INTERNAL REPORTS IN

# SIMULATION, OPTIMIZATION AND CONTROL

No. SOC-15

## NONLINEAR PROGRAMMING PACKAGE FOR CONSTRAINED OPTIMIZATION VERSION FLNLP1

J.W. Bandler and W.Y. Chu

August 1973

FACULTY OF ENGINEERING
McMASTER UNIVERSITY
HAMILTON, ONTARIO, CANADA

# NONLINEAR PROGRAMMING PACKAGE FOR CONSTRAINED OPTIMIZATION: VERSION FLNLP1

J.W. Bandler and W.Y. Chu

Abstract The program developed solves nonlinear programming problems using a minimax approach with the efficient gradient method by Fletcher. It is available for batch-processing as well as on the time-sharing system INTERCOM of McMaster's CDC 6400.

## I. INTRODUCTION

The package FLNLP1 is a modified and more concise version of CONOPT[1] to be used for solving constrained optimization problems. The technique proposed by Bandler and Charalambous [2-4] is used to transform the constrained optimization problem into the minimization of an unconstrained objective function. Practical least pth approximation [5] is used to solve the resulting minimax problem. An equality constraint is treated as two inequality constraints, e.g., $h(x_1,x_2,...,x_n) = 0$ will be treated as $h(x_1,x_2,...,x_n) \geq 0$ and $-h(x_1,x_2,...,x_n) \geq 0$. The program is currently limited to 100 inequality constraints.

Three options are available for the package FLNLP1. The user can access the package for batch-processing, or he can use the package on the time-sharing system INTERCOM of the CDC 6400 computer. One of the options is a demonstration program for the package as used on INTERCOM. Except for this option, the user has to supply the main program and a subroutine called FUNCT which defines the actual objective function, the

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.

constraint functions and all first partial derivatives.

## II. USAGE

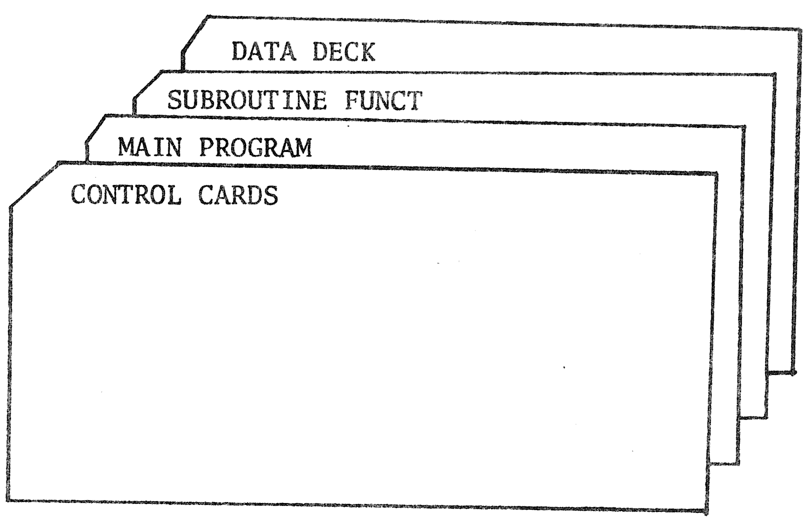
The three options of the package FLNLP1 are stored on permanent files; each has its own filename. They are called:

1. FLNLP1B [for batch-processing]
2. FLNLP1I [for INTERCOM]
3. FLNLP1D [for the demonstration program on INTERCOM]

Instructions for using the package are:

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, FLNLP1B, 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), EPS(N), G(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

NC = The number of inequality constraints

(c) Call subroutine FLNLP1 as follows:

CALL FLNLP1 (N, K, NC, X, G, H, EPS)

(d) Add STOP and END cards.

4. SUBROUTINE FUNCT

This subroutine defines the actual objective function and the constraints. A subroutine FMIMAX is then called which combines the

---

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

objective function and the constraints in a suitable manner to give the unconstrained objective function F and its gradients. The subroutine FUNCT should be written as follows:

```
SUBROUTINE FUNCT (X, F, G, U)
DIMENSION X(N), G(N), C(NC), GU(N), GC(N,NC),
                    A(NT), TT(NT)
```

where N = The number of independent variables (n)

NC = The number of inequality constraints ($n_c$)

NT = NC + 1

and all three numbers should be entered explicitly in the dimension statement.

