GRADNLP — **A PACKAGE FOR SOLVING NONLINEAR PROGRAMMING PROBLEMS USING A NEW (MINIMAX) APPROACH WITH EFFICIENT GRADIENT METHODS**

PURPOSE

GRADNLP is a slightly modified version of GRADMIN to be used for solving constrainted optimization problems. A new technique proposed by Bandler and Charalambous [1] is used to transform the constrained optimization problem into the minimization of an unconstrained objective function. The equality constraint must be treated as two inequality constraints, e.g., $\psi(x_1, x_2, \ldots, x_n) = 0$ will be treated as $\psi(x_1, x_2, \ldots, x_n) \geq 0$ and $-\psi(x_1, x_2, \ldots, x_n) \geq 0$.

AUTHORS

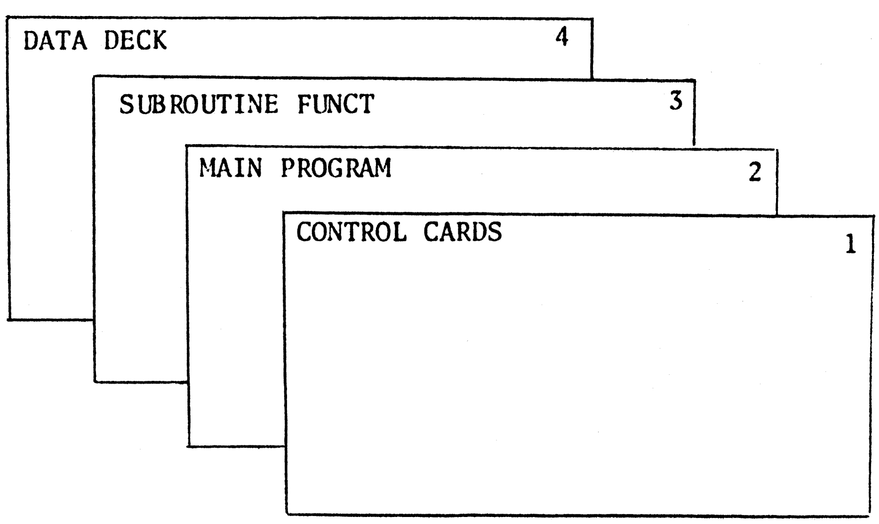
J.W. Bandler and V.K. Jha
Department of Electrical Engineering
McMaster University
Hamilton, Ontario, Canada

PROGRAMMER

V.K. Jha

HOW TO USE

Set up the input deck as follows (McMaster CDC 6400)



1. <u>CONTROL CARDS</u> Use the following set of control cards.

```
AAAA                                        USER NAME
LOADER(PPLOADR)
ATTACH(TAPE, GRADNLP, ID=*********, MR=1)†
RUN(S)
SETINDF
LOAD(TAPE)
REDUCE.
LGO.
END OF RECORD
PROGRAM TST(INPUT, OUTPUT, TAPE5=INPUT, TAPE6=OUTPUT)
```

2. <u>MAIN PROGRAM</u> Write the main program as indicated below

(a) Dimension the following arrays

X(NN), G(NN), X1(NN), X2(NN), G2(NN), PY(NN), ALFA(NN), P(NN,NN), Y(NN), PE(NN), BIGV(NN), XSTRT(NN), DUM1(NN), DUM2(NN), EPS(N), H(K)

where N = The number of independent variables

NN = N + 2

K = N*(N+7)/2

(b) Supply the values of the following parameters

N = The number of independent variables

NN = N + 2

KR = 1 if data deck is to be read

= 0 if data deck is not to be read always set it equal to 1 for the first call of GRADNLP.

---

† Appropriate identification parameter ID should be inserted in *********.

(c) Call subroutine GRADNLP as follows

```
CALL GRADNLP(N, X, XSTRT, X2, X1, G, G2, ALFA, H, P, PY, PE,
             B1GV, EPS, NN, Y, DUM1, DUM2, KR)
```

(d) Add STOP and END cards.

