

INTERNAL REPORTS IN
SIMULATION, OPTIMIZATION
AND CONTROL

No. SOC-19

A SPECIAL PROGRAM FOR LEAST pTH

APPROXIMATION INCLUDING INTERPOLATION

J.R. Popović and J.W. Bandler

October 1973

FACULTY OF ENGINEERING
McMASTER UNIVERSITY
HAMILTON, ONTARIO, CANADA



A SPECIAL PROGRAM FOR LEAST pTH APPROXIMATION INCLUDING
INTERPOLATION

PURPOSE: Minimization of a least pth objective function of k variables using gradient methods. Interpolation brings the discrete problem closer to the continuous minimax approximation problem.

LANGUAGE: FORTRAN IV; 1023 cards, including comments.

AUTHORS: J.R. Popović and J.W. Bandler
Department of Electrical Engineering,
McMaster University,
1280 Main St. W.,
Hamilton, Ontario, Canada. L8S 4L7

AVAILABILITY: A user's manual with an example and program listing is appended.

DESCRIPTION: The program, called FMCLP, can be used for fitting a continuous approximating function to another single specified function or data on a closed interval and thus is relevant in optimization used in computer-aided circuit and system design, and modelling [1].

FMCLP utilizes the practical least pth approximation approach with extremely large values of p proposed by

This work was presented at the 16th Midwest Symp. on Circuit Theory, Waterloo, Canada, April 12-13, 1973.

Bandler and Charalambous [2] in conjunction with efficient gradient minimization algorithms such as Fletcher-Powell [3] and the Fletcher method [4]. Discrete least pth approximation with $p=2$ is the well known discrete least squares approximation and with extremely large values of p the corresponding optimal approximations tend to become discrete minimax (or Chebyshev) approximations. Proper scaling is used to alleviate the ill-conditioning resulting from very large values of p , such as 10^6 . Quadratic interpolation is employed to bring the discrete problem closer to the continuous minimax approximation problem. Using quadratic interpolation the sampling for the objective function takes fewer points.

The user has to write the subprograms by which the weighting function, specified function, approximating function and its derivatives with respect to the parameters are explicitly available. The information about the number of sample points forming the discrete point set, the starting point for the design parameters and the values of p should be supplied as data. Also the choice about quadratic interpolation, which optimization method is to be used, checking the gradients, the stopping criteria and the form of the results may be made. The optimal point, the value of the objective function, the weighted errors and execution time are printed out, and the intermediate results in the optimization procedure if desired.

There is no restriction on the number of design parameters and the sample points.

A recent publication [5] contains the background theory for the optimization algorithm, detailed organization of the program FMCLP and instructions on how to use it. This includes a block diagram of the package and a description of the algorithm for quadratic interpolation with a flowchart of the corresponding subroutine. The examples which demonstrate FMCLP were taken in numerical analysis and system modelling. Document NAPS ----- contains a complete listing and detailed user's manual for the given package fully illustrated with an example.

A few seconds of CDC 6400 computer time and a core requirement of about 14 K₁₀ is sufficient to optimize a five parameter design problem.

ACKNOWLEDGEMENT

Dr. C. Charalambous, who is now with the Department of Combinatorics and Optimization, University of Waterloo, Waterloo, Canada, and some of whose recent work is embodied in the package, is gratefully acknowledged.

REFERENCES

- [1] G.C. Temes and D.Y.F. Zai, "Least pth approximation", IEEE Trans. Circuit Theory, vol. CT-16, pp. 235-237, May 1969.
- [2] J.W. Bandler and C. Charalambous, "Practical least pth approximation with extremely large values of p", Conf. Rec. of the 5th Asilomar Conf. Circuits and Systems (Pacific Grove, Calif., Nov. 1971), pp. 66-70.

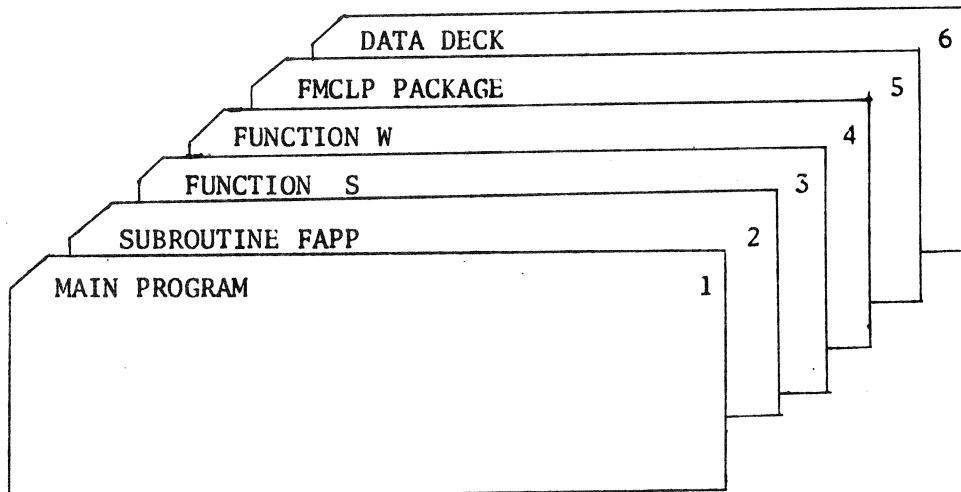
- [3] R. Fletcher and M.J.D. Powell, "A rapidly convergent descent method for minimization", Computer J., vol. 6, pp. 163-168, June 1963.
- [4] R. Fletcher, "A new approach to variable metric algorithms", Computer J., vol. 13, pp. 317-322, Aug. 1970.
- [5] J.R. Popović, J.W. Bandler and C. Charalambous, "General programs for least pth and near minimax approximation," Int. J. Systems Science, to be published.

USER'S MANUAL FOR FMCLP

J.R. POPOVIC and J.W. BANDLER
Department of Electrical Engineering
McMaster University
Hamilton, Ontario, Canada

Purpose To minimize the objective function of k variables \mathbf{x}
defined as a discrete least pth objective with a single
specification using gradient methods.

How to use Set the input deck as follows:



1. Main program

Write the main program as indicated below.

Dimension the following arrays

A(K), ASTRT(K), G(K), GRAD(K), Y(K), PY(K), DUM1(K), DUM2(K),
EPS(K), H(M), X(N2), ERROR(N2), IPA(ITER)

where

K is the number of variable parameters,

M = K(K+7)/2, N2 = N+2,

N is the discrete point set,

ITER is the maximum number of times the optimization method is used.

Call the subroutine FMCLP as follows:

```
CALL FMCLP (A, ASTRT, G, GRAD, Y, PY, DUM1, DUM2, EPS, H,
X, ERROR, IPA)
```

2. Subroutine FAPP

This subroutine calculates the approximating function and its gradients with respect to variable vector \tilde{a} .

Write subroutine FAPP as follows:

```
SUBROUTINE FAPP (X, K, A, APP, GRAD, INDIC)
```

```
DIMENSION A(1), GRAD(1)
```

where X, K, A and INDIC are input and APP and GRAD are output variables.

Compute the value of the approximating function $APP \triangleq F(A, X)$ where $A \triangleq [A_1 \ A_2 \ \dots \ A_K]^T$ and its gradients $GRAD(i) \triangleq \frac{\partial F(A, X)}{\partial A_i}$, $i = 1, 2, \dots, K$. The value of APP is already available at the time when the gradients are to be calculated.

INDIC may have values 1 or 2 and indicates whether the approximating function or its gradients should be calculated, respectively.

3. Function S

Function S is a subprogram of a single input variable X and defines a specified function

$$S = S(X).$$

4. Function W

Function W is also a subprogram of a single input variable X and defines a weighting function

$$W = w(X).$$

5. FMCLP package

A listing is appended to this manual.

6. Data deck

Parameters to be supplied as data are defined below:

| | |
|--------|---|
| N | The number of sample points forming the discrete point set. |
| XA, XB | The left and the right end points of the interval of the independent parameter. |
| NSUB | The number of subintervals over [XA, XB]. |
| IREAD | Integer which denotes whether or not the discrete set of points in [XA, XB] will be read. If IREAD=0 the discrete point set will be arranged equidistantly over the interval; If IREAD=1 the discrete point set will be read from data. |

| | |
|-------------------|--|
| X(I), I=1, N | The discrete point set over the interval. |
| K | The number of the independent variable parameters \sim . |
| ASTRT(I), I=1, K | Starting values for the K variable parameters. |
| IGRDCH | Gradients to be checked if IGRDCH=1; it should be set to 0 if gradients are not to be checked. |
| MET | Optimization method to be called: if MET=1 Fletcher method will be called; if MET=2 Fletcher-Powell method will be called. |
| MAX | Maximum number of permissible iterations. |
| ITER | Has already been defined in the main program as a length of the working array. |
| IPA(I), I=1, ITER | Vector containing the values of p for different least pth objectives. |
| IOPT | Denotes how many times the optimization is repeated with different starting points and/or different optimization techniques. |
| IPRINT | Intermediate output is printed out every IPRINT iterations it should be set to 0 if no intermediate output is desired. |
| IDATA | Input data is printed out if IDATA=1; it should be set to 0 if input data is not to be printed out. |

EST Minimum estimated value of the objective function.
EPS(I), I=1, K Small test quantities used by the Fletcher method.
EPS1 Small test quantity used by the Fletcher-Powell
 method.
DIF Small test quantity used by the subroutine FMCLP.

Setting up the data deck is illustrated in Table 1.

Recommended values for some of the parameters

NSUB = 5

MAX = 100

EPS(I), I=1, K, each 10^{-6}

DIF = 10^{-4}

EST A lower bound of the minimum of the objective function may be
 obtained from physical reasons. If the true minimum is not
 known, choose EST to be small enough (negative values are
 allowed). For approximation problems 0 is convenient.

Comments

Low values of p, e.g., 2, intermediately large values of p,
e.g., 10 to 1,000, as well as extremely large values of p, e.g.,
1,000,000 are optional to the user depending on how close to a
minimax (Chebyshev, equal-ripple) solution he wants to come. Low
values of p will generally allow quicker optimization to nonequal
ripple solutions. Large values of p may slow down optimization
but better near equal ripple solutions will be obtained.

