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 abount 14 K_{10} 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.

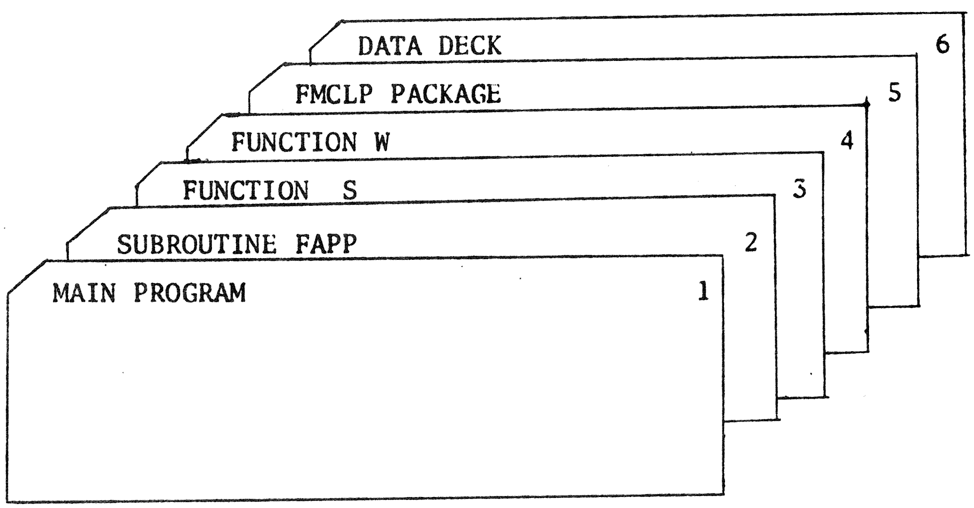
[5] J.R. Popović, J.W. Bandler and C. Charalambous, "General programs for least pth and near minimax approximation," Int. J. Systems Science, to be published.

USER'S MANUAL FOR FMCLP

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

<u>Purpose</u> To minimize the objective function of k variables $\underset{\sim}{a}$ defined as a discrete least pth objective with a single specification using gradient methods.

<u>How to use</u> 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 $\underset{\sim}{a}$.

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 $APP \triangleq F(\underset{\sim}{A},X)$ where $\underset{\sim}{A} \triangleq [A_1 \;\; A_2 \;\; \ldots \;\; A_K]^T$ and its gradients $GRAD(i) \triangleq \frac{\partial F(\underset{\sim}{A},X)}{\partial A_i}$, $i = 1, 2, \ldots, K$. 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 $\underset{\sim}{a}$.
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
-	1	XA, XB	REAL	2E16.8
IREAD = 1	As may as required by N	X(I), I = 1, N	REAL	5E16.8
-	1	EST, DIF	REAL	2E16.8
-	As many as required by K	ASTRT(I), I = 1, K	REAL	5E16.8
-	1	MET, MAX, IPRINT, IDATA	INTEGER	4I10
MET = 1	As many as required by K	EPS(I), I = 1, K	REAL	5E16.8
MET = 2	1	EPS1	REAL	5E16.8
-	As many as required by ITER	IPA(I), I = 1, ITER	INTEGER	8I10

|← IOPT times →|

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 (INPUT,OUTPUT,TAPE5=INPUT,TAPE6=OUTPUT)
C
C                     M A I N   P R O G R A M
C
      DIMENSION A(2), ASTRT(2), G(2), GRAD(2), Y(2), PY(2), DUM1(2),
     1 DUM2(2), EPS(2), H(9), X(12), ERROR(12), IPA(4)
      CALL EMCLP (A,ASTRT,G,GRAD,Y,PY,DUM1,DUM2,EPS,H,X,ERROR,IPA)
      CALL EXIT
      END
```

....

```
      SUBROUTINE FAPP(X,K,A,APP,GRAD,INDIC)
C
C         SUBROUTINE WHICH CALCULATES APPROXIMATING
C         FUNCTION AND ITS GRADIENTS WITH RESPECT TO
C         VARIABLE PARAMETERS
C
      DIMENSION A(1),GRAD(1)
      GO TO(100,200),INDIC
  100 APP=A(1)*X+A(2)*EXP(X)
      RETURN
  200 GRAD(1)=X
      GRAD(2)=EXP(X)
      RETURN
      END
```

....

```
      FUNCTION S (X)
C
C         FUNCTION SUBROUTINE WHICH DEFINES
C         SPECIFIED FUNCTION
C
      S=X**2
      RETURN
      END
```

....

```
      FUNCTION W (X)
C
C         FUNCTION SUBROUTINE WHICH DEFINES
C         WEIGHTING FUNCTION
C
      W=1.
      RETURN
      END
```

Fig. 1

