INTERNAL REPORTS IN

SIMULATION, OPTIMIZATION AND CONTROL

No. SOC-20

A GENERAL PROGRAM FOR DISCRETE LEAST pTH APPROXIMATION

J.R. Popović and J.W. Bandler

October 1973

FACULTY OF ENGINEERING
McMASTER UNIVERSITY
HAMILTON, ONTARIO, CANADA

A GENERAL PROGRAM FOR DISCRETE LEAST pTH APPROXIMATION

PURPOSE: To minimize an objective function of k variables defined as the generalized discrete least pth objective using gradient methods.

LANGUAGE: FORTRAN IV; 1005 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 FMLPO, is applicable to problems of meeting and/or exceeding design specifications on several disjoint closed intervals and thus is relevant to a wide range of specifications and a wide variety of network and system design problems, especially in filter design.

The program utilizes the approach of the practical generalized least pth approximation proposed by Bandler and Charalambous [1]. Gradient minimization algorithms due to Fletcher and Powell [2] and, more recently, to Fletcher [3] are used. Least pth approximation with p=2

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

gives a discrete least squares approximation. With sufficiently large values of p an optimal solution very close to the optimal minimax solution can be obtained. Values of p up to 10^6 have been successfully employed. Proper scaling alleviates the ill-conditioning when large values of p are used and automatically defines both problems, meeting or exceeding design specifications, into one optimization problem.

The program can be used in a less general least pth approximation problem for fitting a continuous function to another one or to data on a closed interval. Although the program is not written for nonlinear programming, we found that it is also applicable to problems with parameter constraints.

The user has to write all the required specifications in each interval, the approximating functions with partial derivatives and weighting functions for different specifications in a straightforward way. The number of intervals and discrete point sets are user specified as well as the values of p, the parameter constraints and the initial parameter values. Also, the choice about which optimization method is to be used, checking the gradients and the stopping criteria may be made. The optimal point, weighted errors from various intervals 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, number of intervals or discrete point sets.

A recent publication [4] contains the background theory for the optimization algorithm,detailed organization of the program FMLPO and instructions on how to use it for both unconstrained and constrained optimization problems. This includes a block diagram of the package and flowcharts of its subroutines. The examples demonstrating FMLPO were taken in system modelling and multi-section transmission-line filter design. Document NAPS ----- contains a complete listing and detailed user's manual for the given package fully illustrated with examples.

Typically less than a minute of CDC 6400 computer time and a core requirement of about 15 K_{10} is sufficient to optimize a constrained problem with five parameters and fifty-two sample points.

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

[2] R. Fletcher and M.J.D. Powell, "A rapidly convergent descent method for minimization", Computer J., vol. 6, pp.163-168, June 1963.

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

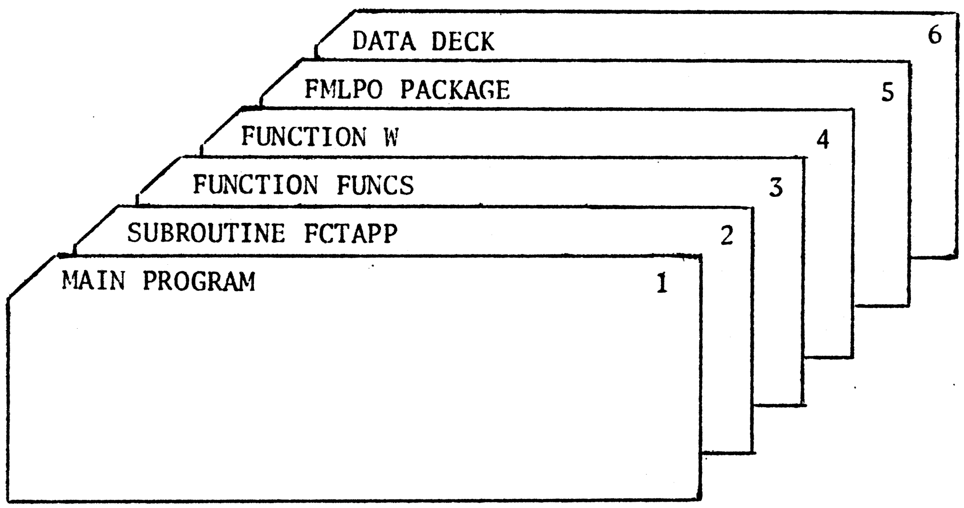
[4] J.R. Popović, J.W. Bandler and C. Charalambous, "General programs for least pth and near minimax approximation", Int. J. Systems Science, to be published.

USER'S MANUAL FOR FMLPO

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

Purpose — To minimize the objective function of k variables $\underset{\sim}{a}$ defined as the generalized discrete least pth objective 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), Y(K), PY(K), DUM1(K), DUM2(K), GRAD(K), EPS(K), H(M), XX(3, NINT), NUMB(N), INUMB(N), X(N), X1(N), ERROR(N), EHELP(N), AP(N), IPA(ITER) where

K is the number of variable parameters,

M = K (K+7)/2,

NINT is the number of intervals,

N is the total discrete point set of independent parameter from all intervals, and

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

Call the subroutine FMLPO as follows:

CALL FMLPO (A, ASTRT, G, Y, PY, DUM1, DUM2, EPS, H, GRAD, NUMB, XX, X, X1, ERROR, EHELP, AP, INUMB, IPA)

2. Subroutine FCTAPP

Subroutine which defines the approximating function in each interval and calculates its gradients with respect to variable vector $\underset{\sim}{a}$.

Write subroutine FCTAPP as follows:

```
SUBROUTINE FCTAPP (X, K, A, APP, GRAD, IINT, INDIC)
DIMENSION A(1), GRAD(1)
```

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

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

Write the approximating function APP $\triangleq$ F(A,X) where $\underset{\sim}{A} \triangleq [A_1\ A_2\ \ldots\ A_K]^T$, and all its gradients GRAD(i) $\triangleq \frac{\partial F(\underset{\sim}{A},X)}{\partial A_i}$,

i=1,2,...,K for each interval IINT = 1,2,..., NINT. The value of APP is already available at the time when the gradients are to be calculated.

3. Function FUNCS

Function subprogram FUNCS defines upper or lower specified function $S_u(x)$ or $S_\ell(x)$, respectively, in various intervals. Write function FUNCS as follows :

FUNCTION FUNCS (X, IINT)

where X and IINT are both input variables representing a discrete point and a current interval, respectively.

Note If the upper and lower specified functions are defined for the same set of the independent parameter x, consider the common interval (or subinterval) twice.