3. SUBROUTINE FUNCT This subroutine defines the actual objective function, the inequality constraints, and the gradients of the objective function and the constraints. A subroutine FMINIMAX is then called which combines the objective function and the constraints in a suitable manner to give the unconstrained objective function F and its gradients. The subroutine FUNCT should be written as follows. The actual numerical values of N and NCONS should be substituted in the dimension statements. NCONS is the number of inequality constraints.

SUBROUTINE FUNCT(N, X, F, G)

DIMENSION X(1), G(1), PHI(NCONS), GU(N), GPHI(N, NCONS)

U = $f(x_1, x_2,\ldots,x_n)$ [The actual objective function]

PHI(1) = $\phi_1(x_1, x_2,\ldots,x_n)$ [The first inequality constraint]

PHI(2) = $\phi_2(x_1, x_2,\ldots,x_n)$

PHI(NCONS) = $\phi_{NCONS}(x_1, x_2,\ldots,x_n)$

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

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

$\vdots$

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

GPHI(1,1) = $\frac{\partial \phi_1}{\partial x_1}$

GPHI(2,1) = $\frac{\partial \phi_1}{\partial x_2}$

$\vdots$

$$GPHI(N,1) = \frac{\partial \phi_1}{\partial x_n}$$

$\vdots$

$$GPHI(N,2) = \frac{\partial \phi_2}{\partial x_n}$$

$\vdots$

$$GPHI(N,NCONS) = \frac{\partial \phi_{NCONS}}{\partial x_n}$$

NCONS = Number of inequality constraints

P = The power to be used in the least pth approximation (set = 1.E5 if no information is available)

EPSPHI = A small number used to determine the margin by which constraints may be violated (Set=1.E-5 if no information is available)

```
CALL FMINMAX(U, GU, PHI, GPHI, N, NCONS, F, G, P, EPSPHI)
RETURN
END
```

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

4. DATA DECK Parameters to be supplied as data are defined below

MET1 First method to be called by GRADNLP

if MET1=1 New Fletcher method will be called

if MET1=2 Jacobson-Oksman method will be called

if MET1=3 Fletcher-Powell method will be called

if MET1=0 No first method will be called

MET2 Second method to be called by GRADNLP

if MET2=1 New Fletcher method will be called

if MET2=2 Jacobson-Oksman method will be called

	if MET2=3 Fletcher-Powell method will be called
	if MET2=0 No second method will be called
MET3	Third method to be called by GRADNLP
	if MET3=1 New Fletcher method will be called
	if MET3=2 Jacobson-Oksman method will be called
	if MET3=3 Fletcher-Powell method will be called
	if MET3=0 No third method will be called
M	A parameter to select the starting point; if M=1 same starting point is used by all the methods called; for any other value of M each method starts with the optimum left by the last method
MAX	Maximum number of permissible iterations
MODE	A parameter to choose the stopping criterion for Jacobson-Oksman method
	if MODE=1 criterion will be $\Delta F \leq$ a small number
	if MODE=2 criterion will be $\|\|$gradients$\|\| \leq$ a small number
IPRINT	Intermediate output is printed out every IPRINT iterations; it should be set = 0 if no intermediate output is desired.
IDATA	Input data is printed out if IDATA=1; it should be set=0 if input data is not to be printed out
EST	Minimum estimated value of the objective function
EPS1	Small test quantity used by the Fletcher-Powell method
ETA(I),I=1,4	Test quantities used by the Jacobson-Oksman method
EPS(I),I=1,N	Test quantities used by the New Fletcher method

XSTRT(I),I=1,NN Starting values for variables $x_1, x_2, \ldots, x_n$ and x_{n+1}, x_{n+2}. Two extra variables x_{n+1} and x_{n+2} are required by the Jacobson-Oksman method. Suggested starting values for x_{n+1} and x_{n+2} are the estimated order of the objective function and the minimum estimated value of the function respectively.

AO Initial value of the positive parameter α used in the formulation of the unconstrained objective function. It should be set=1.0, if no information is available

BO Initial value of the nonnegative parameter β used in the formulation of the unconstrained objective function. It should be selected in such a way that the actual objective function plus β is always positive.