```
      2          10          10          0          2          4          1
[illegible].0E   00        2.0E     00
[illegible].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
      2          10         100       1000
```

30152B WORDS WERE REQUIRED FOR LOADING

Fig. 1 (continued)

	INITIAL SET OF INDEPENDENT VARIABLE	ERRORS
1	0.	1.000000000000E+00
2	2.222222222222E-01	1.421688375175E+00
3	4.444444444444E-01	1.806537077854E+00
4	6.666666666667E-01	2.169956263277E+00
5	8.888888888889E-01	2.531190886386E+00
6	1.111111111111E+00	2.914274987394E+00
7	1.333333333333E+00	3.349223450239E+00
8	1.555555555556E+00	3.873520328779E+00
9	1.777777777778E+00	4.533977541282E+00
10	2.000000000000E+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	3.70668784E+00	4.47340722E-05
1.21496244E+01	1.21496381E+01	1.12618381E-04

GRADIENTS ARE O. K.

NPUT DATA

'OLLOWING METHODS HAVE BEEN CALLED

'LETCHER METHOD

IUMBER OF INDEPENDENT VARIABLES......................................N= 2

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

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

TARTING 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)=1.00000000E-06
EPS(2)=1.00000000E-06

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

Fig. 2

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 7.49777444E-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	1.28990916E-01 4.41758874E-01	-2.31433531E-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 -8.28216978E-05
8	10	1.49000000E-01	1.26146669E+00	2.11935745E-02 4.81513917E-01	8.44108900E-07 3.17448999E-06

IEXIT= 1CRITERION FOR OPTIMUM HAS BEEN SATISFIED

Fig. 2 (continued)

FOLLOWING IS THE OPTIMUM SOLUTION

F = 1.26146669E+00

A(1)= 2.11935745E-02
A(2)= 4.81513917E-01

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	5.566507806018E-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	1.555555555556E+00	-1.055082187242E-01
9	1.777777777778E+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	1.555555555556E+00	-1.055082187242E-01
9	1.777777777778E+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)
C
C        SUBROUTINE WHICH COORDINATES THE OTHER
C        SUBROUTINES IN THE PACKAGE FMCLP
C
      EXTERNAL FUNCT
      EXTERNAL S,W
      DIMENSION A(1), XSTRT(1), G(1), Y(1), PY(1), DUM1(1), DUM2(1), EPS
     1(1), H(1), GRAD(1), X(1), ERROR(1), IPA(1)
      COMMON T1,KO,NFE
      LOGICAL CONV,UNITH
      ERR(Z)=WERR(Z,S,W,A,GRAD)
      UNITH=.TRUE.
      T1=0.
      READ (5,42) N1,N,NPOD,IREAD,IOPT,ITER,IGRDCH
      READ (5,44) XA,XB
      XAB=XB-XA
      NP=N+1
      IF (IREAD.EQ.0) IREAD=2
      GO TO (3,1), IREAD
1     X(2)=XA
      K=N-1
      DELTA=XAB/K
      DO 2 I=1,K
      X(I+2)=XA+I*DELTA
2     CONTINUE
      X(N+1)=XB
      GO TO 4
3     READ (5,44) (X(I),I=2,NP)
4     WRITE (6,52)
      READ (5,44) EST,DIF
      READ (5,44) (XSTRT(I),I=1,N1)
      DO 5 I=1,N1
      A(I)=XSTRT(I)
5     CONTINUE
      DO 6 I=2,NP
      II=I-1
      ERROR(I)=ERR(X(I))
      WRITE (6,53) II,X(I),ERROR(I)
6     CONTINUE
      IF (NPOD.LE.1) GO TO 9
      CALL NEWSET (XA,XB,NPOD,S,W,A,GRAD,N,X,ERROR,IQI)
      WRITE (6,45) IQI
      IF (IQI) 9,9,7
7     WRITE (6,47)
      DO 8 I=2,NP
      II=I-1
      ERROR(I)=ERR(X(I))
      WRITE (6,53) II,X(I),ERROR(I)
8     CONTINUF
9     ERROR(1)=ERROR(2)
      DO 10 I=3,NP
      ERROR(1)=AMAX1(ERROR(1),ERROR(I))
10    CONTINUE
      WRITE (6,50)
      WRITE (6,51)
      WRITE (6,49) FRROR(1)
C
C         DATA FOR THE OPTIMALITY
```

```
C          FOR THE OPTIMIZATION METHOD USED
C
      DO 41 K=1,IOPT
      KR=1
      IF (K-1) 12,12,11
11    READ (5,44) (XSTRT(I),I=1,N1)
12    READ (5,42) MET,MAX,IPRINT,IDATA
      IF (MET.EQ.1) READ (5,44) (EPS(I),I=1,N1)
      IF (MET.EQ.2) READ (5,44) EPS1
      READ (5,42) (IPA(I),I=1,ITER)
      DO 40 KK=1,ITER
C
C          OPTIMIZATION
C
      IP=IPA(KK)
      IF (KK.GT.1) FF=F
      IF (KR.EQ.0) GO TO 13
13    DO 14 I=1,N1
      A(I)=XSTRT(I)
14    CONTINUE
      IF (IGRDCH.NE.1) GO TO 15
      CALL GRDCHK (N1,A,G,PY,Y,GRAD,N,X,ERROR,IP,DUM1)
15    IF (KR.EQ.0) GO TO 16
      IF (IDATA.EQ.0) GO TO 16
      M=2
      CALL INPUT (MET,M,MAX,N1,IPRINT,IDATA,EPS1,EST,EPS,XSTRT)
16    IF (MET.EQ.0) MET=4
      INDEX=0
      GO TO (17,23,33,29), MET
17    CONTINUE
      CALL SECOND (T1)
      IF (IPRINT.EQ.0) GO TO 18
      CALL WRITE1 (1)
18    IF (KR.NE.0) GO TO 20
      DO 19 I=1,N1
      A(I)=DUM1(I)
19    CONTINUE
20    CALL EMNEC (N1,A,F,G,H,UNITH,EST,EPS,MAX,IPRINT,IEXIT,GRAD,N,X,ERR
     1OR,IP)
      DO 21 I=1,N1
      DUM1(I)=A(I)
21    CONTINUE
      CALL SECOND (T2)
      CALL FINAL (A,F,N1,MET)
      T=T2-T1
      IF (T1.EQ.0.) GO TO 22
      WRITE (6,43) T
22    CONTINUE
      GO TO 29
23    CONTINUE
      CALL SECOND (T1)
      IF (IPRINT.EQ.0) GO TO 24
      CALL WRITE1 (2)
24    IF (KR.NE.0) GO TO 26
      DO 25 I=1,N1
      A(I)=DUM2(I)
25    CONTINUE
26    CALL EMFEPC (FUNCT,N1,A,F,G,EST,EPS1,MAX,IER,H,IPRINT,GRAD,N,X,ERRO
     1R,IP)
```

```
      DO 27 I=1,N1
      DUM2(I)=A(I)
27    CONTINUE
      CALL SECOND (T2)
      CALL FINAL (A,F,N1,MET)
      T=T2-T1
      IF (T1.EQ.0.) GO TO 28
      WRITE (6,43) T
28    CONTINUE
29    INDEX=INDEX+1
      IF (M.EQ.1) GO TO 30
      GO TO 32
30    DO 31 I=1,N1
      A(I)=XSTRT(I)
31    CONTINUE
32    CONTINUE
C
33    KR=0
      WRITE (6,46) IP
      WRITE (6,48)
      DO 34 I=2,NP
      II=I-1
      WRITE (6,53) II,X(I),ERROR(I)
34    CONTINUE
      IF (NPOD.LE.1) GO TO 37
      CALL NEWSET (XA,XB,NPOD,S,W,A,GRAD,N,X,ERROR,IQI)
      WRITE (6,45) IQI
      IF (IQI) 37,37,35
35    WRITE (6,47)
      DO 36 I=2,NP
      II=I-1
      ERROR(I)=ERR(X(I))
      WRITE (6,53) II,X(I),ERROR(I)
36    CONTINUE
37    ERROR(1)=ERROR(2)
      DO 38 I=3,NP
      ERROR(1)=AMAX1(ERROR(1),ERROR(I))
38    CONTINUE
      WRITE (6,50)
      WRITE (6,51)
      WRITE (6,49) ERROR(1)
      IGRDCH=IGRDCH+2
      IF (KK-1) 40,40,39
39    FTST=ABS((FF-F)/FF)
      IF (FTST.LT.DIF) GO TO 41
40    CONTINUE
41    CONTINUE
      RETURN
C
C
42    FORMAT (8I10)
43    FORMAT (1H0,//25X,26HEXECUTION TIME IN SECONDS ,F10.5)
44    FORMAT (5E16.8)
45    FORMAT (1H1,14X,15HNUMBER OF Q.I.=,I3)
46    FORMAT (1H1,13X,4HP  =,I7)
47    FORMAT (/8X,31HNEW SET OF INDEPENDENT VARIABLE,15X,6HERRORS/)
48    FORMAT (/13X,20HINDEPENDENT VARIABLE,21X,6HERRORS/)
49    FORMAT (13X,F20.12)
50    FORMAT (//15X,14HABSOLUTE VALUE)
```

```
51     FORMAT (15X,16HOF MAXIMUM ERROR/)
52     FORMAT (1H1,8X,35HINITIAL SET OF INDEPENDENT VARIABLE,9X,6HERRORS/
      1)
53     FORMAT (I9,1X,E23.12,10X,E23.12)
       END
```

```
       FUNCTION WERR (Z,S,W,B,GRAD)
C
C          FUNCTION SUBPROGRAM WHICH CALCULATES
C          WEIGHTED ERROR FUNCTION
C
       EXTERNAL S,W
       DIMENSION B(1), GRAD(1)
       CALL FAPP (Z,N1,B,APP,GRAD,1)
       WERR=(APP-S(Z))*W(Z)
       RETURN
       END
```

```
       SUBROUTINE NEWSET (XA,XB,NPOD,S,W,B,GRAD,N,X,ERROR,IQI)
C
C          SUBROUTINE WHICH CALCULATES THE NEW SET OF THE INDEPENDENT
C          VARIABLES WHICH INCLUDE ALL THE EXTREMA OF THE WEIGHTED
C          ERROR FUNCTION
C
       EXTERNAL S,W
       DIMENSION B(1), GRAD(1), X(1), ERROR(1)
       ERR(Z)=WERR(Z,S,W,B,GRAD)
       EPSN(Z)=ABS(WERR(Z,S,W,B,GRAD))
       IER=0
       NN=N+1
       NNN=N+2
       X(1)=XA
       X(NNN)=XB
       II=2
       IQ=1
       IND=1
       IQI=0
       DO 29 I=1,NN
       IF (X(I)-X(I+1)) 1,29,29
1      ZMIN=X(I+1)
       ZMAX=X(I)
       IQ=IND
       IND=1
       Z=ZMAX
       EMAX=EPSN(Z)
       EMIN=EPSN(ZMIN)
       DELTA=(ZMIN-ZMAX)/NPOD
       NPOD1=NPOD+1
       MM=1
```

```
      MIN=NPOD1
      DO 7 K=1,NPOD
      ETREN=EPSN(Z)
      IF (ETREN-EMAX) 6,6,2
2     EMAX=ETREN
      MM=K
      ZMAX=Z
      ZTEST1=ZMAX+DELTA
      ETEST1=EPSN(ZTEST1)
      IF (EMAX-ETEST1) 6,6,3
3     ZTEST2=ZMAX-DELTA
      IF (XA-(ZMAX-DELTA)) 4,4,6
4     ETEST2=EPSN(ZTEST2)
      IF (EMAX-ETEST2) 6,6,5
5     IND=2
      GO TO 8
6     Z=Z+DELTA
7     CONTINUE
C
8     IF (XA-(ZMAX-DELTA)) 9,9,13
9     IF (MM-NPOD) 10,10,13
10    GO TO (13,11), IND
11    Q1=EPSN(ZMAX-DELTA)
      Q2=EPSN(ZMAX+DELTA)
      RA=(EMAX*2.-Q1-Q2)*2.
      IF (RA) 12,13,12
12    ZMAX=ZMAX+(Q2-Q1)*DELTA/RA
      IQI=IQI+1
      ZMIN=ZTEST1
C
13    IF (I-1) 14,14,17
14    IF (ZMAX-ZMIN) 15,29,29
15    GO TO (29,16), IND
16    X(1)=ZMAX
      XKP=X(2)
      X(2)=ZMIN
      GO TO 29
C
17    IF (X(I-1)-ZMAX) 18,21,21
18    IF (ZMAX-ZMIN) 19,21,21
19    GO TO (23,20), IND
20    X(I)=ZMAX
21    IF (X(I)-ZMIN) 22,25,25
22    XKP=X(I+1)
      X(I+1)=ZMIN
      GO TO 25
23    GO TO (25,24), IQ
24    X(I)=XKP
C
25    IF (I-NN) 29,26,29
26    IF (X(I)-X(I+1)) 27,29,29
27    IF (EPSN(X(I))-EPSN(X(I+1))) 28,29,29
28    II=II-1
29    CONTINUE
      IF (EPSN(X(2))-EPSN(X(1))) 30,31,31
30    II=II-3
C
31    IF (II-2) 32,50,32
C
```

```
32    IF (II+2) 42,33,42
33    JJ=NN/2+1
      IF (N-2) 39,39,34
34    IF (EPSN(X(JJ-1))-EPSN(X(JJ+1))) 35,36,36
35    JJ=JJ-1
      GO TO 37
36    JJ=JJ+1
37    IF (EPSN(X(JJ-1))-EPSN(X(JJ+1))) 38,39,39
38    JJ=JJ-1
39    DO 40 K=2,JJ
      I=JJ+2-K
      X(I)=X(I-1)
40    CONTINUE
      JJJ=JJ+1
      DO 41 K=JJJ,NN
      I=K+1
      X(I-1)=X(I)
41    CONTINUE
C
42    IF (II-1) 46,43,46
43    IF (EPSN(X(NNN))-EPSN(X(2))) 50,44,44
C
44    DO 45 I=2,NN
      X(I)=X(I+1)
45    CONTINUE
46    IF (II+1) 50,47,50
47    IF (EPSN(X(1))-EPSN(X(NN))) 50,50,48
C
48    DO 49 K=2,NN
      I=NN+2-K
      X(I)=X(I-1)
49    CONTINUE
50    CONTINUE
      RETURN
      END
```

