INTERNAL REPORTS IN

SIMULATION, OPTIMIZATION AND CONTROL

No. SOC-108

FLOPT3 -AN INTERACTIVE PROGRAM FOR LEAST PTH OPTIMIZATION WITH EXTRAPOLATION TO MINIMAX SOLUTIONS

J.W. Bandler and W.Y. Chu

October 1975

FACULTY OF ENGINEERING
McMASTER UNIVERSITY
HAMILTON, ONTARIO, CANADA

FLOPT3- AN INTERACTIVE PROGRAM FOR LEAST PTH OPTIMIZATION WITH EXTRAPOLATION TO MINIMAX SOLUTIONS

J.W. Bandler and W.Y. Chu

Abstract FLOPT3 is a package of subroutines primarily for solving least pth optimization problems. Its main features include interactive input/output information exchange, 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 PDP 11/45 computer. Three 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

FLOPT3 is a package of subroutines primarily for solving least pth optimization problems. Its main features include interactive input/output information exchange 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 FLOPT3 is a parallel version of the program FLOPT2 [5]. It has been used on a PDP 11/45 computer and is written in FORTRAN IV. It requires about 16,000 words of core memory. Three examples of varying complexity are used to illustrate the versatility of the program. Up to 100 functions can be currently handled.

II. ARGUMENT LIST

SUBROUTINE FLOPT3 (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 double-precision 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 FLOPT3 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 on final exit from the subroutine FLOPT3.

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/OUTPUT INFORMATION INTERACTION

Some of the parameters in the argument list have to be supplied as input data and there are also other parameters required by the program. All these are handled interactively. The program will request the required input data in the form of replies to questions. The user may enter the parameter values in any format, but only one value per line. The user will be asked to retype his entry should the program consider it to be syntactically wrong. Some questions are to be answered by YES or NO (the abbreviated form Y or N is also allowed). After the user has entered all the data, he may modify any specific entry.

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

MAX 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.

EST The estimated minimum value of the objective function.

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.
PO	The value of p used in the least pth formulation or the initial value of p when a sequence of p values is used.
JORDER	The highest order of estimates required in extrapolation (JORDER $\leq$ IK-1).
FACTOR	The multiplying factor for p when a sequence of p values is used together with extrapolation (FACTOR > 1). When extrapolation is used with the Fiacco-McCormick method, it is the factor by which the sequence of r is decreased.

Fig. 1 shows a list of questions regarding input data information. The data entered are for the Example 1. The user will first be asked to indicate whether he is using the least pth objective formulation. Depending on the answer, some subsequent questions will not appear, e.g., the value of p or the multiplying factor for p. Extrapolation and the highest order are optional. After the input data is complete, the user will be asked to provide instructions about the output printing. The user may have a hardcopy (printing on the teletype) or output will appear on the screen. At the start of each optimization, such instructions are requested. To facilitate viewing of the results, the program will pause at the end of each optimization. A A005 000000 message will appear on the screen. The user should type CONTINUE to continue execution. When extrapolation is used, estimates of the minimax solution will be printed after the pause.

The user may now terminate his program or supply instructions for the next optimization. The user is allowed to audit the error functions at the next starting point. The value of the maximum error and normalized values of all the errors will be printed on the screen. Based on this information, the user may select those error functions which he thinks would be active for the next optimization. Only the selected error functions will be used in the objective formulation. Some computational effort may thus be saved. This manual reduction scheme has the same effect as the automatic reduction scheme in the program FLOPT2.

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 FLOPT3. In using the extrapolation feature, the subroutine FLOPT3 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 FLOPT3 (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(100)
```

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 indirectly by the program),

JD is an integer array used as an index set (set indirectly 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 e_1 w.r.t. x_n
        GO TO 99
2       ER(2) = e_2(x_1,x_2,...,x_n)
        GE(1,2) = partial derivative of e_2 w.r.t. x_1
        .
        .
        .
        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 3 in Section VII).

V. OTHER SUBPROGRAMS

The following is a brief description of the subroutines and function subprograms called by FLOPT3.

DATAIN	reads input data.
FREAD	reads data in free format.
NVAL	obtains the sign and value of an input item.
IREPLY	interprets the answer to a YES/NO question.
DATACY	prints a listing of the input data.
EXIO	prints the reason of exit from the minimization subroutine QUASIN.
RESULT	outputs the optimal solution.
EXTRAP	performs extrapolation in double precision.
GRDCHK	checks the gradient formulation by perturbation.
LEASTP	formulates a least pth objective function and the necessary gradients.
QUASIN	minimizes a function using the Fletcher unconstrained minimization program by a quasi-Newton method.

The overall structure of the package is shown in Fig. 2.

VI. OPERATING PROCEDURES

The program FLOPT3 has been developed such that minimal effort is required of the user. The user, however, has to supply the main program and a function subroutine to define the objective function and partial derivatives. The user is therefore expected to have some knowledge of the Disk Operating System (DOS) of the computer and system programs like FORTRAN IV, Edit-11 Text Editor, PIP (File Utility Package) and Link-11 Linker. The new user to the PDP11 should consult the following manuals:

A. PDP-11 Disk Operating System Monitor Programmer's Handbook, DEC-11-OMONA-A-D

B. PDP-11 PIP, File Utility Package Programmer's Manual, DEC-11-UPUPA-A-D

C. PDP-11 Edit-11 Text Editor Programmer's Manual, DEC-11-EEDA-D

D. PDP-11 FORTRAN IV Programmer's Manual, DEC-11-LFIVA-A-D

E. PDP-11 Link-11 Linker and Libr-11 Librarian Programmer's Manual, DEC-11-ULLMA-A-D

The package FLOPT3 is stored on the system disk with FLOPT3. OBJ as its filename and extension. The file is an object module and formed as a concatenation of 13 object modules. It is under the uic (User Identification Code) of 200,210. To solve his problems, the user has to link his main program, subroutine FUNCT or other service subroutines with FLOPT3 and the FORTRAN library. A sample sequence of procedures for solving an optimization problem is shown below:

<u>$</u> LO 200,210. <CR> (log into the system; 200,210 is the uic used by the authors; the user may get another one from the system manager)

```
$ DA 21-MAY-75  <CR>                  (enter the date of the day)

$ RU EDIT  <CR >                      (call the EDIT program)

# MAIN. FTN < KB: <CR>                (create the main program)

* I. <CR>

                       [ Program ]
                       [  text   ]

<LF>

* [correction, if any] or

* EX  <CR>                            (close the file)

# FUNCT.FTN  <CR>                     (create the subroutine FUNCT)

* I  <CR>

                       [ Program ]
                       [  text   ]

<LF>

* [correction, if any] or

* EX  <CR>                            (close the file)

# ↑C                                  (exit from the

. KI  <CR>                             EDIT program)

$ RU FORTRN   <CR>                    (call the FORTRAN compiler)

# MAIN, PP:  <  MAIN  <CR>            (compile the main program; a listing

                                       will appear on the teletype)

# FUNCT, PP:  <  FUNCT  <CR>          (compile subroutine FUNCT)

# ↑C                                  (exit from FORTRAN compiler)

. KI  <CR>

$ RU LINK  <CR>                       (call the Linker program)

