(N)
      IF (KO.EQ.0) GO TO 1
      WRITE (6,8)
      GO TO 2
1     WRITE (6,9)
2     CONTINUE
      WRITE (6,10) U
      IF (NR.LE.1) GO TO 4
      WRITE (6,5) EM
      DO 3 I=1,NA
      J=JD(I)
      WRITE (6,6) J,EN(J)
3     CONTINUE
      WRITE (6,7)
4     WRITE (6,11) (I,X(I),I,G(I),I=1,N)
      WRITE (6,12) IFN
      RETURN
C
C
C
5     FORMAT (1H0,4X,*EM =*,E16.8)
6     FORMAT (1H0,*EN(*,I2,*) =*,E16.8)
7     FORMAT (1H )
8     FORMAT (1H0/1H0,*OPTIMAL SOLUTION FOUND BY FLETCHER METHOD*/1H ,41
     1(*-*))
9     FORMAT (1H0/1H0,*RESULTS FOUND BY FLETCHER METHOD AT LAST ITERATIO
     1N*/1H ,50(*-*))
10    FORMAT (1H0,//6X,*U =*,E16.8/)
11    FORMAT (1H0,* X(*,I2,*) =*,E16.8,5X,*G(*,I2,*) =*,E16.8)
12    FORMAT (1H0,/1H ,4X,*NUMBER OF FUNCTION EVALUATIONS =*,I5)
      END
```