Recommended values for some of the parameters are

MAX = 100

EPS1 = 1.E-6

ETA(1) = 1.E-4

ETA(2) = 1.E-8

ETA(3) = ETA(4) = 1.E-16

EPS(I),I=1,N Each = 1.E-6

EST A lower bound on the minimum value of the objective function may be obtained from physical reasons if the true minimum is not known, e.g., for approximation problems 0.0 is convenient.

Setting up the data deck

Card No.	Format	Parameters
1	8I5	MET1, MET2, MET3, M, MAX, MODE, IPRINT, IDATA
2	5E16.8	EPS1, (ETA(I), I=1,4)
As many as required	5E16.8	EST, (EPS(I), I=1,N)
As many as required	5E16.8	(XSTRT(I), I=1, NN)
Last	2E16.8	A0, B0

COMMENTS

As explained above any number one, two, or all the three methods and in any order may be called depending upon the values of MET1, MET2 and MET3 chosen by the user. Similarly by choosing the appropriate values of IPRINT and IDATA the user may or may not print out the input data and the intermediate output. Results for some of the problems solved using this package along with the results obtained using the Fiacco-McCormick approach have been included in Appendix A. Appendix B shows the general structure of the package. Though some of the subroutines in this package have exactly the same names as the ones in GRADMIN, they are not exactly the same, and for this reason the user must treat the two packages separately and should not try to mix subroutines from one package to another package.

REFERENCES

1. J.W. Bandler and C. Charalambous, "A new approach to nonlinear programming", 5th Hawaii Int. Conf. on Systems Science, (Honolulu, Jan. 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, August 1970.

4. D.H. Jacobson and W. Oksman, "An algorithm that minimizes homogeneous functions of n variables in n+2 iterations and rapidly minimizes general functions", Division of Engineering and Applied Physics, Harvard University, Cambridge, Mass., Technical Report No. 618, October 1970.

5. J. Kowalik, M.R. Osborne and D.M. Ryan, "A new method for constrained optimization problems", Operations Research, pp. 973-983, November-December 1969.

ACKNOWLEDGEMENTS

The authors acknowledge the assistance of C. Charalambous. Dr. R. Fletcher and Dr. D.H. Jacobson made available listings of their programs.

January 28, 1972

APPENDIX A

The following test problems have been solved using this package.

Problem 1 To minimize (Starting points $x_1=0$, $x_2=0$, $x_3=0$, $x_4=0$)

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

subject to

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

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

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

Problem 2 To minimize (Starting points $x_1=0.5$, $x_2=0.5$, $x_3=0.5$)

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

subject to $x_1 \geq 0$, $x_2 \geq 0$, $x_3 \geq 0$ and $x_1 + x_2 + 2x_3 \leq 3$

Problem 3 To minimize (Starting points $x_1=15$, $x_2=3$, $x_3=25$)

$$U = -x_1 x_2 x_3$$

subject to

$$0 \leq x_1 \leq 20$$

$$0 \leq x_2 \leq 11$$

$$0 \leq x_3 \leq 42$$

$$x_1 + 2x_2 + 2x_3 \leq 72$$

The results given by this package, together with the ones obtained using the Fiacco-McCormick technique [5] are as follows.

A suitable listing of the input deck, and the printouts of input data and final results for Problem 2, are given in Figs. 1-3.