# PRO1  <  MAIN, FUNCT, FLOPT3/CC, FTNLIB/L/U/E  <CR>
```

	(link the main program, subroutine FUNCT, FLOPT3 and the FORTRAN library; note that the main program must be the first in the input files)
# ↑C	(exit from the Linker)
. KI <CR>	
$ RU PRO1 <CR>	(load and execute the program)
I350 000000	(FORTRAN stop message)
$ FI. <CR>	(finish the session and leave the system)

VII. EXAMPLES

A minimax example, a nonlinear programming problem 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 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^{-5}.

Example 1: A minimax example [6]

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, 36 function evaluations yielded $x_1 = 1.13903$, $x_2 = 0.89957$. All the three functions were used in the initial objective formulation. Two functions were then selected at the end of the first optimization. Fig. 3 shows the main program and the subroutine FUNCT. Note that in using the selection 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. 4. Fig. 5 shows the results of the 5th optimization and the final estimates of the minimax solution.

Example 2: Rosen-Suzuki function [7]

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 $\check{\underset{\sim}{x}} = [0 \quad 1 \quad 2 \quad -1]^T$. 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 = 2, 16, 128, 1024, 53 function evaluations yielded $x_1 = -0.00000$, $x_2 = 1.00000$, $x_3 = 2.00000$, $x_4 = -1.00000$. With the selection scheme, only active constraints were considered at the later stages of the process. A listing of the main program and the subroutine FUNCT is shown in Fig. 6. Statements were added (in subroutine FUNCT) to allow printing of the constraints at the extrapolated solutions.

Fig. 7 shows a printout of the input data. Fig. 8 shows the final solution of the problem.

Example 3: A microwave circuit example

The design of a three-section 100-percent relative bandwidth 10:1 transmission-line transformer [8] 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 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 413 network analyses were required, which was about 35% less than what would be required if the reduction scheme was not used. A listing of the main program and subroutine FUNCT is shown in Fig. 9. Note that the sample points are defined in 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. 10 shows the input data for this example. Fig. 11 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. 12. In Fig. 13, 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.

VIII. COMMENTS

The package is so organised that pertinent information of the optimization process can be obtained from the argument list of the subroutine FLOPT3. This allows the user to do some useful things in the main program, especially when using extrapolation. Some suggestions are:

(i) In using the extrapolation procedure, we usually do not know how many cycles of optimization are required and the parameter IK may be set too large. As the program is interactive, the user may terminate his program before the maximum number is reached. However, to ensure that the user can use the information from the argument list of FLOPT3, in the main program the parameter IFINIS is used as an indicator for termination of execution. IFINIS will be equal to n when the user indicates that the program is to terminate. Control will be returned to the main program. It is therefore advisable to put the statement

IF (IFINIS.EQ.N) STOP

inside the DO loop that calls the subroutine FLOPT3 as a user-controlled stopping criterion. See Examples 1,2 and 3.

(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 and 3. When extrapolation is used, but not the least pth formulation (as in the Fiacco-McCormick rethod), the sequence of the controlling parameter r can be updated in the main program.

IX. CONCLUSIONS

A package of subroutines, called FLOPT3, for solving least pth optimization problems has been presented. Its features, which include input/output information interaction, 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], [9], [10].

ACKNOWLEDGEMENT

The authors would like to thank Dr. W. Kinsner, H. deBruin and P. Edmonson for helpful suggestions.

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, "FLOPT2 - A program for least pth optimization with extrapolation to minimax solutions", Faculty of Engineering, McMaster University, Hamilton, Canada, April 1975, Report SOC-84.

[6] 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).

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

[8] 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.

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

[10] 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.

Parameter p	Function evaluations	x	Number of error functions	=	Number of network analyses
8	21		11		231
48	17		6		102
288	11		4		44
1728	9		4		36
	Total 58				Total 413

Table 1. Computational effort for the transformer problem.

```
YOU ARE WELCOME TO USE THE PROGRAM * F L O P T 3 *.
YOU MAY ENTER YOUR DATA IN ANY FORMAT, HOWEVER ONE ENTRY
PER LINE ONLY.    THANK YOU.

1. ARE YOU USING LEAST PTH OBJECTIVE FORMULATION ?
Y

2. MAXIMUM NUMBER OF ITERATIONS ?
1ØØ

3. INTERMEDIATE OUTPUT TO BE PRINTED AT EVERY IPT
ITERATIONS......IPT ?
1Ø

4. MINIMUM ESTIMATED VALUE OF THE OBJECTIVE FUNCTION ?
Ø

5. STARTING VALUES FOR THE  2 VARIABLE PARAMETERS ?
2
2

6.  2 SMALL VALUES FOR TESTING CONVERGENCE ?
1.E-5
1.E-5

7. VALUE OF THE PARAMETER P ?
4

8. DO YOU WANT EXTRAPOLATION ?
Y

9. HIGHEST ORDER OF ESTIMATES REQUIRED ?
3

1Ø. DO YOU WANT THE ESTIMATES TO BE PRINTED ?
Y

11. MULTIPLYING FACTOR IN P VALUE ?
4

ANY MODIFICATION ?
N

DO YOU WANT A HARDCOPY OF THE PRINTOUTS ?
Y

DO YOU WANT A LISTING OF THE INPUT DATA ?
Y
```

Figure 1. Interactive input data acquisition.

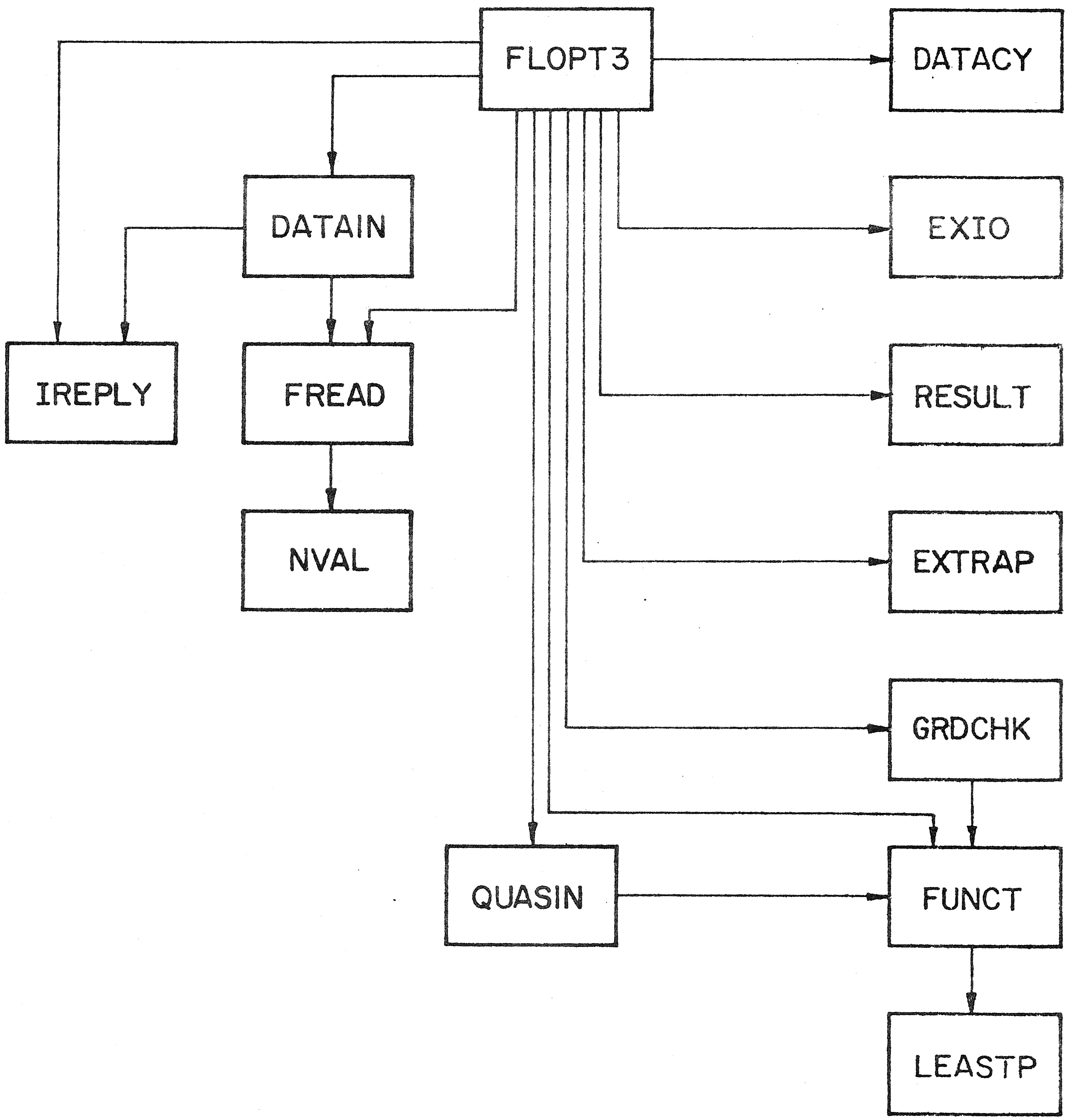

Figure 2. Overall structure of FLOPT3.