TABLE 1

SETTING UP THE DATA DECK FOR FMCLP

| Conditions | Number of cards | Parameters | Type | Format |
|------------|-----------------------------|--|---------|--------|
| - | 1 | K, N, NSUB, IREAD, IOPT, ITER, IGRDCH | INTEGER | 7I10 |
| IREAD = 1 | As may as required by N | XA, XB X(I), I = 1, N EST, DIF | REAL | 2E16.8 |
| - | 1 | ASTRT(I), I = 1, K | REAL | 5E16.8 |
| - | As many as required by K | MET, MAX, IPRINT, IDATA | REAL | 2E16.8 |
| - | 1 | EPS(I), I = 1, K | REAL | 5E16.8 |
| MET = 1 | As many as required by K | EPS1 | REAL | 4I10 |
| MET = 2 | 1 | IPA(I), I = 1, ITER | INTEGER | 5E16.8 |
| - | As many as required by ITER | | | 8I10 |

↓ IOPT ↑ IREAD ↓

Recommendation: start with 2, increase to 10 then to 100, etc., as needed. Optimization for a larger value starts automatically at the optimum of the previous optimization unless otherwise specified. If a continuous minimax solution is desired, the values of p should be kept constant until the factor q. i., which indicates the number of quadratic interpolations, becomes zero.

The program terminates when stopping criteria for the Fletcher-Powell or Fletcher method are satisfied or when the relative change in the objective function in two successive iterations is less than a small prescribed quantity. If the gradients of the approximating function are not supplied correctly, the program will terminate and print out the appropriate message. Also, suitable diagnostic messages are printed out whenever there is any unusual exit.

The package FMCLP requires the CDC system routine SECOND which keeps track of elapsed time. For a different system the cards A90, A102 A110, A122, F16, F55, G21, G25 and G53 should be replaced by cards appropriate to the system or removed together with cards A104, A124, F56, G26 and G54.

Input-output Example

An example which shows how to set out the user's written subprograms and data deck is shown in Fig. 1. It corresponds to Example 2 in the paper [5]. Both optimization methods, Fletcher-Powell and Fletcher, are called.

Typical output of FMCLP for the example when p=2 and the Fletcher method is employed is shown in Fig. 2.

```
PROGRAM TST (TINPUT,OUTPUT,TAPE5=TINPUT,TAPE6=OUTPUT)
```

M A I N P R O G R A M

```
DIMENSION A(2), ASTRT(2), G(2), GRAD(2), Y(2), PY(2), DUM1(2),
1 DUM2(2), EPS(2), H(2), X(12), FRROR(12), TPA(4)
CALL FMCLP (A,ASTRT,G,GRAD,Y,PY,DUM1,DUM2,EPS,H,X,FRROR,TPA)
CALL EXIT
END
```

.....

```
SUBROUTINE FAPP(X,K,A,APP,GRAD,TNDIC)
```

SUBROUTINE WHICH CALCULATES APPROXIMATING
FUNCTION AND ITS GRADIENTS WITH RESPECT TO
MATERIAL PARAMETERS

```
DIMENSION A(1),GRAD(1)
100 DO(100,200),TNDIC
100 APP=A(1)*X+A(2)*EXP(X)
RETURN
200 GRAD(1)=X
GRAD(2)=EXP(X)
RETURN
END
```

.....

```
FUNCTION S (X)
```

FUNCTION SUBROUTINE WHICH DEFINES
SPECIFIED FUNCTION

```
S=X**2
RETURN
END
```

.....

```
FUNCTION W (X)
```

FUNCTION SUBROUTINE WHICH DEFINES
WEIGHTING FUNCTION

```
W=1.
RETURN
END
```

| | | | | | | |
|------|----|------|------|---|---|---|
| ? | 10 | 10 | 0 | ? | 4 | 1 |
| 2.0E | 00 | 2.0E | 00 | | | |
| 0.0E | 00 | 1.0E | -4 | | | |
| 1.0E | 00 | 1.0E | 00 | | | |
| 1 | 50 | 1 | 1 | | | |
| 1.0E | -6 | 1.0E | -6 | | | |
| 2 | 10 | 100 | 1000 | | | |
| 1.0E | 00 | 1.0E | 00 | | | |
| 2 | 50 | 1 | 1 | | | |
| 1.0E | -6 | 100 | 1000 | | | |
| 2 | 10 | | | | | |

30152B WORDS WERE REQUIRED FOR LOADING

Fig. 1 (continued)

| INITIAL SET OF INDEPENDENT VARIABLE | | ERRORS |
|-------------------------------------|-------------------|--------------------|
| 1 | 0. | 1.00000000000E+00 |
| 2 | 2.22222222222E-01 | 1.421688375175E+00 |
| 3 | 4.44444444444E-01 | 1.806537077854E+00 |
| 4 | 6.66666666667E-01 | 2.169956263277E+00 |
| 5 | 8.88888888889E-01 | 2.531190885386E+00 |
| 6 | 1.11111111111E+00 | 2.914274987394E+00 |
| 7 | 1.33333333333E+00 | 3.349223450239E+00 |
| 8 | 1.55555555556E+00 | 3.873520328779E+00 |
| 9 | 1.77777777778E+00 | 4.533977541282E+00 |
| 10 | 2.00000000000E+00 | 5.389056098931E+00 |

ABSOLUTE VALUE
OF MAXIMUM ERROR

5.389056098931E+00

GRADIENTS CHECKING

GRADIENTS HAVE BEEN CHECKED AT THE FOLLOWING POINT

A(1)= 1.00000000E+00
A(2)= 1.00000000E+00

| ANALYTICAL GRADIENTS | NUMERICAL GRADIENTS | PERCENTAGE ERROR |
|----------------------------------|----------------------------------|----------------------------------|
| 3.70668618E+00 1.21496244E+01 | 3.70668784E+00 1.21496381E+01 | 4.47340722E-05 1.12618381E-04 |

GRADIENTS ARE 0. K.

INPUT DATA

FOLLOWING METHODS HAVE BEEN CALLED

FLETCHER METHOD

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

MAXIMUM NUMBER OF ALLOWABLE ITERATIONS.....MAX= 50

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

STARTING VALUE FOR VECTOR A(I).....ASTRT(1)=1.00000000E+00
ASTRT(2)=1.00000000E+00

EST QUANTITIES TO BE USED IN FLETCHER METHOD.....EPS(1)=1.00000000F-06
EPS(2)=1.00000000F-06

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

OPTIMIZATION BY FLETCHER METHOD

| | | ITERATION NUMBER | | FUNCTION EVALUATIONS | TIME ELAPSED (SECONDS) | OBJECTIVE FUNCTION | VARIABLE VECTOR A(I) | GRADIENT VECTOR G(I) |
|---|----|------------------|--|----------------------|------------------------|--------------------|----------------------|----------------------|
| 0 | 1 | 0. | | 1.00853233E+01 | 1.00000000E+00 | 1.00000000E+00 | 3.70668618E+00 | 1.21496244E+01 |
| 1 | 3 | 3.00000000E-02 | | 1.32705502E+00 | 7.74269131E-01 | 2.60108589E-01 | 3.90319152E-01 | 7.4977444E-01 |
| 2 | 4 | 6.00000000E-02 | | 1.32337492E+00 | 7.63638437E-01 | 2.59767018E-01 | 2.66595399E-01 | 3.47296635E-01 |
| 3 | 5 | 6.00000000E-02 | | 1.29938470E+00 | 5.96531400E-01 | 3.03799899E-01 | 4.46527828E-03 | -4.06048809E-01 |
| 4 | 6 | 9.00000000E-02 | | 1.26580211E+00 | 4.28990916E-01 | 4.41758874E-01 | -2.31433534E-01 | -8.45275445E-01 |
| 5 | 7 | 9.00000000E-02 | | 1.26181664E+00 | 2.71904946E-02 | 4.77289071E-01 | -8.55595435E-02 | -2.87088864E-01 |
| 6 | 8 | 1.19000000E-01 | | 1.26146742E+00 | 1.92167215E-02 | 4.82042845E-01 | -2.96364828E-03 | -8.29817351E-03 |
| 7 | 9 | 1.19000000E-01 | | 1.26146669E+00 | 2.10849438E-02 | 4.81546088E-01 | -4.99745406E-05 | -8.28216978F-05 |
| 8 | 10 | 1.49000000E-01 | | 1.26146669E+00 | 2.11935745E-02 | 4.81513917E-01 | 8.44108900E-07 | 3.17448999E-06 |

IEXIT= 1 CRITERION FOR OPTIMUM HAS BEEN SATISFIED

Fig. 2 (continued)

FOLLOWING IS THE OPTIMUM SOLUTION

$$F = 1.26146669E+00$$

$$\begin{aligned} A(1) &= 2.11935745E-02 \\ A(2) &= 4.81513917E-01 \end{aligned}$$

NUMBER OF FUNCTION EVALUATIONS BY THE FLETCHER METHOD

10

EXECUTION TIME IN SECONDS .18600

$$P = 2$$

INDEPENDENT VARIABLE

ERRORS

| | | |
|----|---------------------|---------------------|
| 1 | 0. | 4.815139171854E-01 |
| 2 | 2.222222222222E-01 | 5.566650780618E-01 |
| 3 | 4.444444444444E-01 | 5.628689219208E-01 |
| 4 | 6.666666666667E-01 | 5.075456529761E-01 |
| 5 | 8.888888888889E-01 | 3.999619848678E-01 |
| 6 | 1.111111111111E+00 | 2.516906424447E-01 |
| 7 | 1.333333333333E+00 | 7.718421004554E-02 |
| 8 | 1.5555555555556E+00 | -1.055082187242E-01 |
| 9 | 1.7777777777778E+00 | -2.738460537287E-01 |
| 10 | 2.000000000000E+00 | -3.996795044703E-01 |

NUMBER OF Q.I.= 1

NEW SET OF INDEPENDENT VARIABLE

ERRORS

| | | |
|----|---------------------|---------------------|
| 1 | 0. | 4.815139171854E-01 |
| 2 | 3.534582105336E-01 | 5.682261989240E-01 |
| 3 | 4.444444444444E-01 | 5.628689219208E-01 |
| 4 | 6.666666666667E-01 | 5.075456529761E-01 |
| 5 | 8.888888888889E-01 | 3.999619848678E-01 |
| 6 | 1.111111111111E+00 | 2.516906424447E-01 |
| 7 | 1.333333333333E+00 | 7.718421004554E-02 |
| 8 | 1.5555555555556E+00 | -1.055082187242E-01 |
| 9 | 1.7777777777778E+00 | -2.738460537287E-01 |
| 10 | 2.000000000000E+00 | -3.996795044703E-01 |

ABSOLUTE VALUE
OF MAXIMUM ERROR

5.682261989240E-01

Fig. 2 (continued)