Define the actual objective function

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

Define the inequality constraints

$$C(1) = c_1(x_1, x_2, \ldots, x_n)$$

$$C(2) = c_2(x_1, x_2, \ldots, x_n)$$

$$\vdots$$

$$C(NC) = c_{n_c}(x_1, x_2, \ldots, x_n)$$

Define the gradients of the objective function

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

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

$$\vdots$$

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

Define the gradients of the inequality constraints

$$GC(1, 1) = \frac{\partial c_1}{\partial x_1}$$

$$GC(2, 1) = \frac{\partial c_1}{\partial x_2}$$

$$\vdots$$

$$GC(N, 1) = \frac{\partial c_1}{\partial x_n}$$

$$\vdots$$

$$GC(N, 2) = \frac{\partial c_2}{\partial x_n}$$

$$\vdots$$

$$GC(N, NC) = \frac{\partial c_{n_c}}{\partial x_n}$$

Define the artificial unconstrained objective function by

```
CALL FMIMAX (N, NC, NT F, G, GU, C, GC, U, A, TT)
RETURN
END
```

If any other statements are necessary to define the actual objective function and the constraints, they may be added to this subroutine, e.g.,function U may be defined in another subprogram which may be called by subroutine FUNCT.

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 |
| A0 | Initial value of the parameter $\alpha$ used in formulating the unconstrained objective function [1,4] |
| EPSC | The margin by which constraints may be violated |

| | |
|---|---|
| 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

AO = 1.0

P = 1.E + 5

EPSC= 1.E - 5

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 | 5E16.8 | EST, AO, P, EPSC |
| 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, FLNLP1I, 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, for the INTERCOM option, one more array has to be dimensioned in the main program. The dimension statement should be DIMENSION X(N), EPS(N), G(N), H(K), XS(N) and the CALL statement for the package should be CALL FLNLP1 (N, K, NC, X, G, H, EPS, XS). After typing the main program and subroutine FUNCT, the user may leave the EDITOR mode by typing the commands

SAVE, filename

BYE

The edit file is saved and may be stored as a permanent file for future use.

3. To compile the program just written, the user should type the command

   FTN, I = filename.

4. To execute the program with the package FLNLP1, the necessary commands are

   CONNECT, INPUT, OUTPUT.

   XEQ.

   LOAD = LGO, ℓfn.†

   EXECUTE

---

†ℓfn stands for logical filename and it should be the same in both the ATTACH and LOAD commands.

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. Each value may be separated by a blank, a few blanks, a comma or by typing the RETURN key. If a line cannot accommodate a specific input string, go to 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, FLNLP1D, ID = *********.
CONNECT, INPUT, OUTPUT.
XEQ, ℓfn.
```

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 FLNLP1 on INTERCOM. However, for a long subroutine FUNCT, it is advisable to store the created program on permanent file. The user can then call back his program subsequently as needed.

In all 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. Execution time is usually faster if no intermediate printout is desired. Results for some of the problems solved using this package have been included in Appendix A and B. The example in Appendix B is that used on the demonstration program. Appendix C shows the general structure of the package.

## REFERENCES

[1] J.W. Bandler, J.H.K. Chen and V.K. Jha, "CONOPT - a package for solving nonlinear programming problems using a new (minimax) approach with efficient gradient methods", Department of Electrical Engineering, McMaster University, Hamilton, Ontario, Canada, 1973, internal report.

[2] J.W. Bandler and C. Charalambous, "A new approach to nonlinear programming", Proc. 5th Hawaii Int. Conf. on Systems Science (Honolulu, Jan. 1972), pp. 127-129.

[3] J.W. Bandler and C. Charalambous, "Nonlinear programming using minimax techniques", J. Optimization Theory and Applications, accepted for publication.

[4] C. Charalambous, "Nonlinear least pth approximation and nonlinear programming with applications in the design of networks and systems", Ph.D. Thesis, Department of Electrical Engineering, McMaster University, Hamilton, Ontario, Canada, 1973.

[5] J.W. Bandler and C. Charalambous, "Practical least pth optimization of networks", IEEE Trans. Microwave Theory Tech., vol. MTT-20, pp. 834-840, Dec. 1972.

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

ACKNOWLEDGEMENTS

Useful suggestions from P.C. Liu and J.H.K. Chen are acknowledged.

# APPENDIX A

To minimize

$$U = 9 - 8x_1 - 6x_2 - 4x_3 + 2x_1^2 + 2x_2^2 + x_3^2 + 2x_1x_2 + 2x_1x_3$$

subject to

$$x_1 \geq 0$$

$$x_2 \geq 0$$

$$x_3 \geq 0$$

$$x_1 + x_2 + 2x_3 \leq 3$$

with starting point:

$$x_1 = 1.0$$

$$x_2 = 2.0$$

$$x_3 = 1.0$$

A suitable listing of the main program and subroutine FUNCT, 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,FLNLP1R,ID=HSRORNLER,MR=1.
FTN.
LOAD(TAPE)
LGO.
'         6400 END OF RECORD
      PROGRAM TST (INPUT,OUTPUT,TAPE5=INPUT,TAPE6=OUTPUT)
      DIMENSION X(3),G(3),H(15),EPS(3)
      CALL FLNLP1(3,15,4,X,G,H,EPS)
      STOP
      END
C
      SUBROUTINE FUNCT(X,F,G,U)
      DIMENSION X(3),G(3),C(4[illegible](3),GC(3,4),A(5),TT(5)
      B=2.*X(1)
      D=2.*X(2)
      F=2.*X(3)
      U=0.+B*(X(1)+X(2)+X(3)-4.)+D*(X(2)-3.)+X(3)*X(3)-2.*F
      C(1)=X(1)
      C(2)=X(2)
      C(3)=X(3)
      C(4)=2.-X(1)-X(2)-F
      GU(1)=-8.+2.*B+D+F
      GU(2)=-6.+2.*D+B
      GU(3)=-4.+F+B
      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.
      CALL FMIMAX(3,4,5,F,G,GU,C,GC,U,A,TT)
      RETURN
      END
'         6400 END OF RECORD
  100     1     1
0.0             1.0             1.E+            51.E-           5
1.0             2.0             1.0
1.E-            61.E-           61.E-           6
'            END OF FILE
```