••

```
      SUBROUTINE FUNCT (N1,B,OBJ,G,GRAD,N,X,ERROR,IP)
C
C         SUBROUTINE WHICH SELECTS THE MAXIMUM ERROR
C         AND COMPUTES THE OBJECTIVE FUNCTION AND ITS
C         GRADIENTS W.R.T. THE VARIABLE PARAMETERS
C         IN THE LEAST P-TH SENSE
C
      EXTERNAL S,W
      DIMENSION B(1), G(1), GRAD(1), X(1), ERROR(1)
      ERR(Z)=WERR(Z,S,W,B,GRAD)
      OBJP=0.
      GRADP=0.
      DO 1 K=1,N1
      G(K)=0.
1     CONTINUE
      NN=N+1
      ERROR(1)=0.
      DO 2 I=2,NN
      ERROR(I)=ERR(X(I))
```

```
2     CONTINUE
      DO 4 I=2,NN
      IF (ABS(ERROR(I))-ERROR(1)) 4,4,3
3     ERROR(1)=ABS(ERROR(I))
4     CONTINUE
      DO 6 I=2,NN
      Z=X(I)
      DEC=ERROR(I)/ERROR(1)
      DEL=ABS(DEC)
      OBJI=DEL**IP
      GRADI=DEL**(IP-2)*DEC
      OBJP=OBJP+OBJI
      CALL FAPP (Z,N1,B,APP,GRAD,2)
      DO 5 K=1,N1
      GRAD(K)=GRADI*W(Z)*GRAD(K)
      G(K)=G(K)+GRAD(K)
5     CONTINUE
6     CONTINUE
      PR=1./IP
      OBJ=ERROR(1)*(OBJP**PR)
      GRP=OBJP**(PR-1.)
      DO 7 K=1,N1
      G(K)=GRP*G(K)
7     CONTINUE
      RETURN
      END
```