```
C        MAIN PROGRAM OF EXAMPLE 1
C
       REAL*8 XE(2,5,4)
       DIMENSION X(2),G(2),EPS(2),H(3),W(8),XB(2)
       N=2
       NR=3
       M=1
       IGK=0
       IK=5
       DO 1 IH=1,IK
       CALL FLOPT3(N,M,IGK,X,G,H,W,EPS,XE,IH,IK,FACTOR,XB,IFINIS,NR)
        IF(IFINIS.EQ.N)STOP
       M=0
1       CONTINUE
       STOP
       END




       SUBROUTINE FUNCT(X,G,U)
C        A MINIMAX EXAMPLE
C
       DIMENSION X(2),G(2),ER(3),GE(2,3),ES(3)
       COMMON/WY3/NA,JD(100)
       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)
       GOTO(1,2,3),K
1       ER(1)=Y1+Y2*Y2
        GE(1,1)=Y3
       GE(2,1)=(Y2+Y2)*Y4
       GOTO12
2       ER(2)=8.-4.*(X(1)+X(2))+Y1+Y2
       GE(1,2)=-4.+Y3
       GE(2,2)=-4.+Y4
       GOTO12
3       ER(3)=2.*EXP(-X(1)+X(2))
       GE(1,3)=-ER(3)
       GE(2,3)=ER(3)
12      CONTINUE
       CALL LEASTP(N,U,G,ER,GE,ES)
       RETURN
       END
```

Figure 3. Main program and subroutine FUNCT for the minimax example.

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

NUMBER OF INDEPENDENT VARIABLES.......................N =   2
MAXIMUM NUMBER OF ALLOWABLE ITERATIONS..............MAX = 100
INTERMEDIATE PRINTOUT AT EVERY IPT ITERATIONS.......IPT =  10
ESTIMATE OF LOWER BOUND OF FUNCTION TO BE MINIMIZED.EST =  0.0000000E 00
STARTING VALUE FOR VECTOR X(I)....................X( 1) =  0.2000000E 01
                                                  X( 2) =  0.2000000E 01
TEST QUANTITIES TO BE USED......................EPS( 1) =  0.1000000E-04
                                                EPS( 2) =  0.1000000E-04
INITIAL VALUE OF THE PARAMETER P.....................PO =  0.4000000E 01
HIGHEST ORDER OF ESTIMATES USED IN EXTRAPOLATION.JORDER =   3
MULTIPLYING FACTOR IN P VALUE....................FACTOR =  0.4000000E 01
```

Figure 4. Input data for the minimax example.

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

      U =  Ø.1953528E Ø1

     EM =  Ø.1952453E Ø1
 EN( 1) =  Ø.9997273E ØØ
 EN( 2) =  Ø.1ØØØØØØE Ø1

  X( 1) =  Ø.1138949E Ø1       G( 1) = -Ø.2862335E-Ø3
  X( 2) =  Ø.8995245E ØØ       G( 2) = -Ø.3285604E-Ø3

NO. OF FUNCTION EVALUATIONS =    36
VALUE OF P =  Ø.1Ø24ØØØE Ø4

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

ORDER 1
X( 1) =  Ø.1139Ø3ØD Ø1
X( 2) =  Ø.8995657D ØØ
ORDER 2
X( 1) =  Ø.1139Ø28D Ø1
X( 2) =  Ø.8995667D ØØ
ORDER 3
X( 1) =  Ø.1139Ø28D Ø1
X( 2) =  Ø.8995671D ØØ
CORRESPONDING NORMALIZED ERRORS
     EM =  Ø.1952225E Ø1
 EN( 1) =  Ø.9999992E ØØ
 EN( 2) =  Ø.1ØØØØØØE Ø1
```

Figure 5. Final results for the minimax example.

```
C        MAIN PROGRAM OF EXAMPLE 2
C
       REAL*8 XE(4,6,4)
       DIMENSION X(4),G(4),H(10),W(16),EPS(4),XB(4)
       COMMON/USER/IGRAD
       N=4
       NR=4
       M=1
       IGK=1
       IK=6
       DO 1 IH=1,IK
       IGRAD=1
       CALL FLOPT3(N,M,IGK,X,G,H,W,EPS,XE,IH,IK,FACTOR,XB,IFINIS,NR)
       IGRAD=0
       CALL SETFIL(5,'USER',IERR,'PP')
       CALL FUNCT(XB,G,U)
       ENDFILE 5
       IF(IFINIS.EQ.N) STOP
       M=0
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(100)
       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)
```

Figure 6. Main program and subroutine FUNCT for the Rosen-Suzuki problem.

```
      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.
      ER(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)=-EE-EE+1.
      DO 22 J=1,4
      GE(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.
      ER(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
      GE(J,3)=GF(J)-ALFA*GC(J,3)
33     CONTINUE
      GO TO 9
4      ER(4)=F
      DO 44 J=1,4
      GE(J,4)=GF(J)
44     CONTINUE
9      CONTINUE
      CALL LEASTP(N,U,G,ER,GE,ES)
      RETURN
5      C(1)=-B-R-D-E-X(1)+X(2)-X(3)+X(4)+8.
      C(2)=-B-R-R-D-E-E+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(/'SOLUTION   ',4E15.7)
      PRINT 7,F
7      FORMAT('OBJECTIVE FUNCTION     ',E15.7)
      PRINT 8,(C(I),I=1,3)
8      FORMAT('CONSTRAINTS      ',3E15.7)
      RETURN
      END
```

Figure 6. [Continued]

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

NUMBER OF INDEPENDENT VARIABLES.........................N =    4
MAXIMUM NUMBER OF ALLOWABLE ITERATIONS...............MAX =  100
INTERMEDIATE PRINTOUT AT EVERY IPT ITERATIONS.......IPT =   10
ESTIMATE OF LOWER BOUND OF FUNCTION TO BE MINIMIZED.EST = -0.1000000E 03
STARTING VALUE FOR VECTOR X(I)....................X( 1) =  0.0000000E 00
                                                  X( 2) =  0.0000000E 00
                                                  X( 3) =  0.0000000E 00
                                                  X( 4) =  0.0000000E 00
TEST QUANTITIES TO BE USED......................EPS( 1) =  0.1000000E-04
                                                EPS( 2) =  0.1000000E-04
                                                EPS( 3) =  0.1000000E-04
                                                EPS( 4) =  0.1000000E-04
INITIAL VALUE OF THE PARAMETER P....................PO =  0.2000000E 01
HIGHEST ORDER OF ESTIMATES USED IN EXTRAPOLATION.JORDER =    3
MULTIPLYING FACTOR IN P VALUE...................FACTOR =  0.8000000E 01
```

Figure 7. Input data for the Rosen-Suzuki problem.

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

ORDER 1
X( 1) = -0.6592716D-05
X( 2) =  0.9999988D 00
X( 3) =  0.1999995D 01
X( 4) = -0.9999882D 00
ORDER 2
X( 1) = -0.4500303D-05
X( 2) =  0.1000001D 01
X( 3) =  0.2000002D 01
X( 4) = -0.9999979D 00
ORDER 3
X( 1) = -0.4742049D-05
X( 2) =  0.1000002D 01
X( 3) =  0.2000002D 01
X( 4) = -0.9999989D 00
CORRESPONDING NORMALIZED ERRORS
     EM = -0.4399995E 02
EN( 1) =   0.1000000E 01
EN( 3) =   0.1000001E 01
EN( 4) =   0.1000001E 01


SOLUTION    -0.4742049E-05   0.1000002E 01   0.2000002E 01 -0.9999989E 00
 OBJECTIVE FUNCTION        -0.4400001E 02
CONSTRAINTS       -0.5722046E-05   0.9999847E 00 -0.9536743E-06
```

Figure 8. Final results for the Rosen-Suzuki problem.