Fig. 1 Main program and subroutine FUNCT for the example of Appendix A.
Data for the problem is also shown.

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

NUMBER OF INDEPENDENT VARIABLES.........................N   =    3

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

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

STARTING VALUE FOR VECTOR X(I)....................X( 1)  =    .100000E+0
                                                  X( 2)  =    .200000E+0
                                                  X( 3)  =    .100000E+0

TEST QUANTITIES TO BE USED......................EPS( 1)  =    .100000E-0
                                                EPS( 2)  =    .100000E-0
                                                EPS( 3)  =    .100000E-0

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

INITIAL VALUE OF THE PARAMETER ALPHA..................A0  =    .100000E+0

THE VALUE OF THE PARAMETER.............................P  =    .100000E+0

THE MARGIN BY WHICH CONSTRAINTS MAY BE VIOLATED.....EPSC  =    .100000E-0
```

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

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

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

   ARTIFICIAL UNCONSTRAINED FUNCTION  F =    .111112E+00
           ACTUAL OBJECTIVE FUNCTION  U =    .111111E+00

   X( 1) =    .133333E+01          G( 1) =    .975131E-04
   X( 2) =    .777777E+00          G( 2) =    .967782E-04
   X( 3) =    .444444E+00          G( 3) =    .197230E-03

   INEQUALITY CONSTRAINTS
   C( 1) =    .133333E+01
   C( 2) =    .777777E+00
   C( 3) =    .444444E+00
   C( 4) =    .139133E-05

         NUMBER OF FUNCTION EVALUATIONS =    77
     FINAL VALUE OF THE PARAMETER ALPHA =    .100000E+01
             EXECUTION TIME IN SECONDS =    .808
```

Fig. 3 Results for the example of Appendix A.

# APPENDIX B

To minimize

$$U = x_1^2 + x_2^2 + 2x_3^2 + x_4^2 - 5x_1 - 5x_2 - 21x_3 + 7x_4$$

subject to

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

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

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

with starting point:

$x_1 = 0.0$

$x_2 = 1.0$

$x_3 = 0.0$

$x_4 = 1.0$

Commands to access the demonstration program, a listing of the main program and subroutine FUNCT, instructions for entering data, a record of the input data and the final results are given in Figs. 4-8.

```
MCMASTER UNIV INTERCOM 4.1
DATE  08/08/73
TIME  14.55.23.

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




 08/08/73    LOGGED IN AT  14.56.15.
             WITH USER-ID BR
             EQUIP/PORT 70/06
COMMAND- ATTACH,A,FLNLP1D,ID=HSBOBNLER.
 PF CYCLE NO. = 005
COMMAND- CONNECT,INPUT,OUTPUT.
COMMAND- XEQ,A.
```