LISTING OF FMCLP

| | | |
|--|---|----|
| SUBROUTINE FMCLP (A,Xstrt,G,GRAD,Y,PY,DUM1,DUM2,eps,h,x,error,ipa) | A | 1 |
| C | A | 2 |
| C | A | 3 |
| C | A | 4 |
| C | A | 5 |
| SUBROUTINE WHICH COORDINATES THE OTHER | A | |
| SUBROUTINES IN THE PACKAGE FMCLP | A | |
| EXTERNAL FUNCT | A | 6 |
| EXTERNAL S,W | A | 7 |
| DIMENSION A(1), XSTR(1), G(1), Y(1), PY(1), DUM1(1), DUM2(1), EPS | A | 8 |
| I(1), H(1), GRAD(1), X(1), ERROR(1), IPA(1) | A | 9 |
| COMMON T1,K0,NFE | A | 10 |
| LOGICAL CONV,UNITH | A | 11 |
| ERR(Z)=WERR(Z,S,W,A,GRAD) | A | 12 |
| UNITH=.TRUE. | A | 13 |
| T1=0. | A | 14 |
| READ (5,42) N1,N,NPOD,IREAD,IOPT,ITER,IGRDCH | A | 15 |
| READ (5,44) XA,XB | A | 16 |
| XAB=XB-XA | A | 17 |
| NP=N+1 | A | 18 |
| IF (IREAD.EQ.0) IREAD=2 | A | 19 |
| GO TO (3,1), IREAD | A | 20 |
| 1 X(2)=XA | A | 21 |
| K=N-1 | A | 22 |
| DELTA=XAB/K | A | 23 |
| DO 2 I=1,K | A | 24 |
| X(I+2)=XA+I*DELTA | A | 25 |
| 2 CONTINUE | A | 26 |
| X(N+1)=XB | A | 27 |
| GO TO 4 | A | 28 |
| 3 READ (5,44) (X(I),I=2,NP) | A | 29 |
| 4 WRITE (6,52) | A | 30 |
| READ (5,44) EST,DIF | A | 31 |
| READ (5,44) (XSTR(I),I=1,N1) | A | 32 |
| DO 5 I=1,N1 | A | 33 |
| A(I)=XSTR(I) | A | 34 |
| 5 CONTINUE | A | 35 |
| DO 6 I=2,NP | A | 36 |
| II=I-1 | A | 37 |
| ERROR(I)=ERR(X(I)) | A | 38 |
| WRITE (6,53) II,X(I),ERROR(I) | A | 39 |
| 6 CONTINUE | A | 40 |
| IF (NPOD.LE.1) GO TO 9 | A | 41 |
| CALL NEWSET (XA,XB,NPOD,S,W,A,GRAD,N,X,ERROR,IQI) | A | 42 |
| WRITE (6,45) IQI | A | 43 |
| IF (IQI) 9,9,7 | A | 44 |
| 7 WRITE (6,47) | A | 45 |
| DO 8 I=2,NP | A | 46 |
| II=I-1 | A | 47 |
| ERROR(I)=ERR(X(I)) | A | 48 |
| WRITE (6,53) II,X(I),ERROR(I) | A | 49 |
| 8 CONTINUE | A | 50 |
| ERROR(1)=ERROR(2) | A | 51 |
| DO 10 I=3,NP | A | 52 |
| ERROR(1)=AMAX1(ERROR(1),ERROR(I)) | A | 53 |
| 10 CONTINUE | A | 54 |
| WRITE (6,50) | A | 55 |
| WRITE (6,51) | A | 56 |
| WRITE (6,49) ERROR(1) | A | 57 |
| C DATA FOR THE OPTIMALITY | A | 58 |
| C | A | 59 |