••

```
      SUBROUTINE 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)
      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
      DELX=1.E-4*A(I)
      GO TO 2
1     DELX=1.E-20
2     A(I)=A(I)+DELX
      CALL FUNCT (N,A,FNEW,PY,GRAD,NP,XP,ERROR,IP)
      Y(I)=(FNEW-F)/DELX
      DUM1(I)=Y(I)
      A(I)=A(I)-DELX
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
      WRITE (6,12) G(I),Y(I),PY(I)
5     CONTINUE
      DO 6 I=1,N
      IF (PY(I).GT.10.) GO TO 7
6     CONTINUE
      WRITE (6,13)
      RETURN
7     WRITE (6,14)
      CALL EXIT
C
C
8     FORMAT (1H1)
9     FORMAT (1H0,5X,18HGRADIENTS CHECKING,/,6X,18(1H-),//,6X,50HGRADIEN
     1TS HAVE BEEN CHECKED AT THE FOLLOWING POINT/)
10    FORMAT (10X,2HA(,I2,2H)=,E16.8)
11    FORMAT (///,1H0,5X,20HANALYTICAL GRADIENTS,5X,19HNUMERICAL GRADIEN
     1TS,5X,16HPERCENTAGE ERROR,/)
12    FORMAT (1H0,5X,3(E16.8,9X))
13    FORMAT (1H0,//,6X,19HGRADIENTS ARE O. K.)
14    FORMAT (1H0,//,6X,64HYOUR PROGRAM HAS BEEN TERMINATED BECAUSE GRAD
     1IENTS ARE INCORRECT,/6X,21HPLEASE CHECK IT AGAIN)
      END
```

```
      SUBROUTINE FMNFC (N,X,F,G,H,UNITH,FEST,EPS,MAXFN,IPRINT,IEXIT,GRAD
     1,NP,XP,ERROR,IP)
C
C        PURPOSE
C           TO FIND A LOCAL MINIMUM OF A FUNCTION OF SEVERAL VARIABLES
C           ASSUMING THAT ITS GRADIENTS CAN BE CALCULATED EXPLICITLY
C           BY THE METHOD OF FLETCHER
C
C           THE METHOD IS DESCRIBED IN THE FOLLOWING ARTICLE
C           R. FLETCHER, A NEW APPROACH TO VARIABLE METRIC ALGORITHMS,
C           COMP. JOURNAL, VOL.13, 1970, PP.317-322.
C
      DIMENSION X(1), G(1), H(1), EPS(1), GRAD(1), XP(1), ERROR(1)
      LOGICAL CONV,UNITH
      COMMON T1,KO,NFNS
      CALL SECOND (T3)
      KO=0
      CALL FUNCT (N,X,F,G,GRAD,NP,XP,ERROR,IP)
      IF (F.LT.FEST) GO TO 23
      NFNS=1
      ITN=0
      STEP=1.
      IDX=N
      IDG=N+N
      IH=IDG+N
      IF (.NOT.UNITH) GO TO 2
      IJ=IH+1
      DO 1 I=1,N
      DO 1 J=I,N
      H(IJ)=0.
```

```
      IF (I.EQ.J) H(IJ)=1.0
1     IJ=IJ+1
2     CONV=.TRUE.
      GDX=0.
      DO 6 I=1,N
      Z=0.
      IJ=IH+I
      IF (I.EQ.1) GO TO 4
      II=I-1
      DO 3 J=1,II
      Z=Z-H(IJ)*G(J)
      IJ=IJ+N-J
3     CONTINUE
4     DO 5 J=I,N
      Z=Z-H(IJ)*G(J)
      IJ=IJ+1
5     CONTINUE
      IF (ABS(Z).GT.EPS(I)) CONV=.FALSE.
      H(IDX+I)=Z
      GDX=GDX+G(I)*Z
6     CONTINUE
C
      IF (IPRINT.EQ.0) GO TO 7
      IF (MOD(ITN,IPRINT).NE.0) GO TO 7
      CALL SECOND (T4)
      TIME=T4-T3
      CALL WRITE2 (X,N,G,F,NFNS,ITN,TIME)
7     IEXIT=1
      IF (CONV) GO TO 24
      IEXIT=2
      IF (GDX.GE.0.) GO TO 24
      Z=1.
      IF (ITN.LT.N.AND.UNITH) Z=STEP
      W=2.*(FEST-F)/GDX
      IF (W.LT.Z) Z=W
      STEP=Z
8     GDX=GDX*Z
      DO 9 I=1,N
      H(IDX+I)=H(IDX+I)*Z
      X(I)=X(I)+H(IDX+I)
9     CONTINUE
      CALL FUNCT (N,X,FP,H,GRAD,NP,XP,ERROR,IP)
      IF (FP.LT.FEST) GO TO 23
      NFNS=NFNS+1
      IEXIT=3
      IF (ITN.EQ.MAXFN) GO TO 24
      GPDX=0.
      DO 10 I=1,N
      H(IDG+I)=H(I)-G(I)
      GPDX=GPDX+H(I)*H(IDX+I)
10    CONTINUE
      DGDX=GPDX-GDX
      IF (F.GT.FP-.0001*GDX) GO TO 12
      IEXIT=4
      IF (GPDX.LT.0..AND.ITN.GT.N) GO TO 24
      Z=3.*(F-FP)+GPDX+GDX
      W=SQRT(1.-GDX/Z*GPDX/Z)*ABS(Z)
      Z=1.-(GPDX+W-Z)/(DGDX+2.*W)
      IF (Z.LT.0.1) Z=0.1
```

```
      DO 11 I=1,N
      X(I)=X(I)-H(IDX+I)
 11   CONTINUE
      GO TO 14
 12   F=FP
      DO 13 I=1,N
      G(I)=H(I)
 13   CONTINUE
      IF (DGDX.GT.0.) GO TO 15
      GDX=GPDX
      Z=4.
 14   STEP=Z*STEP
      GO TO 8
 15   IF (GPDX.LT.0.5*GDX) STEP=2.*STEP
      DGHDG=0.
      DO 19 I=1,N
      Z=0.
      IJ=IH+I
      IF (I.EQ.1) GO TO 17
      II=I-1
      DO 16 J=1,II
      Z=Z+H(IJ)*H(IDG+J)
      IJ=IJ+N-J
 16   CONTINUE
 17   DO 18 J=I,N
      Z=Z+H(IJ)*H(IDG+J)
      IJ=IJ+1
 18   CONTINUE
      DGHDG=DGHDG+Z*H(IDG+I)
      H(I)=Z
 19   CONTINUE
      IF (DGHDG.LT.0.0) DGHDG=DGDX*0.01
      IF (DGDX.LT.DGHDG) GO TO 21
      W=1.0+DGHDG/DGDX
      DO 20 I=1,N
      H(IDX+I)=W*H(IDX+I)-H(I)
 20   CONTINUE
      DGDX=DGDX+DGHDG
      DGHDG=DGDX
 21   IJ=IH
      DO 22 I=1,N
      W=H(IDX+I)/DGDX
      Z=H(I)/DGHDG
      DO 22 J=I,N
      IJ=IJ+1
 22   H(IJ)=H(IJ)+W*H(IDX+J)-Z*H(J)
      ITN=ITN+1
      GO TO 2
 23   IEXIT=5
 24   IF (IEXIT.EQ.1) KO=1
      IF (IPRINT.EQ.0) RETURN
      GO TO (25,26,27,26,28), IEXIT
 25   WRITE (6,30) IEXIT
      GO TO 29
 26   WRITE (6,31) IEXIT
      GO TO 29
 27   WRITE (6,32) IEXIT
      GO TO 29
 28   WRITE (6,33) IEXIT
```

```
29    CONTINUE
      RETURN
C
C
30    FORMAT (/,1HO,6HIEXIT=,I2,40HCRITERION FOR OPTIMUM HAS BEEN SATISF
     1IED)
31    FORMAT (/,1HO,6HIEXIT=,I2,43HEITHER OF THE FOLLOWING THINGS HAS HA
     1PPENED,/,9X,26H1. EPS CHOSEN IS TOO SMALL,/,9X,28H2. GRADIENTS ARE
     2NOT CORRECT,/,9X,25H3. MATRIX H GOES SINGULAR)
32    FORMAT (/,1HO,6HIEXIT=,I2,55HMAXIMUM NUMBER OF ALLOWABLE ITERATION
     1 HAS BEEN EXCEEDED)
33    FORMAT (/,1HO,6HIEXIT=,I2,60HFUNCTION VALUE LESS THAN MINIMUM ESTI
     1MATED HAS BEEN DETECTED)
      END
```