4. Function W

Function subprogram W defines an upper and lower positive weighting function $w_u(x)$ and $w_\ell(x)$, respectively, in various intervals.

Write subprogram W as follows:

FUNCTION W(X, IINT)

where X is a discrete point in the IINTth interval.

5. FMLPO package

A listing is appended to this manual.

6. Data deck

Parameters to be supplied as data are defined below:

K	The number of independent variable parameters $\underset{\sim}{a}$.
NINT	The total number of upper and lower intervals.
NUMB(I), I=1, NINT	Number of subintervals in each interval of independent parameters.
XX(1,I), I=1, NINT	The left end point of ith interval.
XX(2,I), I=1, NINT	The right end point of ith interval.
XX(3,I), I=1, NINT	Numbers in floating point which supply information on the specified function in the ith interval: set XX(3,I)=1. for the upper specification and set XX(3,I)=-1. for the lower specification.
IREAD	Integer which denotes whether or not the discrete set of points in each interval will be read. If IREAD=0 the discrete set of points will be set equidistantly in each interval with NUMB(I) subintervals in the ith interval. If IREAD=1 the discrete point set will be read from data.

X(I), I=1, $\sum_{I=1}^{NINT}$ (NUMB(I)+1)	Discrete point set of the independent parameter.
KSI	The artificial margin ξ.
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 objective.
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 FMLPO.

Setting up the data deck is illustrated in Table 1.

Recommended values for some of the parameters

MAX = 100

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

DIF = 10^{-4}

EST A lower bound of the minimum value of the objective function may be obtained from physical reasons. If the true minimum is not known, for the case when the specification is violated EST=0 is convenient, and when the specification is satisfied choose EST sufficiently negative.

Comments

If the variable parameters are to be constrained, then each must have an associated lower and upper desired bound supplied by the user. Fictitious sample points are associated with each variable parameter in the correct sequence. The constraints are treated

TABLE 1

SETTING UP THE DATA DECK FOR FMLPO

Conditions	Number of cards	Parameters	Type	Format
-	1	K, NINT, IOPT, ITER, IREAD, IGRDCH	INTEGER	6I10
-	I=1, NINT	NUMB(I), (XX(J,I), J=1,3)	1 INTEGER, 3 REALS	I10, 3E16.8
IREAD=1	As many as required	X(J), J=1, (NUMB(I)+1)	REAL	5E16.8
-	by NUMB(I), I=1, NINT			
-	1	EST, DIF, KSI	REAL	3E16.8
-	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

exactly like single point specifications with one specification related to one fictitious point. For a single point specification, the number of subintervals is zero and the upper bound is equal to the lower bound.

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. Recommendation: start with 2, increase to 10 then to 100, etc., as needed. Optimization for larger values of p starts automatically at the optimum of the previous optimization unless otherwise specified.

The program terminates when stopping criteria for the Fletcher-Powell or Fletcher method are satsified 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 FMLPO requires the CDC system routine SECOND which keeps track of elapsed time. For a different system the cards A111, A123, A131, A143, G17, G57, H20, H25 and H53 should be replaced by cards appropriate to the system or removed together with cards A125, A145, G58, H26 and H54.

Input-output Example

An example, the same as the Example 3 in the paper [4] but with constraints on parameter a_2 such that

$$0 \leq a_2 \leq 2$$

shows how to set the user's written subprograms and the data deck. The Fletcher-Powell optimization method is called.

The user's written listing is shown in Fig. 1. The typical output of FMLPO for the **example when p=2 is shown in Fig. 2.**

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

              M A I N   P R O G R A M

      DIMENSION A(3),G(3),Y(3),PY(3),ASTRT(3),DUM1(3),DUM2(3),EPS(3),
     1 GRAD(3),H(15),XX(3,4),X(106),X1(106),NUMB(106),ERROR(106),
     2 EHELP(106),AP(106),INUMB(106),IPA(5)
      CALL EMLPO(A,ASTRT,G,Y,PY,DUM1,DUM2,EPS,H,GRAD,NUMB,XX,
     *X,X1,ERROR,EHELP,AP,INUMB,IPA)
      CALL EXIT
      END

 ..................................................................

      SUBROUTINE FCTAPP(X,K,A,APP,GRAD,IINT,INDIC)

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

      DIMENSION A(1),GRAD(1)
      GO TO(100,200),INDIC
  100 GO TO (1,1,2,2),IINT
    1 APP=A(3)/A(2)*EXP(-A(1)*X)*SIN(A(2)*X)
      RETURN
    2 APP=A(2)
      RETURN
  200 GO TO (3,3,4,4),IINT
    3 HP1=1./A(2)*EXP(-A(1)*X)
      HP2=HP1*SIN(A(2)*X)
      HP3=HP1*COS(A(2)*X)
      GRAD(1)=-APP*X
      GRAD(2)=-1./A(2)*APP+A(3)*X*HP3
      GRAD(3)=HP2
      RETURN
    4 GRAD(1)=0.
      GRAD(2)=1.
      GRAD(3)=0.
      RETURN
      END

 ..................................................................

      FUNCTION FUNCS(X,IINT)

          FUNCTION SUBPROGRAM WHICH DEFINES
          UPPER OR LOWER
          SPECIFIED FUNCTION

      GO TO(1,1,2,2),IINT
    1 FUNCS=3./20.*EXP(-X)+1./52.*EXP(-5.*X)-EXP(-2.*X)/65.*(3.*SIN
     *(2.*X)+11*COS(2.*X))
      RETURN
```

Fig. 1

```
2 FUNCS=2.0
  RETURN
3 FUNCS=0.0
  RETURN
  END
```

....

```
      FUNCTION W (X,IINT)
C
C        FUNCTION SUBPROGRAM WHICH DEFINES
C        UPPER OR LOWER
C        WEIGHTING FUNCTION
C
      W=1.
      RETURN
      END
```

....