FOR THE OPTIMIZATION METHOD USED

```

      DO 41 K=1,IOPT          A  60
      KP=1                     A  61
      IF (K-1) 12,12,11        A  62
      READ (5,44) (XSTART(I),I=1,N1)   A  63
      11 READ (5,42) MFT,MAX,IPRINT,IData   A  64
      12 IF (MFT.EQ.1) READ (5,44) (EPS(I),I=1,N1)   A  65
      . IF (MFT.EQ.2) READ (5,44) EPS1   A  66
      READ (5,42) (IPA(I),I=1,ITER)    A  67
      DO 40 KK=1,ITER          A  68
      13
      14
      15
      16
      17
      18
      19
      20
      21
      22
      23
      24
      25
      26
      27
      28
      29
      30
      31
      32
      33
      34
      35
      36
      37
      38
      39
      40
      41
      42
      43
      44
      45
      46
      47
      48
      49
      50
      51
      52
      53
      54
      55
      56
      57
      58
      59
      60
      61
      62
      63
      64
      65
      66
      67
      68
      69
      70
      71
      72
      73
      74
      75
      76
      77
      78
      79
      80
      81
      82
      83
      84
      85
      86
      87
      88
      89
      90
      91
      92
      93
      94
      95
      96
      97
      98
      99
      100
      101
      102
      103
      104
      105
      106
      107
      108
      109
      110
      111
      112
      113
      114
      115
      116
      117
      118
      119
      120
      121
      122
      123
      124
      125
      126
      127
      128
      129
      130
      131
      132
      133
      134
      135
      136
      137
      138
      139
      140
      141
      142
      143
      144
      145
      146
      147
      148
      149
      150
      151
      152
      153
      154
      155
      156
      157
      158
      159
      160
      161
      162
      163
      164
      165
      166
      167
      168
      169
      170
      171
      172
      173
      174
      175
      176
      177
      178
      179
      180
      181
      182
      183
      184
      185
      186
      187
      188
      189
      190
      191
      192
      193
      194
      195
      196
      197
      198
      199
      200
      201
      202
      203
      204
      205
      206
      207
      208
      209
      210
      211
      212
      213
      214
      215
      216
      217
      218
      219
      220
      221
      222
      223
      224
      225
      226
      227
      228
      229
      230
      231
      232
      233
      234
      235
      236
      237
      238
      239
      240
      241
      242
      243
      244
      245
      246
      247
      248
      249
      250
      251
      252
      253
      254
      255
      256
      257
      258
      259
      260
      261
      262
      263
      264
      265
      266
      267
      268
      269
      270
      271
      272
      273
      274
      275
      276
      277
      278
      279
      280
      281
      282
      283
      284
      285
      286
      287
      288
      289
      290
      291
      292
      293
      294
      295
      296
      297
      298
      299
      300
      301
      302
      303
      304
      305
      306
      307
      308
      309
      310
      311
      312
      313
      314
      315
      316
      317
      318
      319
      320
      321
      322
      323
      324
      325
      326
      327
      328
      329
      330
      331
      332
      333
      334
      335
      336
      337
      338
      339
      340
      341
      342
      343
      344
      345
      346
      347
      348
      349
      350
      351
      352
      353
      354
      355
      356
      357
      358
      359
      360
      361
      362
      363
      364
      365
      366
      367
      368
      369
      370
      371
      372
      373
      374
      375
      376
      377
      378
      379
      380
      381
      382
      383
      384
      385
      386
      387
      388
      389
      390
      391
      392
      393
      394
      395
      396
      397
      398
      399
      400
      401
      402
      403
      404
      405
      406
      407
      408
      409
      410
      411
      412
      413
      414
      415
      416
      417
      418
      419
      420
      421
      422
      423
      424
      425
      426
      427
      428
      429
      430
      431
      432
      433
      434
      435
      436
      437
      438
      439
      440
      441
      442
      443
      444
      445
      446
      447
      448
      449
      450
      451
      452
      453
      454
      455
      456
      457
      458
      459
      460
      461
      462
      463
      464
      465
      466
      467
      468
      469
      470
      471
      472
      473
      474
      475
      476
      477
      478
      479
      480
      481
      482
      483
      484
      485
      486
      487
      488
      489
      490
      491
      492
      493
      494
      495
      496
      497
      498
      499
      500
      501
      502
      503
      504
      505
      506
      507
      508
      509
      510
      511
      512
      513
      514
      515
      516
      517
      518
      519
      520
      521
      522
      523
      524
      525
      526
      527
      528
      529
      530
      531
      532
      533
      534
      535
      536
      537
      538
      539
      540
      541
      542
      543
      544
      545
      546
      547
      548
      549
      550
      551
      552
      553
      554
      555
      556
      557
      558
      559
      560
      561
      562
      563
      564
      565
      566
      567
      568
      569
      570
      571
      572
      573
      574
      575
      576
      577
      578
      579
      580
      581
      582
      583
      584
      585
      586
      587
      588
      589
      590
      591
      592
      593
      594
      595
      596
      597
      598
      599
      600
      601
      602
      603
      604
      605
      606
      607
      608
      609
      610
      611
      612
      613
      614
      615
      616
      617
      618
      619
      620
      621
      622
      623
      624
      625
      626
      627
      628
      629
      630
      631
      632
      633
      634
      635
      636
      637
      638
      639
      640
      641
      642
      643
      644
      645
      646
      647
      648
      649
      650
      651
      652
      653
      654
      655
      656
      657
      658
      659
      660
      661
      662
      663
      664
      665
      666
      667
      668
      669
      670
      671
      672
      673
      674
      675
      676
      677
      678
      679
      680
      681
      682
      683
      684
      685
      686
      687
      688
      689
      690
      691
      692
      693
      694
      695
      696
      697
      698
      699
      700
      701
      702
      703
      704
      705
      706
      707
      708
      709
      710
      711
      712
      713
      714
      715
      716
      717
      718
      719
      720
      721
      722
      723
      724
      725
      726
      727
      728
      729
      730
      731
      732
      733
      734
      735
      736
      737
      738
      739
      740
      741
      742
      743
      744
      745
      746
      747
      748
      749
      750
      751
      752
      753
      754
      755
      756
      757
      758
      759
      7510
      7511
      7512
      7513
      7514
      7515
      7516
      7517
      7518
      7519
      7520
      7521
      7522
      7523
      7524
      7525
      7526
      7527
      7528
      7529
      7530
      7531
      7532
      7533
      7534
      7535
      7536
      7537
      7538
      7539
      75310
      75311
      75312
      75313
      75314
      75315
      75316
      75317
      75318
      75319
      75320
      75321
      75322
      75323
      75324
      75325
      75326
      75327
      75328
      75329
      75330
      75331
      75332
      75333
      75334
      75335
      75336
      75337
      75338
      75339
      75340
      75341
      75342
      75343
      75344
      75345
      75346
      75347
      75348
      75349
      75350
      75351
      75352
      75353
      75354
      75355
      75356
      75357
      75358
      75359
      75360
      75361
      75362
      75363
      75364
      75365
      75366
      75367
      75368
      75369
      75370
      75371
      75372
      75373
      75374
      75375
      75376
      75377
      75378
      75379
      75380
      75381
      75382
      75383
      75384
      75385
      75386
      75387
      75388
      75389
      75390
      75391
      75392
      75393
      75394
      75395
      75396
      75397
      75398
      75399
      753100
      753101
      753102
      753103
      753104
      753105
      753106
      753107
      753108
      753109
      753110
      753111
      753112
      753113
      753114
      753115
      753116
      753117
      753118
      753119
      753120
      753121
      753122
      753123
      753124
      753125
      753126
      753127
      753128
      753129
      753130
      753131
      753132
      753133
      753134
      753135
      753136
      753137
      753138
      753139
      753140
      753141
      753142
      753143
      753144
      753145
      753146
      753147
      753148
      753149
      753150
      753151
      753152
      753153
      753154
      753155
      753156
      753157
      753158
      753159
      753160
      753161
      753162
      753163
      753164
      753165
      753166
      753167
      753168
      753169
      753170
      753171
      753172
      753173
      753174
      753175
      753176
      753177
      753178
      753179
      753180
      753181
      753182
      753183
      753184
      753185
      753186
      753187
      753188
      753189
      753190
      753191
      753192
      753193
      753194
      753195
      753196
      753197
      753198
      753199
      753200
      753201
      753202
      753203
      753204
      753205
      753206
      753207
      753208
      753209
      753210
      753211
      753212
      753213
      753214
      753215
      753216
      753217
      753218
      753219
      753220
      753221
      753222
      753223
      753224
      753225
      753226
      753227
      753228
      753229
      753230
      753231
      753232
      753233
      753234
      753235
      753236
      753237
      753238
      753239
      753240
      753241
      753242
      753243
      753244
      753245
      753246
      753247
      753248
      753249
      753250
      753251
      753252
      753253
      753254
      753255
      753256
      753257
      753258
      753259
      753260
      753261
      753262
      753263
      753264
      753265
      753266
      753267
      753268
      753269
      753270
      753271
      753272
      753273
      753274
      753275
      753276
      753277
      753278
      753279
      753280
      753281
      753282
      753283
      753284
      753285
      753286
      753287
      753288
      753289
      753290
      753291
      753292
      753293
      753294
      753295
      753296
      753297
      753298
      753299
      753300
      753301
      753302
      753303
      753304
      753305
      753306
      753307
      753308
      753309
      753310
      753311
      753312
      753313
      753314
      753315
      753316
      753317
      753318
      753319
      753320
      753321
      753322
      753323
      753324
      753325
      753326
      753327
      753328
      753329
      753330
      753331
      753332
      753333
      753334
      753335
      753336
      753337
      753338
      753339
      753340
      753341
      753342
      753343
      753344
      753345
      753346
      753347
      753348
      753349
      753350
      753351
      753352
      753353
      753354
      753355
      753356
      753357
      753358
      753359
      753360
      753361
      753362
      753363
      753364
      753365
      753366
      753367
      753368
      753369
      753370
      753371
      753372
      753373
      753374
      753375
      753376
      753377
      753378
      753379
      753380
      753381
      753382
      753383
      753384
      753385
      753386
      753387
      753388
      753389
      753390
      753391
      753392
      753393
      753394
      753395
      753396
      753397
      753398
      753399
      753400
      753401
      753402
      753403
      753404
      753405
      753406
      753407
      753408
      753409
      753410
      753411
      753412
      753413
      753414
      753415
      753416
      753417
      753418
      753419
      753420
      753421
      753422
      753423
      753424
      753425
      753426
      753427
      753428
      753429
      753430
      753431
      753432
      753433
      753434
      753435
      753436
      753437
      753438
      753439
      753440
      753441
      753442
      753443
      753444
      753445
      753446
      753447
      753448
      753449
      753450
      753451
      753452
      753453
      753454
      753455
      753456
      753457
      753458
      753459
      753460
      753461
      753462
      753463
      753464
      753465
      753466
      753467
      753468
      753469
      753470
      753471
      753472
      753473
      753474
      753475
      753476
      753477
      753478
      753479
      753480
      753481
      753482
      753483
      753484
      753485
      753486
      753487
      753488
      753489
      753490
      753491
      753492
      753493
      753494
      753495
      753496
      753497
      753498
      753499
      753500
      753501
      753502
      753503
      753504
      753505
      753506
      753507
      753508
      753509
      753510
      753511
      753512
      753513
      753514
      753515
      753516
      753517
      753518
      753519
      753520
      753521
      753522
      753523
      753524
      753525
      753526
      753527
      753528
      753529
      753530
      753531
      753532
      753533
      753534
      753535
      753536
      753537
      753538
      753539
      753540
      753541
      753542
      753543
      753544
      753545
      753546
      753547
      753548
      753549
      753550
      753551
      753552
      753553
      753554
      753555
      753556
      753557
      753558
      753559
      753560
      753561
      753562
      753563
      753564
      753565
      753566
      753567
      753568
      753569
      753570
      753571
      753572
      753573
      753574
      753575
      753576
      753577
      753578
      753579
      753580
      753581
      753582
      753583
      753584
      753585
      753586
      753587
      753588
      753589
      753590
      753591
      753592
      753593
      753594
      753595
      753596
      753597
      753598
      753599
      753600
      753601
      753602
      753603
      753604
      753605
      753606
      753607
      753608
      753609
      753610
      753611
      753612
      753613
      753614
      753615
      753616
      753617
      753618
      753619
      753620
      753621
      753622
      753623
      753624
      753625
      753626
      753627
      753628
      753629
      753630
      753631
      753632
      753633
      753634
      753635
      753636
      753637
      753638
      753639
      753640
      753641
      753642
      753643
      753644
      753645
      753646
      753647
      753648
      753649
      753650
      753651
      753652
      753653
      753654
      753655
      753656
      753657
      753658
      753659
      753660
      753661
      753662
      753663
      753664
      753665
      753666
      753667
      753668
      753669
      753670
      753671
      753672
      753673
      753674
      753675
      753676
      753677
      753678
      753679
      753680
      753681
      753682
      753683
      753684
      753685
      753686
      753687
      753688
      753689
      753690
      753691
      753692
      753693
      753694
      753695
      753696
      753697
      753698
      753699
      753700
      753701
      753702
      753703
      753704
      753705
      753706
      753707
      753708
      753709
      753710
      753711
      753712
      753713
      753714
      753715
      753716
      753717
      753718
      753719
      753720
      753721
      753722
      753723
      753724
      753725
      753726
      753727
      753728
      753729
      753730
      753731
      753732
      753733
      753734
      753735
      753736
      753737
      753738
      753739
```

```

DO 27 I=1,N1                                A 119
DUM2(I)=A(I)                                A 120
27 CONTINUE                                  A 121
CALL SECOND (T2)                            A 122
CALL FINAL (A,F,N1,MET)                      A 123
T=T2-T1                                     A 124
IF (T1.EQ.0.) GO TO 28                      A 125
WRITE (6,43) T                                A 126
28 CONTINUE                                  A 127
29 INDEX=INDEX+1                            A 128
IF (M.EQ.1) GO TO 30                      A 129
GO TO 32                                     A 130
30 DO 31 I=1,N1                                A 131
A(I)=XSTRT(I)                                A 132
31 CONTINUE                                  A 133
32 CONTINUE                                  A 134
C
33 KR=0                                       A 135
WRITE (6,46) IP                                A 136
WRITE (6,48)                                A 137
DO 34 I=2,NP                                A 138
II=I-1                                      A 139
34 WRITE (6,53) II,X(I),ERROR(I)              A 140
CONTINUE                                  A 141
IF (NPOD.LE.1) GO TO 37                      A 142
CALL NEWSET (XA,XB,NPOD,S,W,A,GRAD,N,X,ERROR,IQI)
WRITE (6,45) IQI                                A 143
IF (IQI) 37,37,35                           A 144
35 WRITE (6,47)                                A 145
DO 36 I=2,NP                                A 146
II=I-1                                      A 147
36 ERROR(I)=ERR(X(I))                         A 148
WRITE (6,53) II,X(I),ERROR(I)              A 149
CONTINUF                                 A 150
37 ERROR(1)=ERROR(2)                          A 151
DO 38 I=3,NP                                A 152
38 ERROR(1)=AMAX1(ERROR(1),ERROR(I))          A 153
CONTINUE                                  A 154
WRITE (6,50)                                A 155
WRITE (6,51)                                A 156
39 WRITE (6,49) ERROR(1)                         A 157
IGRDCH=IGRDCH+2                            A 158
IF (KK-1) 40,40,39                           A 159
40 FTST=ABS((FF-F)/FF)                        A 160
IF (FTST.LT.DIF) GO TO 41                  A 161
41 CONTINUE                                  A 162
CONTINUE                                  A 163
RETURN                                     A 164
C
42 FORMAT (8I10)                                A 165
43 FORMAT (1H0,//25X,26HEXECUTION TIME IN SECONDS ,F10.5) A 166
44 FORMAT (5E16.8)                                A 167
45 FORMAT (1H1,14X,15HNUMBER OF Q.I.=,I3)        A 168
46 FORMAT (1H1,13X,4HP =,I7)                    A 169
47 FORMAT (/8X,31HNEW SET OF INDEPENDENT VARIABLE,15X,6HERRORS/) A 170
48 FORMAT (/13X,20HINDEPENDENT VARIABLE,21X,6HRRORS/)       A 171
49 FORMAT (13X,F20.12)                           A 172
50 FORMAT (//15X,14HABSOLUTE VALUE)             A 173

```

```

51  FORMAT (15X,16HOF MAXIMUM FRROR/)          A 178
52  FORMAT (1H1,8X,35HINITIAL SET OF INDEPENDENT VARIABLE,9X,6HERRORS/ A 179
1)   FORMAT (I9,1X,E23.12,10X,E23.12)          A 180
53  END                                         A 181
                                            A 182-

```

.....

```

FUNCTION WERR (Z,S,W,B,GRAD)           B 1
C
C      FUNCTION SUBPROGRAM WHICH CALCULATES    B 2
C      WEIGHTED FRROR FUNCTION                 B 3
C
C      EXTERNAL S,W                           B 4
C      DIMENSION B(1), GRAD(1)                B 5
C      CALL FAPP (Z,N1,B,APP,GRAD,1)          B 6
C      WERR=(APP-S(Z))*W(Z)                  B 7
C      RETURN                                     B 8
C      END                                       B 9
                                            B 10
                                            B 11-