....

```
      SUBROUTINE FMFPC (FUNCT,N,X,F,G,EST,EPS,LIMIT,IER,H,IPRINT,GRAD,NP
     1,XP,ERROR,IP)
C
C        PURPOSE
C           TO FIND A LOCAL MINIMUM OF A FUNCTION OF SEVERAL VARIABLES
C           ASSUMING THAT ITS GRADIENTS CAN BE CALCULATED EXPLICITLY
C           BY THE METHOD OF FLETCHER AND POWELL
C
C           THE METHOD IS DESCRIBED IN THE FOLLOWING ARTICLE
C           R. FLETCHER AND M.J.D. POWELL, A RAPIDLY CONVERGENT
C           DESCENT METHOD FOR MINIMIZATION, COMP. JOURNAL,
C           VOL.6, 1963, PP.163-168.
C
      COMMON T1,KO,NUMF
C
C        DIMENSIONED DUMMY VARIABLES
      DIMENSION H(1), X(1), G(1), GRAD(1), XP(1), ERROR(1)
C
C        COMPUTE FUNCTION VALUE AND GRADIENT VECTOR FOR INITIAL ARGUMENT
      KO=0
      CALL SECOND (T3)
      CALL FUNCT (N,X,F,G,GRAD,NP,XP,ERROR,IP)
      KOUNT=0
      NUMF=1
      CALL SECOND (T4)
      TIME=T4-T3
      IF (IPRINT.EQ.0) GO TO 1
      CALL WRITE2 (X,N,G,F,NUMF,KOUNT,TIME)
1     CONTINUE
C
C        RESET ITERATION COUNTER AND GENERATE IDENTITY MATRIX
      IER=0
      KK=0
      N2=N+N
      N3=N2+N
      N31=N3+1
2     K=N31
      DO 5 J=1,N
      H(K)=1.
```

```
      NJ=N-J
      IF (NJ) 6,6,3
 3    DO 4 L=1,NJ
      KL=K+L
      H(KL)=0.
 4    CONTINUE
      K=KL+1
 5    CONTINUE
C
C        START ITERATION LOOP
 6    IF (KOUNT.EQ.0) GO TO 7
      IF (KK.NE.IPRINT) GO TO 7
      KK=0
      CALL SECOND (T4)
      TIME=T4-T3
      CALL WRITE2 (X,N,G,F,NUMF,KOUNT,TIME)
 7    CONTINUE
      KOUNT=KOUNT+1
      KK=KK+1
C
C        SAVE FUNCTION VALUE, ARGUMENT VECTOR AND GRADIENT VECTOR
      OLDF=F
      DO 11 J=1,N
      K=N+J
      H(K)=G(J)
      K=K+N
      H(K)=X(J)
C
C        DETERMINE DIRECTION VECTOR H
      K=J+N3
      T=0.
      DO 10 L=1,N
      T=T-G(L)*H(K)
      IF (L-J) 8,9,9
 8    K=K+N-L
      GO TO 10
 9    K=K+1
 10   CONTINUE
      H(J)=T
 11   CONTINUE
C
C        CHECK WHETHER FUNCTION WILL DECREASE STEPPING ALONG H.
      DY=0.
      HNRM=0.
      GNRM=0.
C
C        CALCULATE DIRECTIONAL DERIVATIVE AND TESTVALUES FOR DIRECTION
C        VECTOR H AND GRADIENT VECTOR G.
      DO 12 J=1,N
      HNRM=HNRM+ABS(H(J))
      GNRM=GNRM+ABS(G(J))
      DY=DY+H(J)*G(J)
 12   CONTINUE
C
C        REPEAT SEARCH IN DIRECTION OF STEEPEST DESCENT IF DIRECTIONAL
C        DERIVATIVE APPEARS TO BE POSITIVE OR ZERO.
      IF (DY) 13,57,57
C
C        REPEAT SEARCH IN DIRECTION OF STEEPEST DESCENT IF DIRECTION
```

```
C        VECTOR H IS SMALL COMPARED TO GRADIENT VECTOR G.
 13   IF (HNRM/GNRM-EPS) 57,57,14
C
C        SEARCH MINIMUM ALONG DIRECTION H
C
C        SEARCH ALONG H FOR POSITIVE DIRECTIONAL DERIVATIVE
 14   FY=F
      ALFA=2.*(EST-F)/DY
      AMBDA=1.
C
C        USE ESTIMATE FOR STEPSIZE ONLY IF IT IS POSITIVE AND LESS THAN
C        1. OTHERWISE TAKE 1. AS STEPSIZE
      IF (ALFA) 17,17,15
 15   IF (ALFA-AMBDA) 16,17,17
 16   AMBDA=ALFA
 17   ALFA=0.
C
C        SAVE FUNCTION AND DERIVATIVE VALUES FOR OLD ARGUMENT
 18   FX=FY
      DX=DY
C
C        STEP ARGUMENT ALONG H
      DO 19 I=1,N
      X(I)=X(I)+AMBDA*H(I)
 19   CONTINUE
C
C        COMPUTE FUNCTION VALUE AND GRADIENT FOR NEW ARGUMENT
      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
C        SEARCH, IF DY IS POSITIVE. IF DY IS ZERO THE MINIMUM IS FOUND
      DY=0.
      DO 20 I=1,N
      DY=DY+G(I)*H(I)
 20   CONTINUE
      IF (DY) 21,41,24
C
C        TERMINATE SEARCH ALSO IF THE FUNCTION VALUE INDICATES THAT
C        A MINIMUM HAS BEEN PASSED
 21   IF (FY-FX) 22,24,24
C
C        REPEAT SEARCH AND DOUBLE STEPSIZE FOR FURTHER SEARCHES
 22   AMBDA=AMBDA+ALFA
      ALFA=AMBDA
C        END OF SEARCH LOOP
C
C        TERMINATE IF THE CHANGE IN ARGUMENT GETS VERY LARGE
      IF (HNRM*AMBDA-1.E10) 18,18,23
C
C        LINEAR SEARCH TECHNIQUE INDICATES THAT NO MINIMUM EXISTS
 23   IER=2
      GO TO 62
C
C        INTERPOLATE CUBICALLY IN THE INTERVAL DEFINED BY THE SEARCH
C        ABOVE AND COMPUTE THE ARGUMENT X FOR WHICH THE INTERPOLATION
C        POLYNOMIAL IS MINIMIZED
 24   T=0.
```

```
25    IF (AMBDA) 26,41,26
26    Z=3.*(FX-FY)/AMBDA+DX+DY
      ALFA=AMAX1(ABS(Z),ABS(DX),ABS(DY))
      DALFA=Z/ALFA
      DALFA=DALFA*DALFA-DX/ALFA*DY/ALFA
      IF (DALFA) 57,27,27
27    W=ALFA*SQRT(DALFA)
      ALFA=DY-DX+W+W
      IF (ALFA) 28,29,28
28    ALFA=(DY-Z+W)/ALFA
      GO TO 30
29    ALFA=(Z+DY-W)/(Z+DX+Z+DY)
30    ALFA=ALFA*AMBDA
      DO 31 I=1,N
      X(I)=X(I)+(T-ALFA)*H(I)
31    CONTINUE
C
C        TERMINATE, IF THE VALUE OF THE ACTUAL FUNCTION AT X IS LESS
C        THAN THE FUNCTION VALUES AT THE INTERVAL ENDS. OTHERWISE REDUCE
C        THE INTERVAL BY CHOOSING ONE END-POINT EQUAL TO X AND REPEAT
C        THE INTERPOLATION.  WHICH END-POINT IS CHOOSEN DEPENDS ON THE
C        VALUE OF THE FUNCTION AND ITS GRADIENT AT X
C
      NUMF=NUMF+1
      CALL FUNCT (N,X,F,G,GRAD,NP,XP,ERROR,IP)
      IF (F-FX) 32,32,33
32    IF (F-FY) 41,41,33
33    DALFA=0.
      DO 34 I=1,N
      DALFA=DALFA+G(I)*H(I)
34    CONTINUE
      IF (DALFA) 35,38,38
35    IF (F-FX) 37,36,38
36    IF (DX-DALFA) 37,41,37
37    FX=F
      DX=DALFA
      T=ALFA
      AMBDA=ALFA
      GO TO 25
38    IF (FY-F) 40,39,40
39    IF (DY-DALFA) 40,41,40
40    FY=F
      DY=DALFA
      AMBDA=AMBDA-ALFA
      GO TO 24
C
C        TERMINATE, IF FUNCTION HAS NOT DECREASED DURING LAST ITERATION
41    IF (OLDF-F+EPS) 57,42,42
C
C        COMPUTE DIFFERENCE VECTORS OF ARGUMENT AND GRADIENT FROM
C        TWO CONSECUTIVE ITERATIONS
42    DO 43 J=1,N
      K=N+J
      H(K)=G(J)-H(K)
      K=N+K
      H(K)=X(J)-H(K)
43    CONTINUE
C
C        TEST LENGTH OF ARGUMENT DIFFERENCE VECTOR AND DIRECTION VECTOR
```

```
C        IF AT LEAST N ITERATIONS HAVE BEEN EXECUTED. TERMINATE, IF
C        BOTH ARE LESS THAN  EPS
      IER=0
      IF (KOUNT-N) 47,44,44
44    T=0.
      DO 45 J=1,N
      K=N+J
      W=H(K)
      K=K+N
      T=T+ABS(H(K))
45    CONTINUE
      IF (HNRM-EPS) 46,46,47
46    IF (T-EPS) 62,62,47
C
C        TERMINATE, IF NUMBER OF ITERATIONS WOULD EXCEED  LIMIT
47    IF (KOUNT-LIMIT) 48,55,55
C
C        PREPARE UPDATING OF MATRIX H
48    ALFA=0.
      Z=0.
      DO 52 J=1,N
      K=J+N3
      W=0.
      DO 51 L=1,N
      KL=N+L
      W=W+H(KL)*H(K)
      IF (L-J) 49,50,50
49    K=K+N-L
      GO TO 51
50    K=K+1
51    CONTINUE
      K=N+J
      KN=K+N
      Z=Z+H(K)*H(KN)
      ALFA=ALFA+W*H(K)
      H(J)=W
52    CONTINUE
C
C        REPEAT SEARCH IN DIRECTION OF STEEPEST DESCENT IF RESULTS
C        ARE NOT SATISFACTORY
      IF (Z*ALFA) 53,2,53
C
C        UPDATE MATRIX H
53    K=N31
      DO 54 L=1,N
      KL=N2+L
      DO 54 J=L,N
      NJ=N2+J
      H(K)=H(K)+H(KL)*H(NJ)/Z-H(L)*H(J)/ALFA
54    K=K+1
      GO TO 6
C        END OF ITERATION LOOP
C
C        NO CONVERGENCE AFTER  LIMIT  ITERATIONS
55    IER=1
      IF (KK.NE.IPRINT) GO TO 56
      CALL WRITE2 (X,N,G,F,NUMF,KOUNT,TIME)
56    CONTINUE
      GO TO 62
```

```
C
C        RESTORE OLD VALUES OF FUNCTION AND ARGUMENTS
 57   DO 58 J=1,N
      K=N2+J
      X(J)=H(K)
 58   CONTINUE
      CALL FUNCT (N,X,F,G,GRAD,NP,XP,ERROR,IP)
      NUMF=NUMF+1
C
C        REPEAT SEARCH IN DIRECTION OF STEEPEST DESCENT IF DERIVATIVE
C        FAILS TO BE SUFFICIENTLY SMALL
      IF (GNRM-EPS) 61,61,59
C
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 (1H0,4HIER=,I2,32H ERROR IN GRADIENTS CALCULATIONS)
 69   FORMAT (1H0,4HIER=,I2,41H CRITERION FOR OPTIMUM HAS BEEN SATISFIED
     1)
 70   FORMAT (1H0,4HIER=,I2,57H MAXIMUM NUMBER OF ALLOWABLE ITERATIONS H
     1AS BEEN EXCEEDED)
 71   FORMAT (1H0,4HIER=,I2,83H CHANGE IN ARGUMENTS GETS TOO LARGE, LINE
     1AR SEARCH INDICATES THAT NO MINIMUM EXISTS)
      END
```