Fig. 4 Commands for accessing the demonstration program (option FLNLP1D) for the example of Appendix B.

```
C***********************************************************************
C                                                                     *
C   THIS IS A DEMONSTRATION PROGRAM FOR THE PACKAGE "FLNLP1".         *
C   THE LISTINGS OF THE MAIN PROGRAM AND THE SUBPROGRAM FUNCT         *
C   DEFINING THE OBJECTIVE FUNCTION REQUIRED FOR OPTIMIZATION         *
C   ARE SHOWN BELOW :                                                 *
C                                                                     *
C***********************************************************************
C
      PROGRAM TST(INPUT,OUTPUT,TAPE5=INPUT,TAPE6=OUTPUT)
      DIMENSION X(4),G(4),EPS(4),H(22),XS(4)
C   CALL THE PACKAGE "FLNLP1"
C     IN THE CALL STATEMENT :
C     THE FIRST ARGUMENT DEFINES THE NUMBER OF VARIABLES, SAY N
C     THE SECOND ONE DEFINES THE QUANTITY N*(N+7)/2
C     THE THIRD ONE DEFINES THE NUMBER OF CONSTRAINTS
      CALL FLNLP1(4,22,3,X,G,H,EPS,XS)
      STOP
      END
C
      SUBROUTINE FUNCT(X,F,G,U)
      DIMENSION X(4),G(4),C(3),GU(4),A(4),GC(4,3),TT(4)
      B=X(1)*X(1)
      R=X(2)*X(2)
      D=X(3)*X(3)
      E=X(4)*X(4)
C   THE ACTUAL OBJECTIVE FUNCTION
      U=B+R+2.*D+E-5.*(X(1)+X(2))-21.*X(3)+7.*X(4)
C   THE INEQUALITY CONSTRAINTS
      C(1)=-B-R-D-E-X(1)+X(2)-X(3)+X(4)+8.
      C(2)=-B-2.*R-D-2.*E+X(1)+X(4)+10.
      C(3)=-2.*B-R-D-2.*X(1)+X(2)+X(4)+5.
C   GRADIENTS OF THE OBJECTIVE FUNCTION WITH RESPECT TO THE VARIABLE
C   PARAMETERS
      GU(1)=2.*X(1)-5.
      GU(2)=2.*X(2)-5.
      GU(3)=4.*X(3)-21.
      GU(4)=2.*X(4)+7.
C   GRADIENTS OF THE CONSTRAINTS WITH RESPECT TO THE VARIABLE PARAMETERS
      GC(1,1)=-2.*X(1)-1.
      GC(2,1)=-2.*X(2)+1.
      GC(3,1)=-2.*X(3)-1.
      GC(4,1)=-2.*X(4)+1.
      GC(1,2)=GC(1,1)+2.
      GC(2,2)=-4.*X(2)
      GC(3,2)=GC(3,1)+1.
      GC(4,2)=-4.*X(4)+1.
      GC(1,3)=2.*GC(1,1)
      GC(2,3)=GC(2,1)
      GC(3,3)=GC(3,1)+1.
      GC(4,3)=1.
C   DEFINE THE ARTIFICIAL UNCONSTRAINED OBJECTIVE FUNCTION
C     IN THE CALL STATEMENT :
C     THE FIRST ARGUMENT DEFINES THE NUMBER OF VARIABLES
C     THE SECOND ONE DEFINES THE NUMBER OF CONSTRAINTS, SAY NC
C     THE THIRD ONE DEFINES THE QUANTITY NC+1
      CALL FMIMAX(4,3,4,F,G,GU,C,GC,U,A,TT)
      RETURN
      END
```

Fig. 5   Main program and subroutine FUNCT for the example of Appendix B.

```
YOU ARE WELCOME TO USE THE PACKAGE " F L N L P 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←      30
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←      20
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←      -100
SPECIFY THE INITIAL VALUE OF THE PARAMETER ALPHA.
5←      10
SPECIFY THE VALUE OF P.
6←      1.E+5
SPECIFY THE TOLERANCE OF VIOLATION FOR THE CONSTRAINTS.
7←      1.E-5
ENTER STARTING VALUES FOR THE VARIABLE PARAMETERS.
8←      0.  1.  0.  1.
ENTER SMALL VALUES FOR TESTING CONVERGENCE.
9←      1.E-6,1.E-6,1.E-6,1.E-6

ANY MODIFICATION
#       YES
WHICH ENTRY
#       1
SPECIFY THE MAXIMUM NUMBER OF ITERATIONS ALLOWED.
1←      100

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 input data for the example of Appendix B.

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

NUMBER OF INDEPENDENT VARIABLES.........................N =   4
MAXIMUM NUMBER OF ALLOWABLE ITERATIONS...............MAX = 100
INTERMEDIATE PRINTOUT AT EVERY IPT ITERATIONS........IPT =  20
STARTING VALUE FOR VECTOR X(I)
      X( 1) =  0.
      X( 2) =   .100000E+01
      X( 3) =  0.
      X( 4) =   .100000E+01
TEST QUANTITIES TO BE USED
   EPS( 1) =   .100000E-05
   EPS( 2) =   .100000E-05
   EPS( 3) =   .100000E-05
   EPS( 4) =   .100000E-05
ESTIMATE OF LOWER BOUND OF FUNCTION TO BE MINIMIZED..EST =  -.100000E+0
INITIAL VALUE OF THE PARAMETER ALPHA..................AO =   .100000E+0
THE VALUE OF THE PARAMETER..............................P =   .100000E+0
THE MARGIN BY WHICH CONSTRAINTS MAY BE VIOLATED.....EPSC =   .100000E-0
```