```

.....

```

SUBROUTINE NEWSET (XA,XB,NPOD,S,W,B,GRAD,N,X,ERROR,IQI) C 1
C
C      SUBROUTINE WHICH CALCULATES THE NEW SET OF THE INDEPNDNT C 2
C      VARIABLES WHICH INCLUDE ALL THE EXTREMA OF THE WEIGHTED     C 3
C      FRROR FUNCTION                                         C 4
C
C      EXTERNAL S,W                           C 5
C      DIMENSION B(1), GRAD(1), X(1), ERROR(1) C 6
C      ERR(Z)=WFRR(Z,S,W,B,GRAD)            C 7
C      EPSN(Z)=ABS(WFRR(Z,S,W,B,GRAD))       C 8
C      IER=0                                     C 9
C      NN=N+1                                    C 10
C      NNN=N+2                                   C 11
C      X(1)=XA                                  C 12
C      X(NNN)=XB                                C 13
C      II=2                                     C 14
C      IQ=1                                     C 15
C      IND=1                                    C 16
C      IQI=0                                    C 17
C      DO 29 I=1,NN                            C 18
C      IF (X(I)-X(I+1)) 1,29,29              C 19
1      ZMIN=X(I+1)                          C 20
      ZMAX=X(I)                           C 21
      IQ=IND                         C 22
      IND=1                           C 23
      Z=ZMAX                         C 24
      EMAX=EPSN(Z)                      C 25
      EMIN=EPSN(ZMIN)                   C 26
      DELTA=(ZMIN-ZMAX)/NPOD           C 27
      NPOD1=NPOD+1                     C 28
      MM=1                           C 29

```

```

MIN=NPOD1 C 32
DO 7 K=1,NPOD C 33
ETRFN=EPSN(Z) C 34
IF (ETREN-EMAX) 6,6,2 C 35
2 EMAX=ETREN C 36
MM=K C 37
ZMAX=Z C 38
ZTEST1=ZMAX+DELTA C 39
FTFST1=EPSN(ZTEST1) C 40
IF (EMAX-ETEST1) 6,6,3 C 41
3 ZTEST2=ZMAX-DELTA C 42
IF (XA-(ZMAX-DELTA)) 4,4,6 C 43
4 ETTEST2=EPSN(ZTEST2) C 44
IF (EMAX-ETEST2) 6,6,5 C 45
5 IND=2 C 46
GO TO 8 C 47
6 Z=Z+DELTA C 48
7 CONTINUE C 49
C 50
8 IF (XA-(ZMAX-DELTA)) 9,9,13 C 51
9 IF (MM-NPOD) 10,10,13 C 52
10 GO TO (13,11), IND C 53
11 Q1=EPSN(ZMAX-DELTA) C 54
Q2=EPSN(ZMAX+DELTA) C 55
RA=(EMAX*2.-Q1-Q2)*2. C 56
IF (RA) 12,13,12 C 57
12 ZMAX=ZMAX+(Q2-Q1)*DELTA/RA C 58
IQI=IQI+1 C 59
ZMIN=ZTEST1 C 60
C 61
13 IF (I-1) 14,14,17 C 62
14 IF (ZMAX-ZMIN) 15,29,29 C 63
15 GO TO (29,16), IND C 64
16 X(1)=ZMAX C 65
XKP=X(2) C 66
X(2)=ZMIN C 67
GO TO 29 C 68
C 69
17 IF (X(I-1)-ZMAX) 18,21,21 C 70
18 IF (ZMAX-ZMIN) 19,21,21 C 71
19 GO TO (23,20), IND C 72
20 X(I)=ZMAX C 73
21 IF (X(I)-ZMIN) 22,25,25 C 74
22 XKP=X(I+1) C 75
X(I+1)=ZMIN C 76
GO TO 25 C 77
23 GO TO (25,24), IQ C 78
24 X(I)=XKP C 79
C 80
25 IF (I-NN) 29,26,29 C 81
26 IF (X(I)-X(I+1)) 27,29,29 C 82
27 IF (FPSN(X(I))-FPSN(X(I+1))) 28,29,29 C 83
28 II=II-1 C 84
29 CONTINUE C 85
IF (EPSN(X(2))-FPSN(X(1))) 30,31,31 C 86
30 II=II-3 C 87
C 88
31 IF (II-2) 32,50,32 C 89
C 90

```

```

32 IF (II+2) 42,33,42 C 91
33 JJ=NN/2+1 C 92
34 IF (N-2) 39,39,34 C 93
35 IF (EPSN(X(JJ-1))-EPSN(X(JJ+1))) 35,36,36 C 94
36 JJ=JJ-1 C 95
37 GO TO 37 C 96
38 JJ=JJ+1 C 97
39 IF (FPSN(X(JJ-1))-FPSN(X(JJ+1))) 38,39,39 C 98
40 JJ=JJ-1 C 99
41 DO 40 K=2,JJ C 100
42 I=JJ+2-K C 101
43 X(I)=X(I-1) C 102
44 CONTINUE C 103
45 JJJ=JJ+1 C 104
46 DO 41 K=JJJ,NN C 105
47 I=K+1 C 106
48 X(I-1)=X(I) C 107
49 CONTINUE C 108
50 C C 109
51 IF (II-1) 46,43,46 C 110
52 IF (EPSN(X(NNN))-EPSN(X(2))) 50,44,44 C 111
53 C C 112
54 DO 45 I=2,NN C 113
55 X(I)=X(I+1) C 114
56 CONTINUE C 115
57 IF (II+1) 50,47,50 C 116
58 IF (EPSN(X(1))-FPSN(X(NN))) 50,50,48 C 117
59 C C 118
60 DO 49 K=2,NN C 119
61 I=NN+2-K C 120
62 X(I)=X(I-1) C 121
63 CONTINUE C 122
64 CONTINUE C 123
65 RETURN C 124
66 END C 125-

```

SUBROUTINE FUNCT (N1,A,B,OB,L,G,CRAD,N,X,ERROR,IP)

D 1

SUBROUTINE WHICH SELECTS THE MAXIMUM ERROR
AND COMPUTES THE OBJECTIVE FUNCTION AND ITS
GRADIENTS W.R.T. THE VARIABLE PARAMETERS
IN THE LEAST P-TH SENSE

D 3

```

EXTERNAL S,W
DIMENSION B(1), G(1), GRAD(1), X(1), ERROR(1)
FRR(Z)=WFRR(Z,S,W,B,GRAD)
OBJP=0.
GRADP=0.
DO 1 K=1,N1
G(K)=0.
CONTINUE
NN=N+1
ERROR(1)=0.
DO 2 I=2,NN
ERROR(I)=ERR(X(I))

```

D 8

D 9

B 10
B 11

D 12

D 13

14
15

D 16

D 17

D 18
D 19

```

?    CONTINUE          D  20
DO 4 I=2,NN          D  21
IF (ABS(ERROR(I))-ERROR(1)) .GT. 4.E-3   D  22
3    ERROR(1)=ABS(ERROR(I))          D  23
CONTINUE          D  24
4    DO 6 I=2,NN          D  25
Z=X(I)              D  26
DEC=ERROR(I)/ERROR(1)      D  27
DEL=ABS(DEC)          D  28
OBJI=DEL**IP          D  29
GRADI=DEL**((IP-2)*DEC)  D  30
OBJP=OBJP+OBJI          D  31
CALL FAPP (Z,N1,B,APP,GRAD,2)  D  32
DO 5 K=1,N1          D  33
GRAD(K)=GRADI*W(Z)*GRAD(K)  D  34
G(K)=G(K)+GRAD(K)          D  35
CONTINUE          D  36
CONTINUE          D  37
PR=1./IP          D  38
OBJ=ERROR(1)*(OBJP**PR)  D  39
GRP=OBJP**((PR-1.))
DO 7 K=1,N1          D  40
G(K)=GRP*G(K)          D  41
CONTINUE          D  42
7    CONTINUE          D  43
RETURN             D  44
END                D  45-

```

.....

```

C SURROUTINE GRDCHK (N,A,G,PY,Y,GRAD,NP,XP,ERROR,IP,DUM1)
C
C     SUBROUTINE WHICH CHECKS THE GRADIENTS
C     W.R.T. ALL VARIABLE PARAMETERS
C
DIMENSION A(1), G(1), PY(1), Y(1), GRAD(1), XP(1), ERROR(1), DUM1(11)
1    CALL FUNCT (N,A,F,G,GRAD,NP,XP,ERROR,IP)
DO 3 I=1,N
IF (ABS(A(I)).LT.1.E-16) GO TO 1
2    DELX=1.E-4*A(I)
GO TO 2
1    DFLX=1.E-20
A(I)=A(I)+DFLX
2    CALL FUNCT (N,A,FNEW,PY,GRAD,NP,XP,ERROR,IP)
Y(I)=(FNEW-F)/DELX
DUM1(I)=Y(I)
A(I)=A(I)-DFLX
3    CONTINUE
DO 4 I=1,N
IF (ABS(Y(I)).LT.1.E-20) DUM1(I)=1.E-20
PY(I)=ABS((Y(I)-G(I))/DUM1(I))*100.
4    CONTINUE
WRITE (6,8)
WRITE (6,9)
WRITE (6,10) (I,A(I),I=1,N)
WRITE (6,11)

```

```

DO 5 I=1,N E 28
WRITE (6,12) G(I),Y(I),PY(I) E 29
CONTINUE E 30
5 DO 6 I=1,N E 31
IF (PY(I).GT.10.) GO TO 7 E 32
CONTINUE E 33
6 WRITE (6,13) E 34
RETURN E 35
7 WRITE (6,14) E 36
CALL EXIT E 37
C E 38
C E 39
FORMAT (1H1) E 40
FORMAT (1HO,5X,18HGRADIENTS CHECKING,/,,6X,18(1H-),//,6X,50HGRADIEN E 41
1 TS HAVE BEEN CHECKED AT THE FOLLOWING POINT//) E 42
FORMAT (10X,2HA(,I2,2H)=,E16.8) E 43
FORMAT (///,1HO,5X,20HANALYTICAL GRADIENTS,5X,19HNUMERICAL GRADIE E 44
1 NT,SX,16HPERCENTAGE ERROR,/ ) E 45
FORMAT (1HO,5X,3(E16.8,9X)) E 46
FORMAT (1HO,/,6X,19HGRADIENTS ARE O. K.) E 47
FORMAT (1HO,/,6X,64HYOUR PROGRAM HAS BEEN TERMINATED BECAUSE GRAD E 48
1 IENTS ARE INCORRECT,/6X,21HPLEASE CHECK IT AGAIN) E 49
END E 50-