```
C         MAIN PROGRAM OF EXAMPLE 3
C
        REAL*8 XE(6,5,4)
        DIMENSION X(6),G(6),H(21),W(24),EPS(6),XB(6)
        DIMENSION GRAD(6)
        COMMON/USER/WN(11)
        COMMON/WY3/NA,JD(100)
        N=6
        NR=11
        WN(1)=.5
        WN(2)=.6
        WN(3)=.7
        WN(4)=.77
        WN(5)=.9
        WN(6)=1.
        WN(7)=1.1
        WN(8)=1.23
        WN(9)=1.3
        WN(10)=1.4
        WN(11)=1.5
        M=1
        IGK=1
        IK=5
        DO 1 IH=1,IK
        CALL FLOPT3(N,M,IGK,X,G,H,W,EPS,XE,IH,IK,FACTOR,XB,IFINIS,NR)
        CALL SETFIL(5,'USER',IERR,'PP')
        PRINT 22
22       FORMAT(/3X,'RESPONSES OF THE TRANSFORMER')
        PRINT 23
23       FORMAT(/6X,'FREQUENCY',3X,'REFLECTION COEF.',5X,'BEST')
        DO 2 I=1,NA
        K=JD(I)
        CALL NET(G,WN(K),ARHOG,GRAD,0)
        CALL NET(XB,WN(K),ARHOB,GRAD,0)
        PRINT 24,WN(K),ARHOG,ARHOB
24       FORMAT(3F15.7)
2        CONTINUE
        ENDFILE 5
        IF(IFINIS.EQ.N) STOP
        M=0
1        CONTINUE
        STOP
        END
```

Figure 9. Main program and subroutine FUNCT for the transformer example.

```
      SUBROUTINE FUNCT(X,G,U)
C      A MICROWAVE CIRCUIT EXAMPLE
C
       DIMENSION X(6),G(6),ER(11),GE(6,11),ES(11),GRAD(6)
      COMMON/USER/WN(11)
      COMMON/WY3/NA,JD(100)
      N=6
      DO 1 I=1,NA
      K=JD(I)
      CALL NET(X,WN(K),ARHO,GRAD,1)
      ER(K)=ARHO
      DO 2 J=1,N
      GE(J,K)=GRAD(J)
2      CONTINUE
1      CONTINUE
      CALL LEASTP(N,U,G,ER,GE,ES)
      RETURN
      END
```

Figure 9. [Continued]

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

NUMBER OF INDEPENDENT VARIABLES.........................N =    6
MAXIMUM NUMBER OF ALLOWABLE ITERATIONS...............MAX =  1ØØ
INTERMEDIATE PRINTOUT AT EVERY IPT ITERATIONS.......IPT =   2Ø
ESTIMATE OF LOWER BOUND OF FUNCTION TO BE MINIMIZED.EST =   Ø.ØØØØØØE ØØ
STARTING VALUE FOR VECTOR X(1)....................X( 1) =   Ø.8ØØØØØE ØØ
                                                  X( 2) =   Ø.15ØØØØE Ø1
                                                  X( 3) =   Ø.12ØØØØE Ø1
                                                  X( 4) =   Ø.3ØØØØØE Ø1
                                                  X( 5) =   Ø.3ØØØØØE ØØ
                                                  X( 6) =   Ø.6ØØØØØE Ø1
TEST QUANTITIES TO BE USED......................EPS( 1) =   Ø.1ØØØØØE-Ø4
                                                EPS( 2) =   Ø.1ØØØØØE-Ø4
                                                EPS( 3) =   Ø.1ØØØØØE-Ø4
                                                EPS( 4) =   Ø.1ØØØØØE-Ø4
                                                EPS( 5) =   Ø.1ØØØØØE-Ø4
                                                EPS( 6) =   Ø.1ØØØØØE-Ø4
INITIAL VALUE OF THE PARAMETER P.....................PO =   Ø.8ØØØØØE Ø1
HIGHEST ORDER OF ESTIMATES USED IN EXTRAPOLATION.JORDER =    3
MULTIPLYING FACTOR IN P VALUE....................FACTOR =   Ø.6ØØØØØE Ø1
```

Figure 10. Input data for the transformer example.

```
IEXIT = 1
NORMAL EXIT

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

      U =   Ø.1974401E ØØ

     EM =   Ø.1973214E ØØ
 EN( 1) =   Ø.1ØØØØØØE Ø1
 EN( 4) =   Ø.9999564E ØØ
 EN( 8) =   Ø.9996739E ØØ
 EN(11) =   Ø.9993590E ØØ

  X( 1) =   Ø.9999480E ØØ      G( 1) = -Ø.2497691E-Ø2
  X( 2) =   Ø.1634748E Ø1      G( 2) = -Ø.3275858E-Ø2
  X( 3) =   Ø.9999981E ØØ      G( 3) =  Ø.1Ø44Ø27E-Ø2
  X( 4) =   Ø.3162416E Ø1      G( 4) =  Ø.2282112E-Ø4
  X( 5) =   Ø.9999595E ØØ      G( 5) = -Ø.2468Ø78E-Ø2
  X( 6) =   Ø.6117451E Ø1      G( 6) =  Ø.8732353E-Ø3

NO. OF FUNCTION EVALUATIONS =    58
VALUE OF P =   Ø.1728ØØØE Ø4
```

Figure 11. Results for the transformer example.

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

ORDER 3
X( 1) =  Ø.9999917D ØØ
X( 2) =  Ø.1634749D Ø1
X( 3) =  Ø.1ØØØØØ1D Ø1
X( 4) =  Ø.3162424D Ø1
X( 5) =  Ø.1ØØØØØ6D Ø1
X( 6) =  Ø.6117469D Ø1
CORRESPONDING NORMALIZED ERRORS
     EM =  Ø.1972921E ØØ
EN( 1) =  Ø.1ØØØØØØE Ø1
EN( 4) =  Ø.9999921E ØØ
EN( 8) =  Ø.9999802E ØØ
EN(11) =  Ø.9999934E ØØ
```

Figure 12. Final estimate of the minimax solution.

```
RESPONSES OF THE TRANSFORMER

    FREQUENCY    REFLECTION COEF.      BEST
    Ø.5ØØØØØØ       Ø.1973214       Ø.1972921
    Ø.77ØØØØØ       Ø.1973128       Ø.19729Ø5
    1.23ØØØØØ       Ø.197257Ø       Ø.1972882
    1.5ØØØØØØ       Ø.1971948       Ø.19729Ø8
```

Figure 13. Responses of the 3-section transformer.

Complete listing

of

FLOPT3

```
      SUBROUTINE FLOPT3(N,M,IGK,X,G,H,W,EPS,XE,IH,IK,FACTOR,XB,IFINIS,
     $  NR)
C
C      THIS SUBROUTINE CO-ORDINATES INPUT, MINIMIZATION AND OUTPUT.
C
      REAL*8 XE(N,IK,1)
      DIMENSION X(N),G(N),EPS(N),H(1),W(1),XB(N)
      COMMON/WY1/IFN,KO,IFNT
      COMMON/WY2/P,JV,EM,EN(100)
      COMMON/WY3/NA,JD(100)
      IF(IFINIS.EQ.N) RETURN
      IFINIS=0
C
C      ... ENABLE GRADIENT COMPUTATION, READ INPUT DATA
      JV=0
      IF(M.EQ.1) CALL DATAIN(N,MAX,IPT,EST,X,EPS,PO,IEX,JORDER,JPRINT,
     $  FACTOR,NR,LSP)
      WRITE(6,42)
42     FORMAT(' DO YOU WANT A HARDCOPY OF THE PRINTOUTS ?'/)
      IF(IREPLY(4).EQ.1)GOTO71
C
C      ... HARDCOPY ON "TELETYPE"
C          SOFTCOPY ON "CRT TERMINAL"
      CALL SETFIL(5,'SOFT',IERR,'KB')
      GOTO72
71     CALL SETFIL(5,'HARD',IERR,'PP')
72     IF(M.EQ.0)GOTO39
C
C      ... INITIALIZATION, SET INDEX SET
      IFNT=0
      MODE=1
       P=PO
      NA=NR
      DO 2 I=1,NR
      JD(I)=I
2      CONTINUE
C
C      ... PRINT INPUT DATA WHEN REQUESTED
      WRITE(6,40)
40     FORMAT(' DO YOU WANT A LISTING OF THE INPUT DATA ?'/)
      IF(IREPLY(3).EQ.1) CALL DATACY(N,MAX,IPT,EST,X,EPS,PO,IEX,JORDER
     $,FACTOR,NR,LSP)
C
C      ... GRADIENT CHECK
      IF(IGK.EQ.1) CALL GRDCHK(N,X,G,W)
39     P=PO
      PRINT 21,IH
21     FORMAT(/' OPTIMIZATION',I3/1H ,15('-'))
      IF(IPT.EQ.0)GOTO3
      PRINT 22
22     FORMAT(/' ITER.',2X,'FUNCT.',6X,'OBJECTIVE',7X,'VARIABLE',8X,
     $  'GRADIENT'/'  NO.',3X,'EVALU.',6X,'FUNCTION',7X,'VECTOR X(I)',
     $  5X,'VECTOR G(I)'/)
```