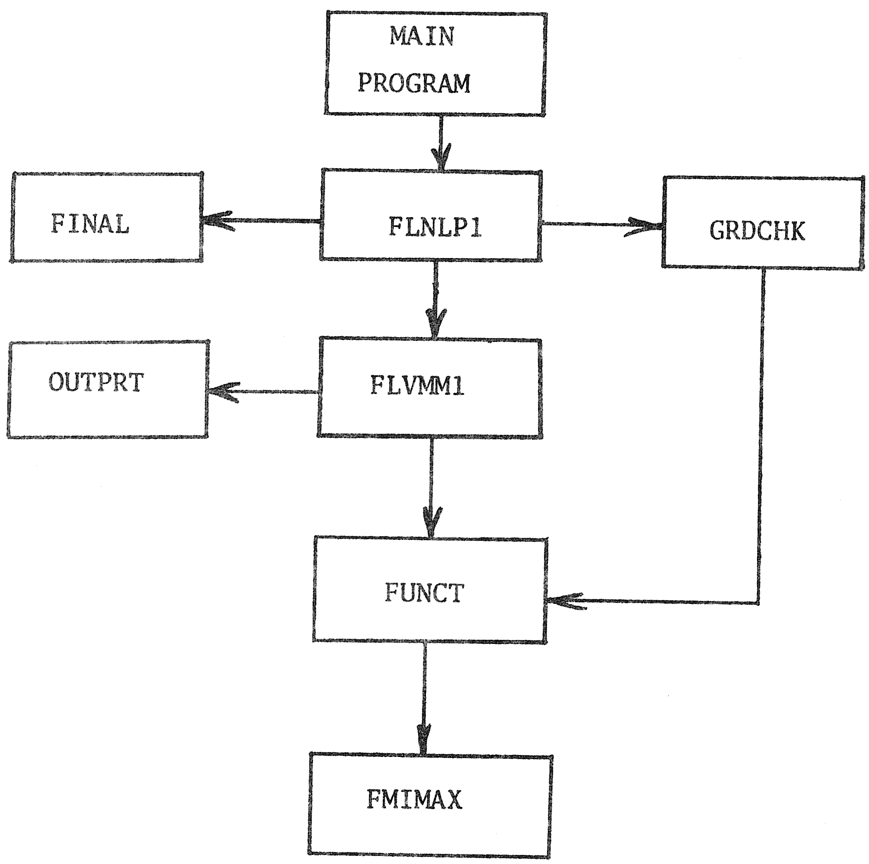
Fig. 7 Input data for the example of Appendix B.

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


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

   ARTIFICIAL UNCONSTRAINED FUNCTION  F =  -.439996E+02

            ACTUAL OBJECTIVE FUNCTION  U =  -.439998E+02

   X( 1) =  -.180004E-05          G( 1) =   .994442E-03
   X( 2) =   .999998E+00          G( 2) =   .288512E-03
   X( 3) =   .199999E+01          G( 3) =   .724725E-03
   X( 4) =  -.999989E+00          G( 4) =   .548195E-03

   INEQUALITY CONSTRAINTS
   C( 1) =   .856355E-04
   C( 2) =   .100010E+01
   C( 3) =   .551036E-04

          NUMBER OF FUNCTION EVALUATIONS =   88

      FINAL VALUE OF THE PARAMETER ALPHA =   .100000E+02

               EXECUTION TIME IN SECONDS =   .810

DO YOU WANT TO TERMINATE THE PROGRAM
#       YES
     STOP
```

Fig. 8 Results for the example of Appendix B.

# APPENDIX C

The general structure of the package is




Attached is the complete FORTRAN listing of the package - option FLNLP1I.

```
      SUBROUTINE FINLP1 (N,K,NC,X,G,H,EPS,XS)
      COMMON /BLK1/ NF
      COMMON /BLK/ KO
      COMMON /AA/ D,EPSC
      COMMON /CC/ ALPHA,IA,IC,M,IEXIT
      LOGICAL UNITH
      DIMENSION X(N), G(N), EPS(N), H(K), XS(N)
      DATA D/2HNO/
      UNITH=.TRUE.
      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,*) AO
      WRITE (6,47)
      READ (5,*) P
      WRITE (6,48)
      READ (5,*) EPSC
      WRITE (6,30)
      READ (5,*) X
      WRITE (6,31)
      READ (5,*) EPS
    1 WRITE (6,19)
      READ (5,20) ANS
      IF (ANS.EQ.D) GO TO 11
      WRITE (6,21)
      READ (5,*) NN
      GO TO (2,3,4,5,6,7,8,9,10), 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,*) AO
      GO TO 1
    7 WRITE (6,47)
      READ (5,*) P
      GO TO 1
    8 WRITE (6,48)
      READ (5,*) EPSC
      GO TO 1
    9 WRITE (6,30)
      READ (5,*) X
      GO TO 1
   10 WRITE (6,31)
```