```

```

SUBROUTINE FMNFC (N,X,F,G,H,UNITH,FEST,EPS,MAXFN,IPRINT,IFEXIT,GRAD
,NP,XP,ERROR,IP) F 1
PURPOSE F 2
TO FIND A LOCAL MINIMUM OF A FUNCTION OF SEVERAL VARIABLES F 3
ASSUMING THAT ITS GRADIENTS CAN BE CALCULATED EXPLICITLY F 4
BY THE METHOD OF FLETCHER F 5
THE METHOD IS DESCRIBED IN THE FOLLOWING ARTICLE F 6
R. FLETCHER, A NEW APPROACH TO VARIABLE METRIC ALGORITHMS, F 7
COMP. JOURNAL, VOL.13, 1970, PP.317-322. F 8
DIMENSION X(1), G(1), H(1), FPS(1), GRAD(1), XP(1), ERROR(1) F 9
LOGICAL CONV,UNITH F 10
COMMON T1,K0,NFNS F 11
CALL SFCOND (T3) F 12
K0=0 F 13
CALL FUNCT (N,X,F,G,GRAD,NP,XP,ERROR,IP) F 14
IF (F.LT.FEST) GO TO 23 F 15
NFNS=1 F 16
ITN=0 F 17
STEP=1. F 18
IDX=N F 19
IDG=N+N F 20
IH=IDG+N F 21
IF (.NOT.UNITH) GO TO 2 F 22
IJ=IH+1 F 23
DO 1 I=1,N F 24
DO 1 J=I,N F 25
H(IJ)=0. F 26

```

```

1 IF (I.EQ.J) H(IJ)=1.0 F 31
IJ=IJ+1 F 32
2 CONV=.TRUE. F 33
GDX=0. F 34
DO 6 I=1,N F 35
Z=0. F 36
IJ=IH+I F 37
IF (I.EQ.1) GO TO 4 F 38
II=I-1 F 39
DO 3 J=1,II F 40
Z=Z-H(IJ)*G(J) F 41
IJ=IJ+N-J F 42
3 CONTINUE F 43
4 DO 5 J=I,N F 44
Z=Z-H(IJ)*G(J) F 45
IJ=IJ+1 F 46
5 CONTINUE F 47
IF (ABS(Z).GT.FPS(I)) CONV=.FALSE. F 48
H(IDX+I)=Z F 49
GDX=GDX+G(I)*Z F 50
CONTINUE F 51
6 C F 52
IF (IPRINT.EQ.0) GO TO 7 F 53
IF (MOD(ITN,IPRINT).NE.0) GO TO 7 F 54
CALL SECOND (T4) F 55
TIME=T4-T3 F 56
CALL WRITE2 (X,N,G,F,NFNS,ITN,TIME) F 57
7 IEXIT=1 F 58
IF (CONV) GO TO 24 F 59
IEXIT=2 F 60
IF (GDX.GE.0.) GO TO 24 F 61
Z=1. F 62
IF (ITN.LT.N.AND.UNITH) Z=STEP F 63
W=2.*(FFST-F)/GDX F 64
IF (W.LT.Z) Z=W F 65
STEP=Z F 66
8 GDX=GDX*Z F 67
DO 9 I=1,N F 68
H(IDX+I)=H(IDX+I)*Z F 69
X(I)=X(I)+H(IDX+I) F 70
9 CONTINUF F 71
CALL FUNCT (N,X,FP,H,GRAD,NP,XP,FRROR,IP) F 72
IF (FP.LT.FFST) GO TO 23 F 73
NFNS=NFNS+1 F 74
IEXIT=3 F 75
IF (ITN.EQ.MAXFN) GO TO 24 F 76
GPDX=0. F 77
DO 10 I=1,N F 78
H(IDG+I)=H(I)-G(I) F 79
GPDX=GPDX+H(I)*H(IDX+I) F 80
10 CONTINUF F 81
DGDX=GPDX-GDX F 82
IF (F.GT.FP-.0001*GDX) GO TO 12 F 83
IEXIT=4 F 84
IF (GPDX.LT.0..AND.ITN.GT.N) GO TO 24 F 85
Z=3.*(F-FP)+GPDX+GDX F 86
W=SQRT(1.-GDX/Z*GPDX/Z)*ABS(Z) F 87
Z=1.-(GPDX+W-Z)/(DGDX+2.*W) F 88
IF (Z.LT.0.1) Z=0.1 F 89

```

```

DO 11 I=1,N          F  90
X(I)=X(I)-H(IDX+I) F  91
11 CONTINUE           F  92
GO TO 14             F  93
12 F=FP               F  94
DO 13 I=1,N          F  95
G(I)=H(I)            F  96
13 CONTINUE           F  97
IF (DGDX.GT.0.) GO TO 15 F  98
GDX=GPDX             F  99
Z=4.
14 STEP=Z*STEP        F 100
GO TO 8               F 101
15 IF (GPDX.LT.0.5*GDX) STEP=2.*STEP F 102
DGHDG=0.              F 103
DO 19 I=1,N          F 104
Z=0.                  F 105
IJ=IH+I               F 106
IF (I.EQ.1) GO TO 17 F 107
II=I-1                F 108
DO 16 J=1,II          F 109
Z=Z+H(IJ)*H(IDG+J)   F 110
IJ=IJ+N-J             F 111
CONTINUE              F 112
17 DO 18 J=I,N          F 113
Z=Z+H(IJ)*H(IDG+J)   F 114
IJ=IJ+1               F 115
CONTINUE              F 116
18 DGHDG=DGHDG+Z*H(IDG+I) F 117
H(I)=Z                F 118
19 CONTINUE              F 119
IF (DGHDG.LT.0.0) DGHDG=DGDX*0.01 F 120
IF (DGDX.LT.DGHDG) GO TO 21 F 121
W=1.0+DGHDG/DGDX      F 122
DO 20 I=1,N          F 123
H(IDX+I)=W*H(IDX+I)-H(I) F 124
20 CONTINUE              F 125
DGDX=DGDX+DGHDG       F 126
DGHDG=DGDX             F 127
21 IJ=IH               F 128
DO 22 I=1,N          F 129
W=H(IDX+I)/DGDX       F 130
Z=H(I)/DGHDG           F 131
DO 22 J=I,N          F 132
IJ=IJ+1               F 133
22 H(IJ)=H(IJ)+W*H(IDX+J)-Z*H(J) F 134
ITN=ITN+1              F 135
GO TO 2               F 136
23 IEXIT=5              F 137
24 IF (IEXIT.EQ.1) KO=1 F 138
IF (IPRINT.EQ.0) RETURN F 139
GO TO (25,26,27,26,28), IEXIT F 140
25 WRITE (6,30) IEXIT    F 141
GO TO 29               F 142
26 WRITE (6,31) IEXIT    F 143
GO TO 29               F 144
27 WRITE (6,32) IEXIT    F 145
GO TO 29               F 146
28 WRITE (6,33) IEXIT    F 147

```

| | | |
|----|---|--------|
| 29 | CONTINUF | F 149 |
| | RETURN | F 150 |
| C | | F 151 |
| C | | F 152 |
| 30 | FORMAT (/,1H0,6HIEEXIT=,I2,40HCRITERION FOR OPTIMUM HAS BEEN SATISF | F 153 |
| | IIED) | F 154 |
| 31 | FORMAT (/,1H0,6HIEEXIT=,I2,43HEITHER OF THE FOLLOWING THINGS HAS HA | F 155 |
| | 1PPENED,/,,9X,26H1. EPS CHOSEN IS TOO SMALL,/,,9X,28H2. GRADIENTS ARE | F 156 |
| | 2NOT CORRECT,/,,9X,25H3. MATRIX H GOES SINGULAR) | F 157 |
| 32 | FORMAT (/,1H0,6HIFXIT=,I2,55HMAXIMUM NUMBER OF ALLOWABLE ITERATION | F 158 |
| | 1 HAS BEEN EXCEEDED) | F 159 |
| 33 | FORMAT (/,1H0,6HIFXIT=,I2,60HFUNCTION VALUE LESS THAN MINIMUM ESTI | F 160 |
| | 1MATED HAS BEEN DETECTED) | F 161 |
| | END | F 162- |

.....

| | |
|---|------|
| SUBROUTINE FMFPC (FUNCT,N,X,F,G,EST,EPS,LIMITT,IER,H,IPRINT,GRAD,NP | G 1 |
| ,XP,ERROR,IP) | G 2 |
| PURPOSE | G 3 |
| TO FIND A LOCAL MINIMUM OF A FUNCTION OF SEVERAL VARIABLES | G 4 |
| ASSUMING THAT ITS GRADIENTS CAN BE CALCULATED EXPLICITLY | G 5 |
| BY THE METHOD OF FLETCHER AND POWELL | G 6 |
| THE METHOD IS DESCRIBED IN THE FOLLOWING ARTICLE | G 7 |
| R. FLETCHER AND M.J.D. POWELL, A RAPIDLY CONVERGENT | G 8 |
| DESCENT METHOD FOR MINIMIZATION, COMP. JOURNAL, | G 9 |
| VOL.6, 1963, PP.163-168. | G 10 |
| COMMON T1,K0,NUMF | G 11 |
| DIMENSIONED DUMMY VARIABLES | G 12 |
| DIMFNSION H(1), X(1), G(1), GRAD(1), XP(1), ERROR(1) | G 13 |
| COMPUTE FUNCTION VALUE AND GRADIENT VECTOR FOR INITIAL ARGUMENT | G 14 |
| K0=0 | G 15 |
| CALL SECOND (T3) | G 16 |
| CALL FUNCT (N,X,F,G,GRAD,NP,XP,ERROR,IP) | G 17 |
| KOUNT=0 | G 18 |
| NUMF=1 | G 19 |
| CALL SFCOND (T4) | G 20 |
| TIME=T4-T3 | G 21 |
| IF (IPRINT.EQ.0) GO TO 1 | G 22 |
| CALL WRITE2 (X,N,G,F,NUMF,KOUNT,TIME) | G 23 |
| CONTINUE | G 24 |
| REFSET ITERATION COUNTER AND GENERATE IDENTITY MATRIX | G 25 |
| IER=0 | G 26 |
| KK=0 | G 27 |
| N2=N+N | G 28 |
| N3=N2+N | G 29 |
| N31=N3+1 | G 30 |
| K=N31 | G 31 |
| DO 5 J=1,N | G 32 |
| H(K)=1. | G 33 |
| | G 34 |
| | G 35 |
| | G 36 |
| | G 37 |
| | G 38 |
| | G 39 |

```

NJ=N-J G 40
IF (NJ) 6,6,3 G 41
3 DO 4 L=1,NJ G 42
KL=K+L G 43
H(KL)=0. G 44
4 CONTINUE G 45
K=KL+1 G 46
5 CONTINUE G 47
C G 48
C START ITERATION LOOP G 49
6 IF (KOUNT.EQ.0) GO TO 7 G 50
IF (KK.NE.IPRINT) GO TO 7 G 51
KK=0 G 52
CALL SECOND (T4) G 53
TIME=T4-T3 G 54
CALL WRITE2 (X,N,G,F,NUMF,KOUNT,TIME) G 55
7 CONTINUE G 56
KOUNT=KOUNT+1 G 57
KK=KK+1 G 58
C G 59
C SAVE FUNCTION VALUE, ARGUMENT VECTOR AND GRADIENT VECTOR G 60
OLDF=F G 61
DO 11 J=1,N G 62
K=N+J G 63
H(K)=G(J) G 64
K=K+N G 65
H(K)=X(J) G 66
C G 67
C DETERMINE DIRECTION VECTOR H G 68
K=J+N3 G 69
T=0. G 70
DO 10 L=1,N G 71
T=T-G(L)*H(K) G 72
IF (L-J) 8,9,9 G 73
8 K=K+N-L G 74
GO TO 10 G 75
9 K=K+1 G 76
10 CONTINUE G 77
H(J)=T G 78
11 CONTINUE G 79
C G 80
C CHECK WHETHER FUNCTION WILL DECREASE STEPPING ALONG H. G 81
DY=0. G 82
HNRM=0. G 83
GNRM=0. G 84
C G 85
C CALCULATE DIRECTIONAL DERIVATIVE AND TESTVALUES FOR DIRECTION G 86
VECTOR H AND GRADIENT VECTOR G. G 87
DO 12 J=1,N G 88
HNRM=HNRM+ABS(H(J)) G 89
GNRM=GNRM+ABS(G(J)) G 90
DY=DY+H(J)*G(J) G 91
CONTINUE G 92
C G 93
C REPEAT SEARCH IN DIRECTION OF STEEPEST DESCENT IF DIRECTION G 94
C DERIVATIVE APPEARS TO BE POSITIVE OR ZERO. G 95
IF (DY) 13,57,57 G 96
C G 97
C REPEAT SEARCH IN DIRECTION OF STEEPEST DESCENT IF DIRECTION G 98