```
    3          4          1          5          0          0
   50       0.0E     00      10.0E     00       1.0E     00
   50       0.0E     00      10.0E     00      -1.0E     00
    0       2.0E      1       2.0E      1       1.0E     00
    0       3.0E      1       3.0E      1      -1.0E     00
-1.0E     00       1.0E     -4       20.0E     -3
 1.0E     00       1.0E     00        1.0E     00
    2        100          1          1
 1.0E     -6
    2         10        100       1000      10000
```

32334B WORDS WERE REQUIRED FOR LOADING

Fig. 1 (continued)

KSI= 2.000000000000E-02

N	INDEPENDENT VARIABLE	ERRORS
	INTERVAL 1	
1	0.	-2.000000000000E-02
2	2.000000000000E-01	1.293042715288E-01
3	4.000000000000E-01	2.057388072794E-01
4	6.000000000000E-01	2.380295548278E-01
5	8.000000000000E-01	2.428939652327E-01
6	1.000000000000E+00	2.303971101044E-01
7	1.200000000000E+00	2.070054636670E-01
8	1.400000000000E+00	1.772456669985E-01
9	1.600000000000E+00	1.445232237326E-01
10	1.800000000000E+00	1.114740399353E-01
11	2.000000000000E+00	8.009309202347E-02
12	2.200000000000E+00	5.178539154951E-02
13	2.400000000000E+00	2.741231059152E-02
14	2.600000000000E+00	7.359508533869E-03
15	2.800000000000E+00	-8.373270500358E-03
16	3.000000000000E+00	-2.007130756018E-02
17	3.200000000000E+00	-2.820558335749E-02
18	3.400000000000E+00	-3.334496516618E-02
19	3.600000000000E+00	-3.608565405246E-02
20	3.800000000000E+00	-3.699971275447E-02
21	4.000000000000E+00	-3.660160908570E-02
22	4.200000000000E+00	-3.533000979455E-02
23	4.400000000000E+00	-3.354136379496E-02
24	4.600000000000E+00	-3.151182352526E-02
25	4.800000000000E+00	-2.944447602499E-02
26	5.000000000000E+00	-2.747945956178E-02
27	5.200000000000E+00	-2.570517106166E-02
28	5.400000000000E+00	-2.416933493606E-02
29	5.600000000000E+00	-2.288916575607E-02
30	5.800000000000E+00	-2.186020832079E-02
31	6.000000000000E+00	-2.106368933096E-02
32	6.200000000000E+00	-2.047238288648E-02
33	6.400000000000E+00	-2.005509639810E-02
34	6.600000000000E+00	-1.977994133628E-02
35	6.800000000000E+00	-1.961657848935E-02
36	7.000000000000E+00	-1.953763080438E-02
37	7.200000000000E+00	-1.951944637326E-02
38	7.400000000000E+00	-1.954237542472E-02
39	7.600000000000E+00	-1.959070233784E-02
40	7.800000000000E+00	-1.965234952081E-02
41	8.000000000000E+00	-1.971844641025E-02
42	8.200000000000E+00	-1.978283508039E-02
43	8.400000000000E+00	-1.984156474243E-02
44	8.600000000000E+00	-1.989241111931E-02
45	8.800000000000E+00	-1.993444337506E-02
46	9.000000000000E+00	-1.996765083509E-02
47	9.200000000000E+00	-1.999263389543E-02
48	9.400000000000E+00	-2.001035794489E-02
49	9.600000000000E+00	-2.002196544131E-02
50	9.800000000000E+00	-2.002863911642E-02
51	1.000000000000E+01	-2.003150828050E-02

Fig. 2

INTERVAL 2

1	0.	2.000000000000E-02
2	2.000000000000E-01	1.693042715288E-01
3	4.000000000000E-01	2.457388072794E-01
4	6.000000000000E-01	2.780295548278E-01
5	8.000000000000E-01	2.828939652327E-01
6	1.000000000000E+00	2.703971101044E-01
7	1.200000000000E+00	2.470054636670E-01
8	1.400000000000E+00	2.172456669985E-01
9	1.600000000000E+00	1.845232237326E-01
10	1.800000000000E+00	1.514740399353E-01
11	2.000000000000E+00	1.200930920235E-01
12	2.200000000000E+00	9.178539154951E-02
13	2.400000000000E+00	6.741231059152E-02
14	2.600000000000E+00	4.735950853387E-02
15	2.800000000000E+00	3.162672949964E-02
16	3.000000000000E+00	1.992869243982E-02
17	3.200000000000E+00	1.179441664251E-02
18	3.400000000000E+00	6.655034833816E-03
19	3.600000000000E+00	3.914345947541E-03
20	3.800000000000E+00	3.000287245528E-03
21	4.000000000000E+00	3.398390914304E-03
22	4.200000000000E+00	4.669990205447E-03
23	4.400000000000E+00	6.458636205035E-03
24	4.600000000000E+00	8.488176474741E-03
25	4.800000000000E+00	1.055552397501E-02
26	5.000000000000E+00	1.252054043822E-02
27	5.200000000000E+00	1.429482893834E-02
28	5.400000000000E+00	1.583066596394E-02
29	5.600000000000E+00	1.711083424393E-02
30	5.800000000000E+00	1.813979167921E-02