```
      READ (5,*) EPS
      GO TO 1
11    WRITE (6,22)
      ALPHA=AO
      DO 12 I=1,N
      XS(I)=X(I)
12    CONTINUE
      IF (IP.EQ.0) GO TO 13
      WRITE (6,33)
      WRITE (6,34) N
      WRITE (6,35) MAX
      WRITE (6,36) IPT
      WRITE (6,37)
      WRITE (6,38) (I,X(I),I=1,N)
      WRITE (6,39)
      WRITE (6,40) (I,EPS(I),I=1,N)
      WRITE (6,41) EST
      WRITE (6,42) AO
      WRITE (6,49) P
      WRITE (6,50) EPSC
13    IC=0
      CALL GRDCHK (N,X,G)
      IF (IPT.EQ.0) GO TO 14
      WRITE (6,43)
      WRITE (6,44)
      WRITE (6,45)
14    M=0
      NF=1
      CALL SECOND (T1)
15    M=M+1
      CALL ELVMM1 (N,K,X,F,G,H,UNITH,EST,EPS,MAX,IPT)
      IC=1
      CALL FUNCT (X,F,G,U)
      IF (IA.EQ.0.OR.IEXIT.EQ.3) GO TO 16
      ALPHA=ALPHA*10.0
      GO TO 15
16    CALL SECOND (T2)
      CALL FINAL (N,X,F,G,NC,U)
      T=T2-T1
      WRITE (6,46) T
      WRITE (6,22)
      READ (5,20) ANS
      IF (ANS.NE.D) GO TO 18
      WRITE (6,23)
      READ (5,20) ANS
      IF (ANS.EQ.D) GO TO 1
      DO 17 I=1,N
      X(I)=XS(I)
17    CONTINUE
      GO TO 1
18    RETURN
C
C
C
19    FORMAT (1H0,*ANY MODIFICATION*/* , *)
20    FORMAT (A10)
21    FORMAT (1H ,*WHICH ENTRY*/* , *)
22    FORMAT (1H0/1H ,*DO YOU WANT TO TERMINATE THE PROGRAM*/* , *)
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 I N I P 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 ,*SPECIFY THE INITIAL VALUE OF THE PARAMETER ALPHA.*/*
     15( *)
30    FORMAT (1H ,*ENTER STARTING VALUES FOR THE VARIABLE PARAMETERS.*/*
     1 8( *)
31    FORMAT (1H ,*ENTER SMALL VALUES FOR TESTING CONVERGENCE.*/* 9( *)
32    FORMAT (//,* YOUR DATA IS NOW BEING PROCESSED.  IT MAY TAKE SOME T
     1IME BEFORE*/* RESULTS ARE AVAILABLE.  PLEASE BE PATIENT.*)
33    FORMAT (1H1,*INPUT DATA*/,1H ,10(*-*))
34    FORMAT (1H0,*NUMBER OF INDEPENDENT VARIABLES*,24(*.*),*N =*,I4)
35    FORMAT (1H ,*MAXIMUM NUMBER OF ALLOWABLE ITERATIONS*,15(*.*),*MAX
     1=*,I4)
36    FORMAT (1H ,*INTERMEDIATE PRINTOUT AT EVERY IPT ITERATIONS*,8(*.*)
     1,*IPT =*,I4)
37    FORMAT (1H ,*STARTING VALUE FOR VECTOR X(I)*)
38    FORMAT (1H ,5X,*X(*,I2,*) =*,F14.6)
39    FORMAT (1H ,*TEST QUANTITIES TO BE USED*)
40    FORMAT (1H ,3X,*EPS(*,I2,*) =*,F14.6)
41    FORMAT (1H ,*ESTIMATE OF LOWER BOUND OF FUNCTION TO BE MINIMIZED*,
     12(*.*),*EST =*,F14.6)
42    FORMAT (1H ,*INITIAL VALUE OF THE PARAMETER ALPHA*,18(*.*),*AO =*,
     1F14.6)
43    FORMAT (1H1)
44    FORMAT (1H ,*OPTIMIZATION BY FLETCHER METHOD*/,1H ,31(*-*))
45    FORMAT (1H0,*ITER.*,2X,*FUNCT.*,6X,*ALPHA*,8X,*OBJECTIVE*,6X,*VARI
     1ABLE*,7X,*GRADIENT*/1H ,1X,*NO.*,3X,*EVALU.*,19X,*FUNCTION*,6X,*VE
     2CTOR X(I)*,4X,*VECTOR G(I)*/)
46    FORMAT (1H0,14X,*EXECUTION TIME IN SECONDS =*,F7.3)
47    FORMAT (1H ,*SPECIFY THE VALUE OF P.*/* 6( *)
48    FORMAT (1H ,*SPECIFY THE TOLERANCE OF VIOLATION FOR THE CONSTRAINT
     1S.*/* 7( *)
49    FORMAT (1H ,*THE VALUE OF THE PARAMETER*,29(*.*),*P =*,F14.6)
50    FORMAT (1H ,*THE MARGIN BY WHICH CONSTRAINTS MAY BE VIOLATED*,5(*.
     1*),*EPSC =*,F14.6)
      END
```