```
C
C       ... OPTIMIZATION BY FLETCHER METHOD(1972)
3        CALL QUASIN(N,X,U,G,H,W,EST,EPS,MODE,MAX,IPT,IEXIT)
        CALL EXIO(IEXIT)
        IFNT=IFNT+IFN
C
C       ... PRINT RESULTS
        CALL RESULT(N,X,G,U,NR,LSP)
C
C       ... PAUSE FOR EASY VIEWING
        PAUSE
        JV=1
        MODE=3
        IF(IK.EQ.1) GOTO55
        IF(IEX.EQ.Ø) GOTO4
C
C       ... STORE CURRENT MINIMUM IN ARRAY G
        DO 5 I=1,N
        G(I)=X(I)
5        CONTINUE
C
C       ... EXTRAPOLATION
        CALL EXTRAP(N,X,XE,IH,IK,FACTOR,XB,JORDER)
        J1=JORDER+1
        IH1=IH-1
        IF(JPRINT.EQ.Ø.OR.IH.EQ.1)GOTO4
        IJ=J1
        IF(IH.LE.J1)IJ=IH
C
C       ... PRINT ESTIMATES
        PRINT 23
23       FORMAT(//' ESTIMATES OF THE MINIMAX SOLUTION BY EXTRAPOLATION'/,
     $  1H ,5Ø('-')/)
        DO 6 L=2,IJ
        L1=L-1
        PRINT 24,L1
24       FORMAT(' ORDER',I2)
        DO 6 J=1,N
        PRINT 25,J,XE(J,IH,L)
25       FORMAT(' X(',I2,') =',D15.7)
6        CONTINUE
        IF(LSP.EQ.Ø) GOTO55
C
C       ... PRINT FUNCTION & ERRORS AT THE BEST SOLUTION
        CALL FUNCT(XB,G,U)
        PRINT 74
74       FORMAT(' CORRESPONDING NORMALIZED ERRORS')
        PRINT 32,EM
        DO 45 I=1,NA
        J=JD(I)
        PRINT 33,J,EN(J)
45       CONTINUE
4        CONTINUE
```

```
      IF(IH.EQ.IK)GOTO55
C
C       ... TERMINATE THE PROGRAM ?
      WRITE(6,26)
26     FORMAT(1HØ,'DO YOU WANT TO TERMINATE THE PROGRAM ?'/)
      IF(IREPLY(Ø).EQ.Ø)GOTO7
55     IFINIS=N
      ENDFILE 5
      RETURN
7      IF(IEX.EQ.Ø)GOTO9
      WRITE(6,27)
27     FORMAT(1HØ,'DO YOU WANT TO AUDIT THE ERROR FUNCTIONS AT THE'/
     $' NEXT STARTING POINT ?'/)
      IF(IREPLY(1).EQ.Ø)GOTO9
C
C       ... RESET THE INDEX SET, CALCULATE FUNCTION & ERRORS
C           AT THE NEXT STARTING POINT
      NA=NR
      DO 1Ø I=1,NR
      JD(I)=I
1Ø     CONTINUE
      CALL FUNCT(X,G,U)
      WRITE(6,3Ø)
3Ø     FORMAT('ØERROR FUNCTIONS AT THE NEXT STARTING POINT')
      WRITE(6,32)EM
32     FORMAT(5X,'EM =',E15.7)
      WRITE(6,33) (I,EN(I),I=1,NR)
33     FORMAT(' EN(',I2,') =',E15.7)
C
C       ... SELECT ERRORS FOR THE NEXT OPTIMIZATION,
C           INDEX SET IS SET BY THE USER
9      WRITE(6,34)
34     FORMAT('ØTYPE Ø IF ALL ERROR FUNCTIONS ARE TO BE USED'/' OTHERWI
     $SE ENTER THE NUMBERS OF THE SELECTED ERRORS'/' AND TERMINATE BY TY
     $PING Ø'/)
      CALL FREAD(VALUE)
      JD(1)=IFIX(VALUE)
      IF(JD(1).EQ.Ø) GOTO17
      NR1=NR+1
      DO 16 I=2,NR1
      CALL FREAD(VALUE)
      JD(I)=IFIX(VALUE)
      NA=I-1
      IF(JD(I).EQ.Ø)GOTO18
16     CONTINUE
18     WRITE(6,35)
35     FORMAT('ØYOU HAVE ENTERED THE FOLLOWING NUMBERS :')
      WRITE(6,36) (JD(I),I=1,NA)
36     FORMAT(12I5)
      WRITE(6,37)
37     FORMAT('ØANY MODIFICATION ?'/)
      IF(IREPLY(2).EQ.1)GOTO9
      GOTO19
```

```
C
C         ... SET THE INDEX SET TO BE THE WHOLE SET
17        NA=NR
        DO 12 I=1,NR
        JD(I)=I
12        CONTINUE
19        WRITE(6,38)
38        FORMAT('ØWE ARE READY FOR THE NEXT OPTIMIZATION,'/' PLEASE WAIT
       $ FOR RESULTS.'/)
C
C         ... INCREASE THE VALUE OF P
        PO=P*FACTOR
        ENDFILE 5
        RETURN
        END
```

```
      SUBROUTINE DATAIN(N,MAX,IPT,EST,X,EPS,PO,IEX,JORDER,JPRINT,
     $ FACTOR,NR,LSP)
C
C      THIS SUBROUTINE READS IN DATA
C
      DIMENSION X(N),EPS(N)
      IEX=0
      IW=0
      WRITE(6,21)
21     FORMAT('1YOU ARE WELCOME TO USE THE PROGRAM * F L O P T 3 *.'/,
     $ ' YOU MAY ENTER YOUR DATA IN ANY FORMAT, HOWEVER ONE ENTRY'/,
     $ ' PER LINE ONLY.    THANK YOU.'/)
C
C      ... READ INPUT DATA
1      WRITE(6,51)
51     FORMAT(' 1. ARE YOU USING LEAST PTH OBJECTIVE FORMULATION ?'/)
       LSP=IREPLY(0)
      IF(IW.EQ.1)GOTO88
2      WRITE(6,52)
52     FORMAT(' 2. MAXIMUM NUMBER OF ITERATIONS ?'/)
C
C      ... GET  MAX
      CALL FREAD(VALUE)
      MAX=IFIX(VALUE)
      IF(IW.EQ.1)GOTO88
3      WRITE(6,53)
53     FORMAT(' 3. INTERMEDIATE OUTPUT TO BE PRINTED AT EVERY IPT'/' IT
     $ERATIONS......IPT ?'/)
C
C      ... GET  IPT
      CALL FREAD(VALUE)
      IPT=IFIX(VALUE)
      IF(IW.EQ.1)GOTO88
4      WRITE(6,54)
54     FORMAT(' 4. MINIMUM ESTIMATED VALUE OF THE OBJECTIVE FUNCTION ?'
     $/)
C
C      ... GET  EST
      CALL FREAD(VALUE)
      EST=VALUE
      IF(IW.EQ.1)GOTO88
5      WRITE(6,55) N
55     FORMAT(' 5. STARTING VALUES FOR THE',I3,' VARIABLE PARAMETERS ?'
     $/)
      DO 32 I=1,N
C
C      ... GET  X(I), I=1,N
      CALL FREAD(VALUE)
      X(I)=VALUE
32     CONTINUE
      IF(IW.EQ.1)GOTO88
6      WRITE(6,56) N
56     FORMAT(' 6.',I3,' SMALL VALUES FOR TESTING CONVERGENCE ?'/)
```