31	6.000000000000E+00	1.893631066904E-02
32	6.200000000000E+00	1.952761711352E-02
33	6.400000000000E+00	1.994490360190E-02
34	6.600000000000E+00	2.022005866372E-02
35	6.800000000000E+00	2.038342151065E-02
36	7.000000000000E+00	2.046236919562E-02
37	7.200000000000E+00	2.048055362674E-02
38	7.400000000000E+00	2.045762457528E-02
39	7.600000000000E+00	2.040929766216E-02
40	7.800000000000E+00	2.034765047919E-02
41	8.000000000000E+00	2.028155358975E-02
42	8.200000000000E+00	2.021716491961E-02
43	8.400000000000E+00	2.015843525757E-02
44	8.600000000000E+00	2.010758888069E-02
45	8.800000000000E+00	2.006555662494E-02
46	9.000000000000E+00	2.003234916491E-02
47	9.200000000000E+00	2.000736610457E-02
48	9.400000000000E+00	1.998964205511E-02
49	9.600000000000E+00	1.997803455869E-02
50	9.800000000000E+00	1.997136088358E-02
51	1.000000000000E+01	1.996849171950E-02

INTERVAL 3

1	2.000000000000E+01	-1.020000000000E+00

INTERVAL 4

1	3.000000000000E+01	1.020000000000E+00

Fig. 2 (continued)

VALUE OF MAXIMUM ERROR

2.428939652327E-01

		ERROR USED IN OBJECTIVE FUNCTION
1	2.000000000000E-01	1.293042715288E-01
2	4.000000000000E-01	2.057388072794E-01
3	6.000000000000E-01	2.380295548278E-01
4	8.000000000000E-01	2.428939652327E-01
5	1.000000000000E+00	2.303971101044E-01
6	1.200000000000E+00	2.070054636670E-01
7	1.400000000000E+00	1.772456669985E-01
8	1.600000000000E+00	1.445232237326E-01
9	1.800000000000E+00	1.114740399353E-01
10	2.000000000000E+00	8.009309202347E-02
11	2.200000000000E+00	5.178539154951E-02
12	2.400000000000E+00	2.741231059152E-02
13	2.600000000000E+00	7.359508533869E-03

INPUT DATA

FOLLOWING METHODS HAVE BEEN CALLED

FLETCHER-POWELL METHOD

NUMBER OF INDEPENDENT VARIABLES..N= 3

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

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
ASTRT(3)= 1.00000000E+00

TEST QUANTITY TO BE USED IN FLETCHER-POWELL METHOD..............EPS1= 1.00000000E-06

ESTIMATE OF LOWER BOUND ON FUNCTION TO BE MINIMIZED..............FST= -1.00000000E+00

Fig. 2 (continued)

OPTIMIZATION BY FLETCHER-POWELL METHOD

ITERATION NUMBER	FUNCTION EVALUATIONS	TIME ELAPSED (SECONDS)	OBJECTIVE FUNCTION	VARIABLE VECTOR A(I)	GRADIENT VECTOR G(I)
0	1	1.82000000E-01	5.87596262E-01	1.00000000E+00 1.00000000E+00 1.00000000E+00	-7.61720864E-01 -3.63249450E-01 7.87723700E-01
1	3	5.45000000E-01	1.39841486E-02	1.66173265E+00 1.31556707E+00 3.15677808E-01	-3.19618675E-03 1.17139365E-02 1.74464105E-01
2	5	8.73000000E-01	1.09542831E-02	1.66274394E+00 1.31323339E+00 2.78224101E-01	3.80269059E-02 3.48032998E-02 -3.36827649E-02
3	12	2.04800000E+00	5.09958114E-03	1.18203682E+00 7.92022694E-01 2.37724444E-01	-1.98822017E-02 -1.79286424E-03 2.72262314E-01
4	17	3.10000000E+00	-1.09633746E-03	1.00355372E+00 5.98761326E-01 1.90398834E-01	-1.04301563E-02 -4.19318874E-03 4.76551556E-02
5	20	3.88600000E+00	-1.30118262E-03	1.06397435E+00 6.61687650E-01 1.94698742E-01	-3.21087499E-03 -6.04475834E-04 5.43066827E-02
6	27	5.75400000E+00	-1.87768358E-03	1.08429008E+00 6.79643889E-01 1.61073963E-01	2.51169089E-04 2.36039190E-05 -7.43087638E-05
7	32	7.07300000E+00	-1.87891792E-03	1.07535169E+00 6.71170222E-01 1.61260557E-01	6.80078032E-05 -5.28782753E-05 8.56739636E-04
8	39	8.94200000E+00	-1.88070667E-03	1.06102484E+00 6.58202233E-01 1.57375865E-01	6.79255526E-05 -8.88710885E-05 4.59720127E-05
9	43	9.99600000E+00	-1.88073328E-03	1.06203794E+00 6.59545964E-01 1.57316828E-01	9.64156316E-05 -7.69871630E-05 -9.77493616E-05
10	56	1.34170000E+01	-1.89502834E-03	9.44238980E-01 8.25425060E-01 1.47636790E-01	1.77479926E-04 7.07537866E-05 -9.41538460E-04
11	64	1.55640000E+01	-1.89531153E-03	9.28499148E-01 8.43544566E-01 1.46622062E-01	1.54612783E-04 8.69375066E-05 -8.61970379E-04
12	74	1.82350000E+01	-1.89628533E-03	9.22881551E-01 8.33515308E-01 1.46849660E-01	3.07434642E-06 -3.45927297E-06 -7.63184321E-05
13	78	1.92980000E+01	-1.89630042E-03	9.23533453E-01 8.35379711E-01 1.47186573E-01	5.71760617E-07 5.62582738E-07 -4.21956934E-06

Fig. 2 (continued)