....

```
      SUBROUTINE INPUT (MET,M,MAX,N,IPRINT,IDATA,EPS1,EST,EPS,ASTRT)
C
C         PRINTS THE INPUT DATA
C         FOR THE OPTIMIZATION PROCESS
C
      DIMENSION ASTRT(1), EPS(1)
      WRITE (6,5)
      IF (MET.NE.1.AND.MET.NE.2) GO TO 4
      INDEX=0
      GO TO (1,2), MET
 1    WRITE (6,6)
      GO TO 3
```

```
2     WRITE (6,7)
3     CONTINUE
      WRITE (6,8) N
      WRITE (6,9) MAX
      WRITE (6,10) IPRINT
      WRITE (6,11) ASTRT(1)
      WRITE (6,12) (I,ASTRT(I),I=2,N)
      IF (MET.EQ.1) WRITE (6,13) EPS(1)
      IF (MET.EQ.1) WRITE (6,14) (I,EPS(I),I=2,N)
      IF (MET.EQ.2) WRITE (6,15) EPS1
      WRITE (6,16) FST
      RETURN
4     WRITE (6,17)
      CALL EXIT
C
C
5     FORMAT (1H1,10HINPUT DATA,/,1X,10(1H-),//,1X,34HFOLLOWING METHODS
     1HAVE BEEN CALLED,/)
6     FORMAT (1H0,15HFLETCHER METHOD)
7     FORMAT (1H0,22HFLETCHER-POWELL METHOD)
8     FORMAT (1H0,/1X,31HNUMBER OF INDEPENDENT VARIABLES,36(1H.),2HN=,I5
     1,/)
9     FORMAT (1H0,38HMAXIMUM NUMBER OF ALLOWABLE ITERATIONS,27(1H.),4HMA
     1X=,I5,/)
10    FORMAT (1H0,57HINTERMEDIATE OUTPUT TO BE PRINTED EVERY IPRINT ITER
     1ATIONS,5(1H.),7HIPRINT=,I5,/)
11    FORMAT (1H0,30HSTARTING VALUE FOR VECTOR A(I),29(1H.),10HASTRT( 1)
     1=,E16.8)
12    FORMAT (1H0,59X,6HASTRT(,I2,2H)=,E16.8)
13    FORMAT (1H0,/,1X,45HTEST QUANTITIES TO BE USED IN FLETCHER METHOD,
     116(1H.),8HEPS( 1)=,E16.8)
14    FORMAT (1H0,61X,4HEPS(,I2,2H)=,E16.8)
15    FORMAT (1H0,/,1X,50HTEST QUANTITY TO BE USED IN FLETCHER-POWELL ME
     1THOD,14(1H.),5HEPS1=,E16.8)
16    FORMAT (1H0,/,1X,51HESTIMATE OF LOWER BOUND ON FUNCTION TO BE MINI
     1MIZED,14(1H.),4HFST=,E16.8)
17    FORMAT (1H0,49HNONE OF THE OPTIMIZATION METHODS HAVE BEEN CALLED,/
     1,1X,29HPLEASE CHECK THE VALUE OF MET,/,1X,9HREMAINDER,/,1X,40HMET=
     21    FLETCHER METHOD WOULD BE CALLED,/,1X,47HMET=2    FLETCHER-POW
     3ELL METHOD WOULD BE CALLED)
      END
```