```
      SUBROUTINE EMIMAX (N,NC,NT,F,G,GU,C,GC,U,A,TT)
      DIMENSION GU(N), C(NC), GC(N,NC), G(N), A(NT), TT(NT)
      COMMON /AA/ P,EPSC
      COMMON /BB/ PC(100)
      COMMON /CC/ ALPHA,IA,IC,M,IEXIT
      PP=P
      AE=0.0
      IA=0
      IF (NC.EQ.0.OR.ALPHA.EQ.0.0) GO TO 12
      DO 1 I=1,NC
      A(I)=U-ALPHA*C(I)
    1 CONTINUE
      AM=A(1)
      A(NT)=U
      DO 2 I=2,NT
      AM=AMAX1(AM,A(I))
    2 CONTINUE
      IF (AM.LE.0.0) PP=-PP
      SUM1=0.0
      DO 6 I=1,NT
      IF (AM) 5,3,4
    3 AE=1.E-8
      GO TO 5
    4 IF (A(I).LE.0.0) GO TO 6
    5 TT(I)=(A(I)-AE)/(AM-AE)
      SUM1=SUM1+TT(I)**PP
    6 CONTINUE
      F=(AM-AE)*SUM1**(1.0/PP)
      DO 11 I=1,N
      SUM2=0.0
      DO 10 J=1,NT
      IF (AM) 8,8,7
    7 IF (A(J).LE.0.0) GO TO 10
    8 IF (J.EQ.NT) GO TO 9
      SUM2=SUM2+TT(J)**(PP-1.0)*(GU(I)-ALPHA*GC(I,J))
      GO TO 10
    9 SUM2=SUM2+TT(J)**(PP-1.0)*GU(I)
   10 CONTINUE
      G(I)=SUM1**(1.0/PP-1.0)*SUM2
   11 CONTINUE
      GO TO 14
   12 F=U
      DO 13 I=1,N
      G(I)=GU(I)
   13 CONTINUE
   14 IF (IC.EQ.0.OR.NC.EQ.0) GO TO 16
      DO 15 I=1,NC
      PC(I)=C(I)
      CT=C(I)+EPSC
      IF (CT.LT.0.0) IA=1
   15 CONTINUE
   16 RETURN
      END
```

```
      SUBROUTINE FLVMM1 (N,K,X,F,G,H,UNITH,EST,EPS,MAX,IPT)
      DIMENSION X(N), G(N), EPS(N), H(K)
      COMMON /BLK1/ NF
      COMMON /BLK2/ KO
      COMMON /CC/ ALPHA,IA,IC,M,IEXIT
      LOGICAL CONV,UNITH
      KO=0
      CALL FUNCT (X,F,G,U)
      IF (F.LT.EST) GO TO 24
      IF (M.NE.1) GO TO 1
      ITN= 0
      ID=N+N
      IH=ID+N
1     STEP=1.0
      IF (.NOT.UNITH) GO TO 3
      IJ=IH+1
      DO 2 I=1,N
      DO 2 J=I,N
      H(IJ)=0.0
      IF (I.EQ.J) H(IJ)=1.0
2     IJ=IJ+1
3     CONV=.TRUE.
      GDX=0.0
      DO 7 I=1,N
      Z=0.0
      IJ=IH+I
      IF (I.EQ.1) GO TO 5
      L=I-1
      DO 4 J=1,L
      Z=Z-H(IJ)*G(J)
      IJ=IJ+N-J
4     CONTINUE
5     DO 6 J=I,N
      Z=Z-H(IJ)*G(J)
      IJ=IJ+1
6     CONTINUE
      IF (ABS(Z).GT.EPS(I)) CONV=.FALSE.
      H(N+I)=Z
      GDX=GDX+G(I)*Z
7     CONTINUE
      IF (IPT.EQ.0) GO TO 8
      IF (MOD(ITN,IPT).NE.0) GO TO 8
      CALL OUTPRT (X,N,G,F,NF,ITN)
8     IEXIT=1
      IF (CONV) GO TO 25
      IEXIT=2
      IF (GDX.GE.0.0) GO TO 25
      Z=1.0
      IF (ITN.LT.N.AND.UNITH) Z=STEP
      W=2.*(EST-F)/GDX
      IF (W.LT.Z) Z=W
      STEP=Z
9     GDX=GDX*Z
      DO 10 I=1,N
      H(N+I)=H(N+I)*Z
      X(I)=X(I)+H(N+I)
10    CONTINUE
      NF=NF+1
      CALL FUNCT (X,FP,H,U)
```