```
14        84     2.[illegible]E+01     -1.89630047E-03        9.23682354E-01     -3.33535869E-08
                                                              8.35165085E-01     -1.25275309E-08
                                                              1.47203632E-01      1.33521708E-07

15        84     2.[illegible]E+01     -1.89630047E-03        9.23688957E-01      3.69837765E-13
                                                              8.35161409E-01      2.10977528E-11
                                                              1.47204064E-01      1.74031901E-10

IER= 0 CRITERION FOR OPTIMUM HAS BEEN SATISFIED


                         FOLLOWING IS THE OPTIMUM SOLUTION
                         ---------------------------------

                              F = -1.89630047E-03

                           A( 1)=   9.23688957E-01
                           A( 2)=   8.35161400E-01
                           A( 3)=   1.47204063E-01

              NUMBER OF FUNCTION EVALUATIONS BY THE FLETCHER-POWELL METHOD        89

              EXECUTION TIME IN SECONDS    22.25600
```

Fig. 2 (continued)

Q = -2

N	INDEPENDENT VARIABLE	ERRORS
	INTERVAL 1	
1	0.	-2.000000000000E-02
2	2.000000000000E-01	-8.991265410800E-03
3	4.000000000000E-01	-1.535588915954E-02
4	6.000000000000E-01	-2.320653330371E-02
5	8.000000000000E-01	-2.728167220791E-02
6	1.000000000000E+00	-2.727657596876E-02
7	1.200000000000E+00	-2.469459019274E-02
8	1.400000000000E+00	-2.124535435066E-02
9	1.600000000000E+00	-1.818074190609E-02
10	1.800000000000E+00	-1.615282641224E-02
11	2.000000000000E+00	-1.531733402275E-02
12	2.200000000000E+00	-1.551434511224E-02
13	2.400000000000E+00	-1.643779261072E-02
14	2.600000000000E+00	-1.775828843492E-02
15	2.800000000000E+00	-1.919451590163E-02
16	3.000000000000E+00	-2.054293946449E-02
17	3.200000000000E+00	-2.167984915326E-02
18	3.400000000000E+00	-2.254855350585E-02
19	3.600000000000E+00	-2.314115103000E-02
20	3.800000000000E+00	-2.348068296348E-02
21	4.000000000000E+00	-2.360651980516E-02
22	4.200000000000E+00	-2.356382919284E-02
23	4.400000000000E+00	-2.339683132017E-02
24	4.600000000000E+00	-2.314504822034E-02
25	4.800000000000E+00	-2.284165723916E-02
26	5.000000000000E+00	-2.251317287134E-02
27	5.200000000000E+00	-2.217987188547E-02
28	5.400000000000E+00	-2.185656720653E-02
29	5.600000000000E+00	-2.155349085437E-02
30	5.800000000000E+00	-2.127715617638E-02
31	6.000000000000E+00	-2.103113934598E-02
32	6.200000000000E+00	-2.081675960684E-02
33	6.400000000000E+00	-2.063365734449E-02
34	6.600000000000E+00	-2.048027711059E-02
35	6.800000000000E+00	-2.035426497960E-02
36	7.000000000000E+00	-2.025278954275E-02
37	7.200000000000E+00	-2.017279519954E-02
38	7.400000000000E+00	-2.011119588115E-02
39	7.600000000000E+00	-2.006501705095E-02
40	7.800000000000E+00	-2.003149365326E-02
41	8.000000000000E+00	-2.000813150714E-02
42	8.200000000000E+00	-1.999273925284E-02
43	8.400000000000E+00	-1.998343750167E-02
44	8.600000000000E+00	-1.997365111583E-02
45	8.800000000000E+00	-1.997708969080E-02
46	9.000000000000E+00	-1.997772073522E-02
47	9.200000000000E+00	-1.997973863949E-02
48	9.400000000000E+00	-1.998253243415E-02
49	9.600000000000E+00	-1.998565407452E-02
50	9.800000000000E+00	-1.998878866578E-02
51	1.000000000000E+01	-1.999172746792E-02

Fig. 2 (continued)

INTERVAL 2

1	0.	2.000000000000E-02
2	2.000000000000E-01	3.100873458920E-02
3	4.000000000000E-01	2.454411084046E-02
4	6.000000000000E-01	1.679346669669E-02
5	8.000000000000E-01	1.271832779209E-02
6	1.000000000000E+00	1.272342403124E-02
7	1.200000000000E+00	1.530540980726E-02
8	1.400000000000E+00	1.875464564934E-02
9	1.600000000000E+00	2.181925809391E-02
10	1.800000000000E+00	2.384717358776E-02
11	2.000000000000E+00	2.468266597725E-02
12	2.200000000000E+00	2.448565488776E-02
13	2.400000000000E+00	2.356220738928E-02
14	2.600000000000E+00	2.224171156508E-02
15	2.800000000000E+00	2.080548409837E-02
16	3.000000000000E+00	1.945706053551E-02
17	3.200000000000E+00	1.832015084674E-02
18	3.400000000000E+00	1.745144649415E-02
19	3.600000000000E+00	1.685884897000E-02
20	3.800000000000E+00	1.651971703652E-02
21	4.000000000000E+00	1.639348019484E-02
22	4.200000000000E+00	1.643617080716E-02
23	4.400000000000E+00	1.660316867983E-02
24	4.600000000000E+00	1.685495177966E-02
25	4.800000000000E+00	1.715874275084E-02
26	5.000000000000E+00	1.748682712866E-02
27	5.200000000000E+00	1.782012811453E-02
28	5.400000000000E+00	1.814343279347E-02
29	5.600000000000E+00	1.844650914563E-02
30	5.800000000000E+00	1.872284382362E-02
31	6.000000000000E+00	1.896886065402E-02
32	6.200000000000E+00	1.918324039316E-02
33	6.400000000000E+00	1.936634265551E-02
34	6.600000000000E+00	1.951972288941E-02
35	6.800000000000E+00	1.964573502040E-02
36	7.000000000000E+00	1.974721045725E-02
37	7.200000000000E+00	1.982720480046E-02
38	7.400000000000E+00	1.988880411885E-02
39	7.600000000000E+00	1.993498294995E-02
40	7.800000000000E+00	1.996850634674E-02
41	8.000000000000E+00	1.999186849286E-02
42	8.200000000000E+00	2.000726074716E-02
43	8.400000000000E+00	2.001656249833E-02
44	8.600000000000E+00	2.002134889417E-02
45	8.800000000000E+00	2.002291030920E-02
46	9.000000000000E+00	2.002227926478E-02
47	9.200000000000E+00	2.002026136051E-02
48	9.400000000000E+00	2.001746756585E-02
49	9.600000000000E+00	2.001434592548E-02
50	9.800000000000E+00	2.001121133422E-02
51	1.000000000000E+01	2.000827253208E-02

INTERVAL 3

1	2.000000000000E+01	-1.184838599728E+00

INTERVAL 4

1	3.000000000000E+01	8.551614002724E-01

Fig. 2 (continued)

LISTING OF FMLPO

```
      SUBROUTINE FMLPO (A,XSTRT,G,Y,PY,DUM1,DUM2,EPS,H,GRAD,NUMB,XX,X,X1
     1,ERROR,EHELP,AP,INUMB,IPA)
C
C        SUBROUTINE WHICH COORDINATES THE OTHER
C        SUBROUTINES IN THE PACKAGE FMLPO
C
      EXTERNAL FUNCS,W,FCT,FUNGT
      LOGICAL CONV,UNITH
      DIMENSION A(1), G(1), Y(1), PY(1), XSTRT(1), DUM1(1), DUM2(1), EPS
     1(1), H(1), GRAD(1), NUMB(1), XX(3,1), X(1), X1(1), ERROR(1), EHELP
     2(1), AP(1), INUMB(1), IPA(1)
      COMMON T1,KO,NFE
      ERR(Z)=EPSNP(Z,IINT,FCT,W,A,N1,GRAD,APP,PSI,XX,1)
      UNITH=.TRUE.
      T1=0.
      READ (5,38) N1,NINT,IOPT,ITER,IREAD,IGRDCH
      DO 1 I=1,NINT
      READ (5,40) NUMB(I),(XX(J,I),J=1,3)
    1 CONTINUE
      K=0
      IF (IREAD.EQ.0) IREAD=2
      GO TO (2,4), IREAD
    2 DO 3 J=1,NINT
      K=K+1
      KL=K+NUMB(J)
      READ (5,42) (X(I),I=K,KL)
      K=KL
    3 CONTINUE
    4 READ (5,42) EST,DIF,PSI
      WRITE (6,39) PSI
      WRITE (6,46)
      READ (5,42) (XSTRT(I),I=1,N1)
      DO 5 I=1,N1
      A(I)=XSTRT(I)
    5 CONTINUE
      K=0
      GO TO (6,9), IREAD
    6 DO 8 J=1,NINT
      IINT=J
      WRITE (6,51) IINT
      K=K+1
      KL=K+NUMB(J)
      DO 7 I=K,KL
      ERROR(I)=ERR(X(I))
      L=I-K+1
      WRITE (6,50) L,X(I),ERROR(I)
      EHELP(I)=ERROR(I)*XX(3,J)
      AP(I)=APP
    7 CONTINUE
      K=KL
    8 CONTINUE
      GO TO 11
    9 DO 10 J=1,NINT
      IINT=J
      WRITE (6,51) IINT
      L=NUMB(J)+1
      IF (NUMB(J).EQ.0) Z=XX(1,J)
      DO 10 I=1,L
      IF (NUMB(J).GT.0) Z=XX(1,J)+(XX(2,J)-XX(1,J))*(I-1)/NUMB(J)
```