....

```
      SUBROUTINE FINAL (A,F,N,MET)
C
C         PRINTS THE RESULTS
C         FOR THE OPTIMIZATION PROCESS
C
      COMMON T1,KO,NFF
      DIMENSION A(1)
      WRITE (6,5)
      IF (KO.EQ.0) GO TO 1
      WRITE (6,6)
      GO TO 2
1     WRITE (6,7)
```

```
2     CONTINUE
      WRITE (6,8) F
      WRITE (6,9) (I,A(I),I=1,N)
      GO TO (3,4), MET
3     WRITE (6,10) NFE
      RETURN
4     WRITE (6,11) NFE
      RETURN
C
C
C
5     FORMAT (1H1)
6     FORMAT (41X,33HFOLLOWING IS THE OPTIMUM SOLUTION,/,41X,33(1H-))
7     FORMAT (45X,25HRESULTS AT LAST ITERATION/,45X,25(1H-))
8     FORMAT (//,48X,3HF =,E16.8,/)
9     FORMAT (45X,2HA(,I2,2H)=,E16.8)
10    FORMAT (//25X,53HNUMBER OF FUNCTION EVALUATIONS BY THE FLETCHER ME
     1THOD,I10)
11    FORMAT (//25X,60HNUMBER OF FUNCTION EVALUATIONS BY THE FLETCHER-PO
     1WELL METHOD,I10)
      END
```