```
      IF (FP.LT.EST) GO TO 24
      IEXIT=3
      IF (ITN.EQ.MAX) GO TO 25
      GPDX=0.0
      DO 11 I=1,N
      H(ID+I)=H(I)-G(I)
      GPDX=GPDX+H(I)*H(N+I)
11    CONTINUE
      DGDX=GPDX-GDX
      IF (F.GT.FP-0.0001*GDX) GO TO 13
      IEXIT=4
      IF (GPDX.LT.0.0.AND.ITN.GT.N) [illegible]
      Z=2.0*(F-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 12 I=1,N
      X(I)=X(I)-H(N+I)
12    CONTINUE
      GO TO 15
13    F=FP
      DO 14 I=1,N
      G(I)=H(I)
14    CONTINUE
      IF (DGDX.GT.0.0) GO TO 16
      GDX=GPDX
      Z=4.0
15    STEP=Z*STEP
      GO TO 9
16    IF (GPDX.LT.0.5*GDX) STEP=2.0*STEP
      DGHDG=0.0
      DO 20 I=1,N
      Z=0.0
      IJ=IH+I
      IF (I.EQ.1) GO TO 18
      L=I-1
      DO 17 J=1,L
      Z=Z+H(IJ)*H(ID+J)
      IJ=IJ+N-J
17    CONTINUE
18    DO 19 J=I,N
      Z=Z+H(IJ)*H(ID+J)
      IJ=IJ+1
19    CONTINUE
      DGHDG=DGHDG+Z*H(ID+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(N+I)=W*H(N+I)-H(I)
21    CONTINUE
      DGDX=DGDX+DGHDG
      DGHDG=DGDX
22    IJ=IH
      DO 23 I=1,N
      W=H(N+I)/DGDX
      Z=H(I)/DGHDG
```

```
      DO 23 J=I,N
      IJ=IJ+1
23    H(IJ)=H(IJ)+W*H(N+J)-Z*H(J)
      ITN=ITN+1
      GO TO 3
24    IFEXIT=5
25    IF (IFEXIT.EQ.1) KO=1
      GO TO (26,27,28,27,29), IFEXIT
26    WRITE (6,31) IFEXIT
      GO TO 30
27    WRITE (6,32) IFEXIT
      GO TO 30
28    WRITE (6,33) IFEXIT
      GO TO 30
29    WRITE (6,34) IFEXIT
30    CONTINUE
      RETURN
C
C
C
31     FORMAT (1H1,*IEXIT =*,I2,/1H ,*CRITERION FOR OPTIMUM (CHANGE IN VE
      1CTOR X .LT. EPS) HAS BEEN SATISEFIED*)
32     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*)
33     FORMAT (1H1,*IEXIT =*,I2,/1H ,*MAXIMUM NUMBER OF ALLOWABLE ITERATI
      1ON HAS BEEN EXCEEDED*)
34     FORMAT (1H ,*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)
      J=0
      CALL FUNCT (X,F,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,F2,T,U)
      X(I)=X(I)-2.0*DX
      CALL FUNCT (X,F1,T,U)
      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) J=1
1     CONTINUE
      IF (J.EQ.1) GO TO 2
      WRITE (6,6)
      RETURN
2     WRITE (6,7)
      CALL EXIT
C
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 ,9X,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,F,NE,ITN)
      DIMENSION X(N), G(N)
      COMMON /CC/ ALPHA,IA,IC,M,IEXIT
      WRITE (6,1) ITN,NE,ALPHA,F,((X(I),G(I)),I=1,N)
      RETURN
C
C
C
1     FORMAT (1H ,I3,5X,I3,6X,F10.4,1X,F14.6,1X,8(F14.6,1X,F14.6,/,44X)
     1)
      END
```

```
      SUBROUTINE FINAL (N,X,F,G,NC,U)
      DIMENSION X(N), G(N)
      COMMON /BLK/ KO
      COMMON /BLK1/ NF
      COMMON /CC/ ALPHA,IA,IC,M,IEXIT
      COMMON /BP/ PC(100)
      IF (KO.EQ.0) GO TO 1
      WRITE (6,4)
      GO TO 2
1     WRITE (6,5)
2     WRITE (6,6) F
      WRITE (6,9) U
      WRITE (6,7) (I,X(I),I,G(I),I=1,N)
      IF (NC.EQ.0) GO TO 3
      WRITE (6,10)
      WRITE (6,11) (I,PC(I),I=1,NC)
3     WRITE (6,8) NF
      WRITE (6,12) ALPHA
      RETURN
C
C
C
4     FORMAT (1H0,/1H0,*OPTIMAL SOLUTION FOUND BY FLETCHER METHOD*/1H ,4
     11(*-*))
5     FORMAT (1H0,/1H0,*RESULTS FOUND BY FLETCHER METHOD AT LAST ITERATI
     1ON*/1H ,50(*-*))
6     FORMAT (1H0,3X,*ARTIFICIAL UNCONSTRAINED FUNCTION  F =*,E14.6)
7     FORMAT (1H ,3X,*X(*,I2,*) =*,E14.6,10X,*G(*,I2,*) =*,E14.6)
8     FORMAT (1H0,9X,*NUMBER OF FUNCTION EVALUATIONS =*,I5)
9     FORMAT (1H0,11X,*ACTUAL OBJECTIVE FUNCTION  U =*,E14.6/)
10    FORMAT (1H0,3X,*INEQUALITY CONSTRAINTS*)
11    FORMAT (1H ,3X,*C(*,I2,*) =*,E14.6)
12    FORMAT (1H0,5X,*FINAL VALUE OF THE PARAMETER ALPHA =*,E14.6)
      END
```