```
      ER=ERR(Z)
      WRITE (6,50) I,Z,ER
      K=K+1
      ERROR(K)=ERR(Z)
      EHELP(K)=ERROR(K)*XX(3,J)
      X(K)=Z
10    AP(K)=APP
11    EMAX=EHELP(1)
      DO 12 M=2,K
      EMAX=AMAX1(EMAX,EHELP(M))
12    CONTINUE
      WRITE (6,48)
      WRITE (6,49)
      WRITE (6,47) EMAX
      CALL ERRO (FCT,W,A,N1,K,GRAD,APP,PSI,2,NUMB,XX,X,X1,ERROR,EHELP,AP
     1,EMAX,N,INUMB,NINT,IP)
      WRITE (6,44)
      WRITE (6,45)
      WRITE (6,50) (J,X1(J),ERROR(J),J=1,N)
C
C          DATA FOR THE OPTIMALITY
C          FOR THE OPTIMIZATION METHOD USED
C
      DO 37 K=1,IOPT
      KR=1
      IF (K-1) 14,14,13
13    READ (5,42) (XSTRT(I),I=1,N1)
14    READ (5,38) MET,MAX,IPRINT,IDATA
      IF (MET.EQ.1) READ (5,42) (EPS(I),I=1,N1)
      IF (MET.EQ.2) READ (5,42) EPS1
      READ (5,38) (IPA(I),I=1,ITER)
      DO 36 KK=1,ITER
C
C          OPTIMIZATION
C
      IP=IPA(KK)
      IF (KK.GT.1) FF=F
      IF (KR.EQ.0) GO TO 15
15    DO 16 I=1,N1
      A(I)=XSTRT(I)
16    CONTINUE
      IF (IGRDCH.NE.1) GO TO 17
      CALL GRDCHK (N1,A,G,PY,Y,GRAD,APP,PSI,NUMB,XX,X,X1,ERROR,EHELP,AP,
     1EMAX,N,INUMB,NINT,IP,DUM1)
17    IF (KR.EQ.0) GO TO 18
      IF (IDATA.EQ.0) GO TO 18
      CALL INPUT (MET,M,MAX,N1,IPRINT,IDATA,EPS1,EST,EPS,XSTRT)
18    IF (MET.EQ.0) MET=4
      INDEX=0
      GO TO (19,25,32,31), MET
19    CONTINUE
      CALL SECOND (T1)
      IF (IPRINT.EQ.0) GO TO 20
      CALL WRITE1 (1)
20    IF (KR.NE.0) GO TO 22
      DO 21 I=1,N1
      A(I)=DUM1(I)
21    CONTINUE
22    CALL EMNEG (N1,A,F,G,H,UNITH,EST,EPS,MAX,IPRINT,IEXIT,GRAD,APP,PSI
```

```
     1,NUMB,XX,X,X1,ERROR,EHELP,AP,EMAX,N,INUMB,NINT,IP)
      DO 23 I=1,N1
      DUM1(I)=A(I)
23    CONTINUE
      CALL SECOND (T2)
      CALL FINAL (A,F,N1,MET)
      T=T2-T1
      IF (T1.EQ.0.) GO TO 24
      WRITE (6,41) T
24    CONTINUE
      GO TO 31
25    CONTINUE
      CALL SECOND (T1)
      IF (IPRINT.EQ.0) GO TO 26
      CALL WRITE1 (2)
26    IF (KR.NE.0) GO TO 28
      DO 27 I=1,N1
      A(I)=DUM2(I)
27    CONTINUE
28    CALL EMFPG (FUNGT,N1,A,F,G,EST,EPS1,MAX,IER,H,IPRINT,GRAD,APP,PSI,
     1NUMB,XX,X,X1,ERROR,EHELP,AP,EMAX,N,INUMB,NINT,IP)
      DO 29 I=1,N1
      DUM2(I)=A(I)
29    CONTINUE
      CALL SECOND (T2)
      CALL FINAL (A,F,N1,MET)
      T=T2-T1
      IF (T1.EQ.0.) GO TO 30
      WRITE (6,41) T
30    CONTINUE
31    INDEX=INDEX+1
C
32    KR=0
      WRITE (6,43) IP
      WRITE (6,46)
      KN=0
      KQ=0
      DO 34 J=1,NINT
      IINT=J
      WRITE (6,51) IINT
      KQ=KQ+1
      KL=KQ+NUMB(J)
      DO 33 I=KQ,KL
      L=I-KQ+1
      ER=ERR(X(I))
      WRITE (6,50) L,X(I),ER
      KN=KN+1
33    CONTINUE
      KQ=KL
34    CONTINUE
      WRITE (6,44)
      WRITE (6,45)
      WRITE (6,50) (J,X1(J),ERROR(J),J=1,N)
      WRITE (6,48)
      WRITE (6,49)
      WRITE (6,47) EMAX
      IGRDCH=IGRDCH+2
      IF (KK-1) 36,36,35
35    FTST=ABS((FF-F)/FF)
```

```
      IF (FTST.LT.DIF) GO TO 37
36    CONTINUE
37    CONTINUE
      RETURN
C
C
38    FORMAT (8I10)
39    FORMAT (1H1,20X,4HKSI=,E23.12/////)
40    FORMAT (I10,3F16.8)
41    FORMAT (1H0,//25X,26HEXECUTION TIME IN SECONDS ,F10.5)
42    FORMAT (5F16.8)
43    FORMAT (1H1,19X,4HQ  =,I7/////)
44    FORMAT (//41X,13HERROR USED IN)
45    FORMAT (40X,18HOBJECTIVE FUNCTION/)
46    FORMAT (8X,1HN,6X,20HINDEPENDENT VARIABLE,8X,6HERRORS)
47    FORMAT (13X,E20.12)
48    FORMAT (//20X,8HVALUE OF)
49    FORMAT (18X,13HMAXIMUM ERROR/)
50    FORMAT (I9,3X,2F23.12)
51    FORMAT (/20X,9HINTERVAL ,I2/)
      END
```

••

```
      FUNCTION FCT (Z,FUNCS,W,IINT,PSI,XX)
C
C         FUNCTION SUBPROGRAM WHICH DEFINES
C         MODIFIED UPPER AND LOWER
C         SPECIFIED FUNCTION
C
      EXTERNAL FUNCS,W
      DIMENSION XX(3,1)
      FCT=FUNCS(Z,IINT)+PSI*XX(3,IINT)/W(Z,IINT)
      RETURN
      END
```

••