```

```

C     VECTOR H IS SMALL COMPARED TO GRADIENT VECTOR G.          G  99
13    IF (HNRM/GNRM-FPS) 57,57,14                                G 100
C     SEARCH MINIMUM ALONG DIRECTION H                           G 101
C
C     SEARCH ALONG H FOR POSITIVE DIRECTIONAL DERIVATIVE       G 102
14    FY=F                                                       G 103
      ALFA=2.*(EST-F)/DY                                         G 104
      AMBDA=1.                                                       G 105
C
C     USE ESTIMATE FOR STEPSIZE ONLY IF IT IS POSITIVE AND LESS THAN G 106
C     1. OTHERWISE TAKE 1. AS STEPSIZE                           G 107
      IF (ALFA) 17,17,15                                         G 108
      IF (ALFA-AMBDA) 16,17,17                                     G 109
      AMBDA=ALFA                                                 G 110
      ALFA=0.                                                       G 111
C
C     SAVE FUNCTION AND DERIVATIVE VALUES FOR OLD ARGUMENT      G 112
18    FX=FY                                                       G 113
      DX=DY                                                       G 114
C
C     STEP ARGUMENT ALONG H                                     G 115
      DO 19 I=1,N                                               G 116
      X(I)=X(I)+AMBDA*H(I)                                       G 117
19    CONTINUE                                                   G 118
C
C     COMPUTE FUNCTION VALUE AND GRADIENT FOR NEW ARGUMENT      G 119
      CALL FUNCT (N,X,F,G,GRAD,NP,XP,ERROR,IP)
      NUMF=NUMF+1
      FY=F
C
C     COMPUTE DIRECTIONAL DERIVATIVE DY FOR NEW ARGUMENT. TERMINATE G 120
C     SEARCH, IF DY IS POSITIVE. IF DY IS ZERO THE MINIMUM IS FOUND G 121
      DY=0.
      DO 20 I=1,N                                               G 122
      DY=DY+G(I)*H(I)
20    CONTINUF
      IF (DY) 21,41,24
C
C     TERMINATE SEARCH ALSO IF THE FUNCTION VALUE INDICATES THAT G 123
C     A MINIMUM HAS BEEN PASSED                                 G 124
      IF (FY-FX) 22,24,24
C
C     REPEAT SEARCH AND DOUBLE STEPSIZE FOR FURTHER SEARCHES   G 125
      AMBDA=AMBDA+ALFA
      ALFA=AMBDA
C     END OF SEARCH LOOP                                      G 126
C
C     TERMINATE IF THE CHANGE IN ARGUMENT GETS VERY LARGE      G 127
      IF (HNRM*AMBDA-1.E10) 18,18,23
C
C     LINNAR SEARCH TECHNIQUE INDICATES THAT NO MINIMUM EXISTS G 128
      IER=2
      GO TO 62
C
C     INTERPOLATE CUBICALLY IN THE INTERVAL DEFINED BY THE SEARCH G 129
C     ABOVE AND COMPUTE THE ARGUMENT X FOR WHICH THE INTERPOLATION G 130
C     POLYNOMIAL IS MINIMIZED                                  G 131
24    T=0.

```

```

25 IF (AMBDA) 26,41,26 G 158
26 Z=3.* (FX-FY)/AMBDA+DX+DY G 159
ALFA=AMAX1(ABS(Z),ABS(DX),ABS(DY)) G 160
DALFA=Z/ALFA G 161
DALFA=DALFA*DALFA-DX/ALFA*DY/ALFA G 162
IF (DALFA) 57,27,27 G 163
27 W=ALFA*SQRT(DALFA) G 164
ALFA=DY-DX+W+W G 165
IF (ALFA) 28,29,28 G 166
28 ALFA=(DY-Z+W)/ALFA G 167
GO TO 30 G 168
29 ALFA=(Z+DY-W)/(Z+DX+Z+DY) G 169
30 ALFA=ALFA*AMBDA G 170
DO 31 I=1,N G 171
X(I)=X(I)+(T-ALFA)*H(I) G 172
CONTINUE G 173
C G 174
C TERMINATE, IF THE VALUE OF THE ACTUAL FUNCTION AT X IS LESS G 175
C THAN THE FUNCTION VALUES AT THE INTERVAL ENDS. OTHERWISE REDUCE G 176
C THE INTERVAL BY CHOOSING ONE END-POINT EQUAL TO X AND REPEAT G 177
C THE INTERPOLATION. WHICH END-POINT IS CHOOSEN DEPENDS ON THE G 178
C VALUE OF THE FUNCTION AND ITS GRADIENT AT X G 179
C G 180
NUMF=NUMF+1 G 181
CALL FUNCT (N,X,F,G,GRAD,NP,XP,ERROR,IP) G 182
IF (F-FX) 32,32,33 G 183
32 IF (F-FY) 41,41,33 G 184
33 DALFA=0. G 185
DO 34 I=1,N G 186
DALFA=DALFA+G(I)*H(I) G 187
34 CONTINUE G 188
IF (DALFA) 35,38,38 G 189
35 IF (F-FX) 37,36,38 G 190
36 IF (DX-DALFA) 37,41,37 G 191
37 FX=F G 192
DX=DALFA G 193
T=ALFA G 194
AMBDA=ALFA G 195
GO TO 25 G 196
38 IF (FY-F) 40,39,40 G 197
39 IF (DY-DALFA) 40,41,40 G 198
40 FY=F G 199
DY=DALFA G 200
AMBDA=AMBDA-ALFA G 201
GO TO 24 G 202
C G 203
C TERMINATE, IF FUNCTION HAS NOT DECREASED DURING LAST ITERATION G 204
41 IF (OLDF-F+EPS) 57,42,42 G 205
C G 206
C COMPUTE DIFFERENCE VECTORS OF ARGUMENT AND GRADIENT FROM G 207
C TWO CONSECUTIVE ITERATIONS G 208
42 DO 43 J=1,N G 209
K=N+J G 210
H(K)=G(J)-H(K) G 211
K=N+K G 212
H(K)=X(J)-H(K) G 213
43 CONTINUE G 214
C G 215
C TEST LENGTH OF ARGUMENT DIFFERENCE VECTOR AND DIRECTION VECTOR G 216

```

```

C IF AT LEAST N ITERATIONS HAVE BEEN EXECUTED. TERMINATE, IF G 217
C BOTH ARE LESS THAN FPS G 218
IFR=0 G 219
IF (KOUNT-N) 47,44,44 G 220
44 T=0. G 221
DO 45 J=1,N G 222
K=N+J G 223
W=H(K) G 224
K=K+N G 225
T=T+ABS(H(K)) G 226
45 CONTINUE G 227
IF (HNRM-EPS) 46,46,47 G 228
46 IF (T-EPS) 62,62,47 G 229
C C TERMINATE, IF NUMBER OF ITERATIONS WOULD EXCEED LIMIT G 230
47 IF (KOUNT-LIMIT) 48,55,55 G 231
C C PREPARE UPDATING OF MATRIX H G 232
48 ALFA=0. G 233
Z=0. G 234
DO 52 J=1,N G 235
K=J+N3 G 236
W=0. G 237
DO 51 L=1,N G 238
KL=N+L G 239
W=W+H(KL)*H(K) G 240
IF (L-J) 49,50,50 G 241
49 K=K+N-L G 242
GO TO 51 G 243
50 K=K+1 G 244
51 CONTINUE G 245
K=N+J G 246
KN=K+N G 247
Z=Z+H(K)*H(KN) G 248
ALFA=ALFA+W*H(K) G 249
52 H(J)=W G 250
CONTINUE G 251
C REPEAT SEARCH IN DIRECTION OF STEEPEST DESCENT IF RESULTS G 252
C ARE NOT SATISFACTORY G 253
IF (Z*ALFA) 53,2,53 G 254
C C UPDATE MATRIX H G 255
53 K=N31 G 256
DO 54 L=1,N G 257
KL=N2+L G 258
DO 54 J=L,N G 259
NJ=N2+J G 260
H(K)=H(K)+H(KL)*H(NJ)/Z-H(L)*H(J)/ALFA G 261
54 K=K+1 G 262
GO TO 6 G 263
C END OF ITERATION LOOP G 264
C C NO CONVERGENCE AFTER LIMIT ITERATIONS G 265
55 IER=1 G 266
IF (KK.NE.IPRINT) GO TO 56 G 267
CALL WRITE2 (X,N,G,F,NUMF,KOUNT,TIME) G 268
56 CONTINUE G 269
GO TO 62 G 270