```
      DO 33 I=1,N
C
C       ... GET  EPS(I), I=1,N
      CALL FREAD(VALUE)
      EPS(I)=VALUE
33      CONTINUE
      IF(IW.EQ.1)GOTO88
      IF(LSP.EQ.1)GOTO7
      GOTO8
7       WRITE(6,57)
57      FORMAT(' 7. VALUE OF THE PARAMETER P ?'/)
C
C       ... GET  P
      CALL FREAD(VALUE)
      PO=VALUE
      IF(IW.EQ.1)GOTO88
8       WRITE(6,58)
58      FORMAT(' 8. DO YOU WANT EXTRAPOLATION ?'/)
      IF(IREPLY(1).EQ.Ø)GOTO11
      IEX=1
      IF(IW.EQ.1)GOTO88
9       WRITE(6,59)
59      FORMAT(' 9. HIGHEST ORDER OF ESTIMATES REQUIRED ?'/)
C
C       ... GET  JORDER
      CALL FREAD(VALUE)
      JORDER=IFIX(VALUE)
      IF(IW.EQ.1)GOTO88
1Ø      WRITE(6,6Ø)
6Ø      FORMAT(' 1Ø. DO YOU WANT THE ESTIMATES TO BE PRINTED ?'/)
      JPRINT=IREPLY(2)
      IF(IW.EQ.1)GOTO88
11      WRITE(6,61)
61      FORMAT(' 11. MULTIPLYING FACTOR IN P VALUE ?'/)
C
C       ... GET  FACTOR
      CALL FREAD(VALUE)
      FACTOR=VALUE
      IF(IW.EQ.1)GOTO88
      IW=1
C
C       ... OPTION TO MODIFY
88      WRITE(6,64)
64      FORMAT(' ANY MODIFICATION ?'/)
      IF(IREPLY(3).EQ.Ø)RETURN
      WRITE(6,92)
92      FORMAT(' WHICH ENTRY ?'/)
95      CALL FREAD(VALUE)
      NN=IFIX(VALUE)
C
C       ... WARNING MESSAGE FOR NON-EXISTENT ENTRY
      IF(NN.LT.1.OR.NN.GT.11)GOTO12
      GOTO(1,2,3,4,5,6,7,8,9,1Ø,11),NN
12      WRITE(6,62)
62      FORMAT(' ** INVALID REQUEST, PLEASE RETYPE **'/)
      GOTO95
      END
```

```
      SUBROUTINE FREAD(VALUE)
C
C      THIS SUBROUTINE READS DATA IN FREE FORMAT
C
      COMMON/WY5/LINE(3Ø),ICOL,IERR
      DATA IBLK/1H /
95     IERR=Ø
      ISGN=1
      ICOL=1
      VALUE=Ø.Ø
C
C      ... ONLY THE FIRST 3Ø COLUMNS ARE TO BE INTERPRETED
      READ(6,111) (LINE(I),I=1,3Ø)
111    FORMAT(3ØA1)
C
C      ... SKIP EXTRA BLANKS
1      IF(ICOL.GT.3Ø) RETURN
      IF(LINE(ICOL).NE.IBLK) GO TO 2
      ICOL=ICOL+1
      GO TO 1
C
C      ... CHECK FOR MINUS SIGN
2      ICOL=ICOL-1
      N=NVAL(Ø)
      IF(N-13) 4,5,99
4      IF(N-11) 7,11,9
5      ISGN=Ø
C
C      ... GET INTEGER PART
9      N=NVAL(Ø)
      IF(N-1Ø) 7,8Ø,1Ø
1Ø     IF(LINE(ICOL).EQ.IBLK) GOTO8Ø
      IF(N-12) 11,99,32
7      VALUE=VALUE*1Ø.Ø+FLOAT(N)
      GOTO9
C
C      ... GET FRACTION PART
11     WT=1.Ø
12     N=NVAL(Ø)
      IF(N-1Ø) 13,8Ø,14
14     IF(LINE(ICOL).EQ.IBLK)GOTO8Ø
32     IF(N.EQ.14) GOTO15
      GOTO99
13     WT=WT*Ø.1
      VALUE=VALUE+WT*FLOAT(N)
      GOTO12
C
C      ... GET EXPONENT PART
15     IEXP=Ø
      IXS=1
      N=NVAL(Ø)
```

```
C
C        ... CHECK FOR MINUS SIGN
      IF(N-13)16,17,99
16     IF(N-11) 18,99,19
17     IXS=0
19     N=NVAL(0)
18     IF(N-10) 20,21,22
20     IEXP=IEXP*10+N
      GOTO19
21     IF(IXS.EQ.0) IEXP=-IEXP
      VALUE=VALUE*10.0**IEXP
      GOTO80
22     IF(LINE(ICOL).NE.IBLK)GOTO99
      GOTO21
C
C        ... NORMAL EXIT
80     IF(ISGN.EQ.0) VALUE=-VALUE
      RETURN
99     IERR=IERR+1
C
C        ... SKIP TO END OF FIELD, PRINT ERROR MESSAGE
92     ICOL=ICOL+1
      IF(ICOL.GT.30) GOTO93
      IF(LINE(ICOL).NE.IBLK) GOTO92
93     WRITE(6,94)
94     FORMAT(' ** ILLEGAL DATA, PLEASE RETYPE **'/)
      GOTO95
      END
```

```
      INTEGER FUNCTION NVAL(IDUM)
C
C      OBTAIN SIGN OR VALUE
C
       DIMENSION NUMER(15)
      COMMON/WY5/LINE(30),ICOL,IERR
      DATA NUMER/1H0,1H1,1H2,1H3,1H4,1H5,1H6,1H7,1H8,1H9,1H ,1H.,1H+,1
     $H-,1HE/
C
C      ... INTERPRET THE FIRST 30 COLUMNS
      IF(ICOL.LT.30)GOTO1
      ICOL=31
      NVAL=10
      RETURN
C
C      ... DETERMINE THE CONTENT OF EACH COLUMN
1      ICOL=ICOL+1
      DO 2 I=1,15
      IF(LINE(ICOL).EQ.NUMER(I))GOTO3
2      CONTINUE
      I=16
3      NVAL=I-1
      RETURN
      END
```

```
      INTEGER FUNCTION IREPLY(IDUMMY)
C
C      INTERPRETS YES OR NO ANSWER
C
      INTEGER ANS
      DIMENSION NY(4)
      DATA NY/2HN ,2HNO,2HY ,2HYE/
9      READ(6,1)ANS
1      FORMAT(A2)
C
C      ... IREPLY=1 FOR "YES", IREPLY=0 FOR "NO"
      IF(ANS.EQ.NY(1).OR.ANS.EQ.NY(2))GOTO2
      IF(ANS.EQ.NY(3).OR.ANS.EQ.NY(4))GOTO3
      GOTO4
2      IREPLY=0
      RETURN
3      IREPLY=1
      RETURN
C
C      ... WARNING MESSAGE
4      WRITE(6,5)
5      FORMAT(' ** ANSWER NOT CLEAR, PLEASE RETYPE **'/)
      GOTO9
      END
```