```
      FUNCTION FPSNP (Z,IINT,FCT,W,A,N1,GRAD,APP,PSI,XX,IPOINT)
C
C         FUNCTION SUBPROGRAM WHICH CALCULATES
C         UPPER AND LOWER WEIGHTED ERROR FUNCTION
C
      EXTERNAL FUNCS,W,FCT
      DIMENSION A(1), GRAD(1), XX(3,1)
      IF (IPOINT) 1,2,1
1     CONTINUE
      CALL FCTAPP (Z,N1,A,APP,GRAD,IINT,1)
2     CONTINUE
      IF (PSI) 3,4,3
3     EPSNP=(APP-FCT(Z,FUNCS,W,IINT,PSI,XX))*W(Z,IINT)
      RETURN
4     FPSNP=(APP-FUNCS(Z,IINT))*W(Z,IINT)
```

```
      RETURN
      END
```

••

```
      SUBROUTINE ERRO (FCT,W,A,N1,K,GRAD,APP,PSI,INDIC,NUMB,XX,X,X1,ERRO
     1R,EHELP,AP,EMAX,N,INUMB,NINT,IP)
C
C         SUBROUTINE WHICH SELECTS THE WEIGHTED
C         ERROR FUNCTION OF INTEREST FOR
C         THE OBJECTIVE FUNCTION
C
      EXTERNAL FUNCS,W,FCT
      DIMENSION A(1), GRAD(1), NUMB(1), XX(3,1), X(1), X1(1), ERROR(1),
     1EHELP(1), AP(1), INUMB(1)
      ERR(Z)=EPSNP(Z,IINT,FCT,W,A,N1,GRAD,APP,PSI,XX,IPOINT)
      GO TO (1,9), INDIC
    1 CONTINUE
      IPOINT=1
      K=0
      KL=0
      DO 7 J=1,NINT
      IINT=J
      IF (J.EQ.1) GO TO 2
      KL=KL+L
    2 L=NUMB(J)+1
      DO 6 I=1,L
      K=K+1
      IF (J.EQ.1) GO TO 5
      DO 4 KK=1,KL
      IF (X(K)-X(KK)) 4,3,4
    3 AP(K)=AP(KK)
      APP=AP(K)
      IPOINT=0
      GO TO 5
    4 CONTINUE
    5 ERROR(K)=ERR(X(K))
      EHELP(K)=ERROR(K)*XX(3,J)
      IF (IPOINT.NE.0) AP(K)=APP
      IPOINT=1
    6 CONTINUE
    7 CONTINUE
      EMAX=EHELP(1)
      DO 8 M=2,K
      EMAX=AMAX1(EMAX,EHELP(M))
    8 CONTINUE
    9 CONTINUE
      IF (EMAX) 10,11,11
   10 IP=-IABS(IP)
      GO TO 12
   11 IP=IABS(IP)
   12 K=0
      N=0
      INUMB(1)=0
      DO 16 J=1,NINT
      IINT=J
      L=NUMB(J)+1
```

```
      DO 15 I=1,L
      K=K+1
      IF (IP) 14,13,13
13    IF (EHELP(K)) 15,14,14
14    N=N+1
      X1(N)=X(K)
      ERROR(N)=ERROR(K)
      EHELP(N)=AP(K)
15    CONTINUE
      INUMB(J+1)=N
16    CONTINUE
      RETURN
      END
```

....

```
      SUBROUTINE FUNGT (N1,A,OBJ,G,GRAD,APP,PSI,NUMB,XX,X,X1,ERROR,EHELP
     1,AP,EMAX,N,INUMB,NINT,IP)
C
C          SUBROUTINE WHICH COMPUTES THE OBJECTIVE FUNCTION
C          AND ITS GRADIENTS W.R.T. THE VARIABLE PARAMETERS
C          IN THE LEAST P-TH SENSE
C
      EXTERNAL FUNCS,W,FCT
      DIMENSION A(1), GRAD(1), NUMB(1), XX(3,1), X(1), X1(1), ERROR(1),
     1EHELP(1), AP(1), INUMB(1), G(1)
      OBJP=0.
      GRADP=0.
      DO 1 K=1,N1
      G(K)=0.
1     CONTINUE
      CALL ERRO (FCT,W,A,N1,K,GRAD,APP,PSI,1,NUMB,XX,X,X1,ERROR,EHELP,AP
     1,EMAX,N,INUMB,NINT,IP)
      DO 7 I=1,N
      Z=X1(I)
      DEL=ERROR(I)/EMAX
      OBJI=DEL**IP
      GRADI=DEL**(IP-1)
      OBJP=OBJP+OBJI
      DO 4 J=1,NINT
      IF (I-INUMB(J+1)) 2,2,4
2     IF (I-INUMB(J)) 4,4,3
3     IINT=J
      GO TO 5
4     CONTINUE
5     CONTINUE
      APP=EHELP(I)
      CALL FCTAPP (Z,N1,A,APP,GRAD,IINT,2)
      DO 6 K=1,N1
      GRAD(K)=GRADI*W(Z,IINT)*GRAD(K)
      G(K)=G(K)+GRAD(K)
6     CONTINUE
7     CONTINUE
      PR=1./IP
      OBJ=EMAX*(OBJP**PR)
      GRP=OBJP**(PR-1.)
```

```
      DO 8 K=1,N1
      G(K)=GRP*G(K)
    8 CONTINUE
      RETURN
      END
```

••

```
      SUBROUTINE GPDGHK (N,A,G,PY,Y,GRAD,APP,PSI,NUMB,XX,XP,X1,ERROR,EHE
     1LP,AP,EMAX,NP,INUMB,NINT,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), NUMB(1), XX(3,1), XP(1
     1), X1(1), ERROR(1), EHELP(1), AP(1), INUMB(1), DUM1(1)
      CALL FUNGT (N,A,F,G,GRAD,APP,PSI,NUMB,XX,XP,X1,ERROR,EHELP,AP,EMAX
     1,NP,INUMB,NINT,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 FUNGT (N,A,FNEW,PY,GRAD,APP,PSI,NUMB,XX,XP,X1,ERROR,EHELP,AP,
     1EMAX,NP,INUMB,NINT,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 (1HO,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 (///,1HO,5X,20HANALYTICAL GRADIENTS,5X,19HNUMERICAL GRADIEN
     1TS,5X,16HPERCENTAGE ERROR,/)
```

```
12    FORMAT (1HO,5X,3(E16.8,9X))
13    FORMAT (1HO,//,6X,19HGRADIENTS ARE O. K.)
14    FORMAT (1HO,//,6X,64HYOUR PROGRAM HAS BEEN TERMINATED BECAUSE GRAD
     1IENTS ARE INCORRECT,/6X,21HPLEASE CHECK IT AGAIN)
      END
```

....