```

G 276

G 277

G 278

G 279

G 280

G 281

G 282

G 283

G 284

G 285

G 286

G 287

G 288

G 289

G 290

G 291

G 292

G 293

G 294

G 295

G 296

G 297

G 298

G 299

G 300

G 301

G 302

G 303

G 304

G 305

G 306

G 307

G 308

G 309

G 310

G 311

G 312

G 313

G 314

G 315

G 316-

H 1

H 2

H 3

H 4

H 5

H 6

H 7

H 8

H 9

H 10

H 11

H 12

C RESTORE OLD VALUES OF FUNCTION AND ARGUMENTS
 57 DO 58 J=1,N
 K=N2+J
 X(J)=H(K)
 CONTINUE
 CALL FUNCT (N,X,F,G,GRAD,NP,XP,ERROR,IP)
 NUMF=NUMF+1
 C REPEAT SEARCH IN DIRECTION OF STEEPEST DESCENT IF DERIVATIVE
 C FAILS TO BE SUFFICIENTLY SMALL
 IF (GNRM-EPS) 61,61,59
 C TEST FOR REPEATED FAILURE OF ITERATION
 59 IF (IER) 62,60,60
 60 IER=-1
 GO TO 2
 61 IER=0
 62 II=IER+2
 IF (II.EQ.2) KO=1
 IF (IPRINT.EQ.0) RETURN
 GO TO (63,64,65,66), II
 63 WRITE (6,68) IER
 GO TO 67
 64 WRITE (6,69) IER
 GO TO 67
 65 WRITE (6,70) IER
 GO TO 67
 66 WRITE (6,71) IER
 67 RETURN
 C
 C
 C
 68 FORMAT (1HO,4HIER=,I2,32H ERROR IN GRADIENTS CALCULATIONS)
 69 FORMAT (1HO,4HIER=,I2,41H CRITERION FOR OPTIMUM HAS BEEN SATISFIED
 1)
 70 FORMAT (1HO,4HIER=,I2,57H MAXIMUM NUMBER OF ALLOWABLE ITERATIONS H
 1AS BEEN EXCEEDED)
 71 FORMAT (1HO,4HIER=,I2,83H CHANGE IN ARGUMENTS GETS TOO LARGE, LINE
 1AR SEARCH INDICATES THAT NO MINIMUM EXISTS)
 END

.....
 SURROUNING INPUT (MFT,M,MAX,N,IPRINT,IData,FPS1,FST,EPS,ASTRT)

H 1

H 2

H 3

H 4

H 5

H 6

H 7

H 8

H 9

H 10

H 11

H 12

C
 DIMENSION ASTRT(1), FPS(1)
 WRITE (6,5)
 IF (MFT.NE.1.AND.MFT.NE.2) GO TO 4
 INDEX=0
 GO TO (1,2), MFT
 WRITE (6,6)
 GO TO 3

```

2      WRITE (6,7)                                H 13
3      CONTINUE
4      WRITE (6,8) N                                H 14
5      WRITE (6,9) MAX                             H 15
6      WRITE (6,10) IPRINT                         H 16
7      WRITE (6,11) ASTRT(1)                      H 17
8      WRITE (6,12) (I,ASTRT(I),I=2,N)            H 18
9      IF (MET.EQ.1) WRITE (6,13) EPS(1)          H 19
10     IF (MET.EQ.1) WRITE (6,14) (I,EPS(I),I=2,N) H 20
11     IF (MET.EQ.2) WRITE (6,15) EPS1             H 21
12     WRITE (6,16) FST                            H 22
13     RETURN                                         H 23
14     WRITE (6,17)
15     CALL EXIT                                     H 24
C
C
16     FORMAT (1H1,10HINPUT DATA,/,,1X,10(1H-),//,,1X,34HFOLLOWING METHODS H 25
17     HAVE BEEN CALLED,/,,1X,34HFOLLOWING METHODS H 26
18     FORMAT (1HO,15HFLETCHER METHOD)           H 27
19     FORMAT (1HO,22HFLETCHER-POWELL METHOD)    H 28
20     FORMAT (1HO,/1X,31HNUMBER OF INDEPENDENT VARIABLES,36(1H.),2HN=,I5 H 29
21     ,/)                                         H 30
22     FORMAT (1HO,38HMAXIMUM NUMBER OF ALLOWABLE ITERATIONS,27(1H.),4HMA H 31
23     IX=,I5,/,,1X,34HFOLLOWING METHODS H 32
24     FORMAT (1HO,57HINTERMEDIATE OUTPUT TO BE PRINTED EVERY IPRINT ITER H 33
25     IATIONS,5(1H.),7HIPRINT=,I5,/,,1X,34HFOLLOWING METHODS H 34
26     FORMAT (1HO,30HSTARTING VALUE FOR VECTOR A(1),29(1H.),10HASTR( 1) H 35
27     1=,E16.8)                                    H 36
28     FORMAT (1HO,59X,6HASTR(,I2,2H)=,F16.8)      H 37
29     FORMAT (1HO,/,,1X,45HTEST QUANTITIES TO BE USED IN FLETCHER METHOD, H 38
30     116(1H.),8HEPS( 1)=,E16.8)                 H 39
31     FORMAT (1HO,61X,4HEPS(,I2,2H)=,F16.8)      H 40
32     FORMAT (1HO,/,,1X,50HTEST QUANTITY TO BE USED IN FLETCHER-POWELL ME H 41
33     THOD,14(1H.),5HFPS1=,E16.8)                H 42
34     FORMAT (1HO,/,,1X,51HESTIMATE OF LOWER BOUND ON FUNCTION TO BE MINI H 43
35     1MTZFD,14(1H.),4HFST=,F16.8)               H 44
36     FORMAT (1HO,49HNONE OF THE OPTIMIZATION METHODS HAVE BEEN CALLED,/ H 45
37     ,1X,29HPLEASE CHECK THE VALUE OF MET,/,,1X,9HREMAINDER,/,,1X,40HMET= H 46
38     21      FLETCHER METHOD WOULD BE CALLED,/,,1X,47HMET=2      FLETCHER-POW H 47
39     3ELL METHOD WOULD BE CALLED)                H 48
40     END                                           H 49
41
42
43
44
45
46
47
48
49
50
51
52
53-

```

```

C
C
54    SUBROUTINE FINAL (A,F,N,MET)
55
56    PRINTS THE RESULTS
57    FOR THE OPTIMIZATION PROCESS
58
59    COMMON T1,K0,NFF
60    DIMENSION A(1)
61    WRITE (6,5)
62    IF (K0.EQ.0) GO TO 1
63    WRITE (6,6)
64    GO TO 2
65    WRITE (6,7)

```

| | |
|---|----|
| I | 1 |
| I | 2 |
| I | 3 |
| I | 4 |
| I | 5 |
| I | 6 |
| I | 7 |
| I | 8 |
| I | 9 |
| I | 10 |
| I | 11 |
| I | 12 |

```

2 CONTINUE I 13
  WRITE (6,8) F I 14
  WRITE (6,9) (I,A(I),I=1,N)
  GO TO (3,4), MET I 15
3 WRITE (6,10) NFE I 16
  RETURN I 17
4 WRITE (6,11) NFF I 18
  RETURN I 19
C I 20
C I 21
C I 22
C I 23
5 FORMAT (1H1) I 24
6 FORMAT (41X,33HFOLLOWING IS THE OPTIMUM SOLUTION,/41X,33(1H-)) I 25
7 FORMAT (45X,25HRESULTS AT LAST ITERATION/,45X,25(1H-)) I 26
8 FORMAT (//,48X,3HF =,F16.8,/) I 27
9 FORMAT (45X,2HA(,I2,2H)=,E16.8) I 28
10 FORMAT (//25X,53HNUMBER OF FUNCTION EVALUATIONS BY THE FLETCHER ME I 29
    THOD,I10) I 30
11 FORMAT (//25X,60HNUMBER OF FUNCTION EVALUATIONS BY THE FLETCHER-PO I 31
    WELL METHOD,I10) I 32
  END I 33-

```

```

SUBROUTINE WRITE1 (N) J 1
C J 2
C PRINTS THE INTERMEDIATE RESULTS J 3
C J 4
COMMON TIME,K0,NFE J 5
  WRITE (6,5)
  GO TO (1,2), N J 6
1  WRITE (6,6)
  GO TO 3 J 7
2  WRITE (6,7) J 8
3  CONTINUE J 9
  IF (TIME.EQ.0.) GO TO 4 J 10
  WRITE (6,8) J 11
  RETURN J 12
4  WRITE (6,9) J 13
  RETURN J 14
C J 15
C J 16
C J 17
5 FORMAT (1H1) J 18
6 FORMAT (1HO,31HOPTIMIZATION BY FLEETCHER METHOD,/1HO,31(1H-)) J 19
7 FORMAT (1HO,38HOPTIMIZATION BY FLETCHER-POWELL METHOD,/1HO,38(1H-)) J 20
11 J 21
8 FORMAT (1HO,9HITERATION,2X,8HFUNCTION,6X,12HTIME ELAPSED,8X,9HOBJE J 22
    JCTIVE,14X,20H VARIABLE VECTOR A(I),9X,20HGRADIENT VECTOR G(I),/1HO, J 23
    26HNUMBFR,5X,11HEVALUATIONS,3X,9H(SECONDS),11X,8HFUNCTION,/) J 24
9  FORMAT (1HO,9HITERATION,2X,8HFUNCTION,8X,9HOBJFCTIVE,14X,20HVARIAJ J 25
    RILF VECTO A(I),9X,20HGRADIENT VECTO G(I),/1HO,6HNUMBFR,5X,11HFVAL J 26
    EUATIONS,5X,8HFUNCTION,/) J 27
  END J 28

```

```

C
C
C
SURROUTINE WRITE2 (A,N,G,F,NUMF,ITER,TIME)
      PRINTS THE INTERMEDIATE RESULTS
C
COMMON T1,K0,NFE
DIMENSION A(1), G(1)
IF (T1.EQ.0.) GO TO 1
WRITE (6,2) ITER,NUMF,TIME,F,((A(I),G(I)),I=1,N)
RETURN
1  WRITE (6,3) ITER,NUMF,F,((A(I),G(I)),I=1,N)
RETURN
C
C
2  FORMAT (1H0,I5,7X,I5,5X,E16.8,3X,E16.8,12X,95(E16.8,13X,E16.8,,70
   IX))
3  FORMAT (1H0,I5,7X,I5,8X,F16.8,7X,95(E16.8,13X,F16.8,,49X))
END
      K    1
      K    2
      K    3
      K    4
      K    5
      K    6
      K    7
      K    8
      K    9
      K   10
      K   11
      K   12
      K   13
      K   14
      K   15
      K   16
      K   17-

```