```
      SUBROUTINE DATACY(N,MAX,IPT,EST,X,EPS,PO,IEX,JORDER,FACTOR,
     $ NR,LSP)
C
C      THIS SUBROUTINE PRINTS INPUT DATA
C
      DIMENSION X(N),EPS(N)
      PRINT 1
      PRINT 2,N
      PRINT 3,MAX
      PRINT 4,IPT
      PRINT 5,EST
      PRINT 6,X(1)
      PRINT 7,(I,X(I),I=2,N)
      PRINT 8,EPS(1)
      PRINT 9,(I,EPS(I),I=2,N)
      IF(LSP.EQ.1) PRINT 10,PO
      IF(IEX.EQ.0) RETURN
      PRINT 11,JORDER
      PRINT 12,FACTOR
      RETURN
1      FORMAT(//' INPUT DATA'/1H ,10('-')/)
2      FORMAT(' NUMBER OF INDEPENDENT VARIABLES',23('.'),'N =',I4)
3      FORMAT(' MAXIMUM NUMBER OF ALLOWABLE ITERATIONS',14('.'),
     $ 'MAX =',I4)
4      FORMAT(' INTERMEDIATE PRINTOUT AT EVERY IPT ITERATIONS',7('.'),
     1'IPT =',I4)
5      FORMAT(' ESTIMATE OF LOWER BOUND OF FUNCTION TO BE MINIMIZED',
     1 '.EST =',E14.6)
6      FORMAT(' STARTING VALUE FOR VECTOR X(I)',20('.'),'X( 1) =',
     $ E14.6)
7      FORMAT(51X,'X(',I2,') =',E14.6)
8      FORMAT(' TEST QUANTITIES TO BE USED',22('.'),'EPS( 1) =',E14.6)
9      FORMAT(49X,'EPS(',I2,') =',E14.6)
10     FORMAT(' INITIAL VALUE OF THE PARAMETER P',21('.'),'PO =',E14.6)
11     FORMAT(' HIGHEST ORDER OF ESTIMATES USED IN EXTRAPOLATION.',
     1'JORDER =',I4)
12     FORMAT(' MULTIPLYING FACTOR IN P VALUE',20('.'),'FACTOR =',
     $ E14.6)
      END
```

```
      SUBROUTINE EXIO(IEXIT)
C
C       THIS SUBROUTINE PRINTS THE REASON OF EXIT FROM THE MINIMIZATION
 C      SUBROUTINE
C
      IF(IEXIT.EQ.Ø)GOTO1
      GOTO2
1       PRINT 4,IEXIT
4       FORMAT(/' IEXIT =',I2/' THE ESTIMATE OF THE HESSIAN MATRIX IS NO
     $T POSITIVE DEFINITE')
      GOTO3
2       GOTO(5,6,7),IEXIT
5       PRINT 8,IEXIT
      GOTO3
6       PRINT 9,IEXIT
      GOTO3
7       PRINT 1Ø,IEXIT
3       RETURN
8       FORMAT(/' IEXIT =',I2/' NORMAL EXIT')
9       FORMAT(/' IEXIT =',I2/' EPS CHOSEN IS TOO SMALL')
1Ø      FORMAT(/' IEXIT =',I2/' MAXIMUM NUMBER OF ALLOWABLE ITERATIONS H
     $AS BEEN REACHED')
      END
```

```
      SUBROUTINE RESULT(N,X,G,U,NR,LSP)
C
C      THIS SUBROUTINE PRINTS RESULTS
C
      DIMENSION X(N),G(N)
      COMMON/WY1/IFN,KO,IFNT
      COMMON/WY2/P,JV,EM,EN(100)
      COMMON/WY3/NA,JD(100)
      IF(KO.EQ.0)GOTO1
      PRINT 3
      GOTO2
1      PRINT 4
2      PRINT 5,U
      IF(LSP.EQ.0)GOTO8
      PRINT 9,EM
      DO 15 I=1,NA
      J=JD(I)
      PRINT 10,J,EN(J)
15     CONTINUE
      PRINT 11
8      PRINT 6,(I,X(I),I,G(I),I=1,N)
      PRINT 7,IFNT
      IF(LSP.EQ.1) PRINT 12,P
      RETURN
3      FORMAT(//' OPTIMAL SOLUTION FOUND BY FLETCHER METHOD'/,1H ,
     $ 41('-'))
4      FORMAT(//' RESULTS FOUND BY FLETCHER METHOD AT LAST ITERATION'/,
     $ 1H , 50('-'))
5      FORMAT(/6X,'U =',E15.7/)
6      FORMAT(2X,'X(',I2,') =',E15.7,6X,'G(',I2,') =',E15.7)
7      FORMAT(/' NO. OF FUNCTION EVALUATIONS =',I5)
9      FORMAT(5X,'EM =',E15.7)
10     FORMAT(' EN(',I2,') =',E15.7)
11     FORMAT(1H )
12     FORMAT(' VALUE OF P =',E15.7)
      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)
      PRINT 3
      PRINT 4
      DO 1 I=1,N
      Z=X(I)
      DX=1.E-4*Z
      IF(ABS(DX).LT.1.E-7)DX=1.E-4
C
C       ... PERTURB IN THE FORWARD DIRECTION
      X(I)=Z+DX
      CALL FUNCT(X,W,U2)
C
C       ... PERTURB IN THE BACKWARD DIRECTION
      X(I)=Z-DX
      CALL FUNCT(X,W,U1)
      Y=0.5*(U2-U1)/DX
      X(I)=Z
      IF(ABS(Y).LT.1.E-14)Y=1.E-14
      IF(ABS(G(I)).LT.1.E-14)G(I)=1.E-14
C
C       ... CALCULATE PERCENTAGE ERROR
      YP=ABS((Y-G(I))/Y)*100.
      PRINT 5,G(I),Y,YP
       IF(YP.GT.10.)IC=1
1      CONTINUE
      IF(IC.EQ.1)GOTO2
      PRINT 6
      RETURN
2      PRINT 7
      CALL EXIT
3      FORMAT(//' GRADIENT CHECK AT STARTING POINT'/1H ,32('-'))
4      FORMAT(/' ANALYTICAL GRADIENTS',5X,'NUMERICAL GRADIENTS'
     9,5X,'PERCENTAGE ERROR')
5      FORMAT(3X,E15.7,9X,E15.7,8X,E15.7)
6      FORMAT(/' GRADIENTS ARE O.K.')
7      FORMAT(/' GRADIENTS ARE INCORRECT, EXECUTION IS ABORTED')
      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,EM,EN(100)
      COMMON/WY3/NA,JD(100)
      J=JD(1)
      EM=ER(J)
C
C      ... FIND THE MAXIMUM ERROR
       DO 1 I=2,NA
      J=JD(I)
      EM=AMAX1(EM,ER(J))
1      CONTINUE
      IF(EM.NE.0.)GOTO3
      DO 2 I=1,NA
      J=JD(I)
      ER(J)=ER(J)-1.E-7
2      CONTINUE
      EM=EM-1.E-7
3      Q=SIGN(P,EM)
      S1=0.
C
C      ... FORM THE OBJECTIVE FUNCTION
      DO 5 I=1,NA
      J=JD(I)
      IF(EM.LT.0.)GOTO4
      IF(ER(J).LE.0.)GOTO5
C
C      ... NORMALIZE ERRORS
4      EN(J)=ER(J)/EM
      ES(J)=EN(J)**Q
      S1=S1+ES(J)
5      CONTINUE
      ST=S1**(1./Q)
C
C      ... OBJECTIVE FUNCTION
      U=EM*ST
C
C      ... FOR JV=1, GRADIENTS WILL NOT BE CALCULATED
      IF(JV.EQ.1)RETURN
C
C      ... FORM THE GRADIENTS
      ST=ST/S1
      DO 8 I=1,N
       S2=0.
      DO 7 J=1,NA
      K=JD(J)
      IF(EM.LT.0.)GOTO6
      IF(ER(K).LE.0.)GOTO7
6      S2=S2+ES(K)*GE(I,K)/EN(K)
7      CONTINUE
      G(I)=ST*S2
8      CONTINUE
      RETURN
      END
```