```
      SUBROUTINE FMNEG (N,X,F,G,H,UNITH,FEST,EPS,MAXFN,IPRINT,IFXIT,GRAD
     1,APP,PSI,NUMB,XX,XP,X1,ERROR,EHELP,AP,FMAX,NP,INUMB,NINT,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), NUMB(1), XX(3,1), XP(
     11), X1(1), ERROR(1), EHELP(1), AP(1), INUMB(1)
      LOGICAL CONV,UNITH
      COMMON T1,KO,NFNS
      CALL SECOND (T3)
      KO=0
      CALL FUNGT (N,X,F,G,GRAD,APP,PSI,NUMB,XX,XP,X1,ERROR,EHELP,AP,FMAX
     1,NP,INUMB,NINT,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 FUNGT (N,X,FP,H,GRAD,APP,PSI,NUMB,XX,XP,X1,ERROR,FHELP,AP,EMA
     1X,NP,INUMB,NINT,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


30    FORMAT (/,1H0,6HIEXIT=,I2,40HCRITERION FOR OPTIMUM HAS BEEN SATISF
     1IED)
31    FORMAT (/,1H0,6HIEXIT=,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 (/,1H0,6HIEXIT=,I2,55HMAXIMUM NUMBER OF ALLOWABLE ITERATION
     1 HAS BEEN EXCEEDED)
33    FORMAT (/,1H0,6HIEXIT=,I2,60HFUNCTION VALUE LESS THAN MINIMUM ESTI
     1MATED HAS BEEN DETECTED)
      END
```

••

```
      SUBROUTINE FMFPG (FUNGT,N,X,F,G,EST,EPS,LIMIT,IER,H,IPRINT,GRAD,AP
     1P,PSI,NUMB,XX,XP,X1,ERROR,EHELP,AP,EMAX,NP,INUMB,NINT,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
      DIMENSION H(1), X(1), G(1), GRAD(1), NUMB(1), XX(3,1), XP(1), X1(1
     1), ERROR(1), EHELP(1), AP(1), INUMB(1)
C
C        COMPUTE FUNCTION VALUE AND GRADIENT VECTOR FOR INITIAL ARGUMENT
      KO=0
      CALL SECOND (T3)
      CALL FUNGT (N,X,F,G,GRAD,APP,PSI,NUMB,XX,XP,X1,ERROR,EHELP,AP,EMAX
     1,NP,INUMB,NINT,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 FUNGT (N,X,F,G,GRAD,APP,PSI,NUMB,XX,XP,X1,ERROR,EHELP,AP,EMAX
     1,NP,INUMB,NINT,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)
      CONTINUF
C
C        TFRMINATF, IF THF VALUE OF THF ACTUAL FUNCTION AT X IS LESS
C        THAN THE FUNCTION VALUFS AT THE INTFRVAL ENDS. OTHFRWISF REDUCF
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 FUNGT (N,X,F,G,GRAD,APP,PSI,NUMB,XX,XP,X1,FRROR,FHFLP,AP,FMAX
     1,NP,INUMB,NINT,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        COMPUTF DIFFFRFNCE VFCTORS OF ARGUMFNT AND GRADIFNT FROM
C        TWO CONSECUTIVF ITFRATIONS
   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 CONTINUF
C
C        TFST LENGTH OF ARGUMFNT DIFFERENCE VECTOR AND DIRECTION VECTOR
C        IF AT LEAST N ITERATIONS HAVE BEEN EXECUTFD. TERMINATE, IF
C        BOTH ARE LESS THAN  EPS
      IFR=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-FPS) 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 FUNGT (N,X,F,G,GRAD,APP,PSI,NUMB,XX,XP,X1,ERROR,EHELP,AP,EMAX
     1,NP,INUMB,NINT,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 (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
```

....

```
      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) EST
      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 (1HO,15HFLETCHER METHOD)
7     FORMAT (1HO,22HFLETCHER-POWELL METHOD)
8     FORMAT (1HO,/1X,31HNUMBER OF INDEPENDENT VARIABLES,36(1H.),2HN=,I5
     1,/)
9     FORMAT (1HO,38HMAXIMUM NUMBER OF ALLOWABLE ITERATIONS,27(1H.),4HMA
     1X=,I5,/)
10    FORMAT (1HO,57HINTERMEDIATE OUTPUT TO BE PRINTED EVERY IPRINT ITER
     1ATIONS,5(1H.),7HIPRINT=,I5,/)
11    FORMAT (1HO,30HSTARTING VALUE FOR VECTOR A(I),29(1H.),10HASTRT( 1)
     1=,E16.8)
12    FORMAT (1HO,59X,6HASTRT(,I2,2H)=,E16.8)
13    FORMAT (1HO,/,1X,45HTEST QUANTITIES TO BE USED IN FLETCHER METHOD,
     116(1H.),8HEPS( 1)=,E16.8)
14    FORMAT (1HO,61X,4HEPS(,I2,2H)=,E16.8)
15    FORMAT (1HO,/,1X,50HTEST QUANTITY TO BE USED IN FLETCHER-POWELL ME
     1THOD,14(1H.),5HEPS1=,E16.8)
16    FORMAT (1HO,/,1X,51HESTIMATE OF LOWER BOUND ON FUNCTION TO BE MINI
     1MIZED,14(1H.),4HEST=,E16.8)
17    FORMAT (1HO,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,NFE
      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 =,F16.8,/)
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,NFF
      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 (1HO,31HOPTIMIZATION BY FLETCHER METHOD,/,1HO,31(1H-))
7     FORMAT (1HO,38HOPTIMIZATION BY FLETCHER-POWELL METHOD,/,1HO,38(1H-
     1))
8     FORMAT (1HO,9HITERATION,2X,8HFUNCTION,6X,12HTIME ELAPSED,8X,9HOBJE
     1CTIVE,14X,20HVARIABLE VECTOR A(I),9X,20HGRADIENT VECTOR G(I),/1HO,
     26HNUMBER,5X,11HEVALUATIONS,3X,9H(SECONDS),11X,8HFUNCTION,/)
9     FORMAT (1HO,9HITERATION,2X,8HFUNCTION,8X,9HOBJECTIVE,14X,20HVARIAB
     1LE VECTOR A(I),9X,20HGRADIENT VECTOR G(I),/1HO,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,NFF
      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
```