••

```
      SUBROUTINE WRITE1 (N)
C
C          PRINTS THE INTERMEDIATE RESULTS
C
      COMMON TIME,KO,NFE
      WRITE (6,5)
      GO TO (1,2), N
1     WRITE (6,6)
      GO TO 3
2     WRITE (6,7)
3     CONTINUE
      IF (TIME.EQ.0.) GO TO 4
      WRITE (6,8)
      RETURN
4     WRITE (6,9)
      RETURN
C
C
5     FORMAT (1H1)
6     FORMAT (1H0,31HOPTIMIZATION BY FLETCHER METHOD,/,1H0,31(1H-))
7     FORMAT (1H0,38HOPTIMIZATION BY FLETCHER-POWELL METHOD,/,1H0,38(1H-
     1))
8     FORMAT (1H0,9HITERATION,2X,8HFUNCTION,6X,12HTIME ELAPSED,8X,9HOBJE
     1CTIVE,14X,20HVARIABLE VECTOR A(I),9X,20HGRADIENT VECTOR G(I),/1H0,
     26HNUMBER,5X,11HEVALUATIONS,3X,9H(SECONDS),11X,8HFUNCTION,/)
9     FORMAT (1H0,9HITERATION,2X,8HFUNCTION,8X,9HOBJECTIVE,14X,20HVARIAB
     1LE VECTOR A(I),9X,20HGRADIENT VECTOR G(I),/1H0,6HNUMBER,5X,11HEVAL
     2UATIONS,5X,8HFUNCTION,/)
      END
```

••

```
      SUBROUTINE WRITE2 (A,N,G,F,NUMF,ITER,TIME)
C
C         PRINTS THE INTERMEDIATE RESULTS
C
      COMMON T1,KO,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 (1HO,I5,7X,I5,5X,E16.8,3X,E16.8,12X,95(E16.8,13X,E16.8,/,70
     1X))
    3 FORMAT (1HO,I5,7X,I5,8X,E16.8,7X,95(E16.8,13X,E16.8,/,49X))
      END
```