```
      SUBROUTINE EXTRAP(N,X,XE,I,IK,FACTOR,XB,JORDER)
C
C     THIS SUBROUTINE PERFORMS EXTRAPOLATION
C
      REAL*8 XE(N,IK,1),S
      DIMENSION X(1),XB(1)
      II=I+1
      DO 1 J=1,N
      XE(J,I,1)=DBLE(X(J))
1     CONTINUE
      IF(I.LT.2) GO TO 11
C
C     ... EXTRAPOLATE TO MINIMAX SOLUTION
      IF(I.GT.JORDER) GOTO2
      IJ=I
      GOTO3
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.)
4     CONTINUE
5     CONTINUE
C
C     ... PUT THE BEST ESTIMATE IN ARRAY XB
      DO 6 J=1,N
      XB(J)=SNGL(XE(J,I,IJ))
6     CONTINUE
C
C     ... ESTIMATE THE MINIMUM OF THE NEXT OPTIMIZATION
      DO 7 J=1,N
      XE(J,II,IJ)=XE(J,I,IJ)
7     CONTINUE
      DO 9 K=2,IJ
      L=IJ+1-K
      S=FACTOR**L
      DO 8 J=1,N
      XE(J,II,L)=((S-1.)*XE(J,II,L+1)+XE(J,I,L))/S
8     CONTINUE
9     CONTINUE
C
C     ... SET THE ESTIMATED MINIMUM TO BE THE NEXT STARTING POINT
      DO 10 J=1,N
      X(J)=SNGL(XE(J,II,1))
10    CONTINUE
      RETURN
11    DO 12 J=1,N
      XB(J)=SNGL(XE(J,I,1))
12    CONTINUE
      RETURN
      END
```

```
      SUBROUTINE QUASIN (N,X,U,G,H,W,EST,EPS,MODE,MAX,IPT,IEXIT)
C
C      THIS SUBROUTINE IS THE FLETCHER(1972) METHOD OF MINIMIZATION
C
      DIMENSION X(1),G(1),H(1),W(1),EPS(1)
      COMMON/WY1/IFN,KO,IFNT
      KO=Ø
      NP=N+1
      N1=N-1
      NN=N*NP/2
      IS=N
      IU=N
      IV=N+N
      IB=IV+N
      IEXIT=Ø
      IF(MODE.EQ.3)GOTO15
      IF(MODE.EQ.2)GOTO1Ø
      IJ=NN+1
      DO 5 I=1,N
      DO 6 J=1,I
      IJ=IJ-1
6      H(IJ)=Ø.
5      H(IJ)=1.
      GOTO15
1Ø     CONTINUE
      IJ=1
      DO 11 I=2,N
      Z=H(IJ)
      IF(Z.LE.Ø.)RETURN
      IJ=IJ+1
      I1=IJ
      DO 11 J=I,N
      ZZ=H(IJ)
      H(IJ)=H(IJ)/Z
      JK=IJ
      IK=I1
      DO 12 K=I,J
      JK=JK+NP-K
      H(JK)=H(JK)-H(IK)*ZZ
12     IK=IK+1
11     IJ=IJ+1
      IF(H(IJ).LE.Ø.)RETURN
15     CONTINUE
      IJ=NP
      DMIN=H(1)
      DO 16 I=2,N
      IF(H(IJ).GE.DMIN)GOTO16
      DMIN=H(IJ)
16     IJ=IJ+NP-I
      IF(DMIN.LE.Ø.)RETURN
      Z=EST
      ITN=Ø
      CALL FUNCT(X,G,U)
```

```
      IFN=1
      DF=U-EST
      IF(DF.LE.0.)DF=1.
20     CONTINUE
      IF(IPT.EQ.0)GOTO21
      IF(MOD(ITN,IPT).NE.0)GOTO21
      PRINT 501,ITN,IFN,U,((X(I),G(I)),I=1,N)
501    FORMAT(1H ,I3,5X,I3,4X,E15.7,1X,20(E15.7,E16.7/32X))
21     CONTINUE
      ITN=ITN+1
      W(1)=-G(1)
      DO 22 I=2,N
      IJ=I
      I1=I-1
      Z=-G(I)
      DO 23 J=1,I1
      Z=Z-H(IJ)*W(J)
23     IJ=IJ+N-J
22     W(I)=Z
      W(IS+N)=W(N)/H(NN)
      IJ=NN
      DO 25 I=1,N1
      IJ=IJ-1
      Z=0.
      DO 26 J=1,I
      Z=Z+H(IJ)*W(IS+NP-J)
26     IJ=IJ-1
25     W(IS+N-I)=W(N-I)/H(IJ)-Z
      GS=0.
      DO 29 I=1,N
29     GS=GS+W(IS+I)*G(I)
      IEXIT=2
      IF(GS.GE.0.)GOTO92
      GSO=GS
      ALPHA=-(DF+DF)/GS
      IF(ALPHA.GT.1.)ALPHA=1.
      DF=U
      TOT=0.
30     CONTINUE
      IEXIT=3
       IF(ITN.EQ.MAX)GOTO92
      ICON=0
      IEXIT=1
      DO 31 I=1,N
      Z=ALPHA*W(IS+I)
      IF(ABS(Z).GE.EPS(I))ICON=1
31     X(I)=X(I)+Z
      CALL FUNCT(X,W,FY)
      IFN=IFN+1
      GYS=0.
      DO 32 I=1,N
32     GYS=GYS+W(I)*W(IS+I)
      IF(FY.GE.U)GOTO40
```

```
      IF(ABS(GYS/GSO).LE..9)GOTO50
      IF(GYS.GT.0.)GOTO40
      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
      GOTO30
40     CONTINUE
      DO 41 I=1,N
41     X(I)=X(I)-ALPHA*W(IS+I)
      IF(ICON.EQ.0)GOTO92
      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
      GOTO30
50     CONTINUE
      ALPHA=ALPHA+TOT
      U=FY
      IF(ICON.EQ.0)GOTO90
      DF=DF-U
      DGS=GYS-GSO
      LINK=1
      IF(DGS+ALPHA*GSO.GT.0.)GOTO52
      DO 51 I=1,N
51     W(IU+I)=W(I)-G(I)
      SIG=1./(ALPHA*DGS)
      GOTO70
52     CONTINUE
      ZZ=ALPHA/(DGS-ALPHA*GSO)
      Z=DGS*ZZ-1.
      DO 53 I=1,N
53     W(IU+I)=Z*G(I)+W(I)
      SIG=1./(ZZ*DGS*DGS)
      GOTO70
60     CONTINUE
      LINK=2
      DO 61 I=1,N
61     W(IU+I)=G(I)
      IF(DGS+ALPHA*GSO.GT.0.)GOTO62
      SIG=1./GSO
      GOTO70
62     CONTINUE
      SIG=-ZZ
      GOTO70
65     CONTINUE
      DO 66 I=1,N
66     G(I)=W(I)
      GOTO20
```

```
70        CONTINUE
        W(IV+1)=W(IU+1)
        DO 71 I=2,N
        IJ=I
        I1=I-1
        Z=W(IU+I)
        DO 72 J=1,I1
        Z=Z-H(IJ)*W(IV+J)
72        IJ=IJ+N-J
71        W(IV+I)=Z
        IJ=1
        DO 75 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
75        IJ=IJ+NP-I
        IJ=1
        DO 80 I=1,N1
        IJ=IJ+1
        I1=I+1
         DO 80 J=I1,N
        IUJ=IU+J
        W(IUJ)=W(IUJ)-H(IJ)*W(IV+I)
        H(IJ)=H(IJ)+W(IB+I)*W(IUJ)
80        IJ=IJ+1
        GOTO(60,65),LINK
90        CONTINUE
        DO 91 I=1,N
91        G(I)=W(I)
92        CONTINUE
        IF(IEXIT.EQ.1)KO=1
        IF(IPT.EQ.0)RETURN
        PRINT 501,ITN,IFN,U,((X(I),G(I)),I=1,N)
        RETURN
        END
```