Problem 1	New Approach	Fiacco-McCormick
Independent variables		
x_1	-0.00003	-.00001
x_2	0.99978	0.99998
x_3	1.99999	1.99996
x_4	-.99998	-.99996
Objective Function U	-43.9991	-43.9985
No. of Function Evaluations	520	145
Initial Values of Parameters	α=100, β=1000	r=1.0
Final Values of Parameters	α=100, β=1000	$r=10^{-7}$
Problem 2		
Independent Variables		
x_1	1.33333	1.33335
x_2	0.77777	0.77776
x_3	0.44444	0.44440
Objective Function U	0.11111	0.11114
No. of Function Evaluations	48	127
Initial Values of Parameters	α=1.0, β=0	r=1.0
Final Values of Parameters	α=1.0, β=0	$r=10^{-9}$
Problem 3		
Independent Variables		
x_1	20.00000	19.99996
x_2	11.00000	10.99996
x_3	15.00000	15.00004
Objective Function U	-3300.00	-3299.98
No. of Function Evaluations	127	188
Initial Values of Parameters	$\alpha=10^7$, $\beta=10^4$	r=1.0
Final Values of Parameters	$\alpha=10^7$, $\beta=10^4$	$r=10^{-7}$

```
HSDJ.                                                                    JHA V.
LOADER(PPLOADR)
ATTACH(TAPE,GRADNLP,ID=GABABNLER,MR=1)
RUN(S)
SETINDF.
LOAD(TAPE)
REDUCE.
LGO.
'        6400 END OF RECORD
      PROGRAM TST (INPUT,OUTPUT,TAPE5=INPUT,TAPE6=OUTPUT)
      DIMENSION DUM1(5),DUM2(5),X(5),XSTRT(5),X2(5),X1(5),G(5),G2(5),
     1ALFA(5),H(15),P(5,5),Y(5),PY(5),PE(5),BIGV(5),EPS(3)
      N=3
      NN=N+2
      KR=1
      CALL GRADNLP(N,X,XSTRT,X2,X1,G,G2,ALFA,H,P,PY,PE,BIGV,EPS,
     1NN,Y,DUM1,DUM2,KR)
      STOP
      END
      SUBROUTINE FUNCT(N,X,F,G)
      DIMENSION GPHI(3,4),X(1),G(1),PHI(4),GU(4)
      U=9.-8.*X(1)-6.*X(2)-4.*X(3)+2.*X(1)*X(1)+2.*X(2)**2+X(3)**2+2.*
     1X(1)*X(2)+2.*X(1)*X(3)
      PHI(1)=X(1)
      PHI(2)=X(2)
      PHI(3)=X(3)
      PHI(4)=3.-X(1)-X(2)-2.*X(3)
      GU(1)=-8.+4.*X(1)+2.*X(2)+2.*X(3)
      GU(2)=-6.+4.*X(2)+2.*X(1)
      GU(3)=-4.+2.*X(3)+2.*X(1)
      GPHI(1,1)=1.0
      GPHI(1,2)=0.
      GPHI(1,3)=0.
      GPHI(1,4)=-1.0
      GPHI(2,1)=0.
      GPHI(2,2)=1.
      GPHI(2,3)=0.
      GPHI(2,4)=-1.0
      GPHI(3,1)=0.
      GPHI(3,2)=0.
      GPHI(3,3)=1.0
      GPHI(3,4)=-2.0
      NCONS=4
      EPSPHI=1.E-5
      P=1.E5
      CALL FMINMAX(U,GU,PHI,GPHI,N,NCONS,F,G,P,EPSPHI)
      CALL FMINMAX(U,GU,PHI,GGPHI,N,NCONS,F ,G,P,EPSPHI)
      RETURN
      END
'        6400 END OF RECORD
   1     3     2     1  500      1      1      1
 1.E-              61.E-             41.E-             8 1.E-          161.E-        1
           0.                    1.E-6              1.E-6              1.E-6
0.5                0.5                0.5                2.0                0.
1.0                0.
'           END OF FILE


                              CD TOT    0056
```

Figure 1

```
NPUT DATA

OLLOWING METHODS HAVE BEEN CALLED

EW FLETCHER METHOD
LETCHER-POWELL METHOD
ACOBSON-OKSMAN METHOD

UMBER OF INDEPENDENT VARIABLES.....................................N=     3

AXIMUM NUMBER OF ALLOWABLE ITERATIONS.............................MAX=   500

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

TARTING POINT TO BE SAME FOR ALL THE METHODS IF M=1...............M=     1

TARTING VALUE FOR VECTOR X(I).............................XSTRT( 1)=   5.00000000E-(
                                                          XSTRT( 2)=   5.00000000E-(
                                                          XSTRT( 3)=   5.00000000E-(

EST QUANTITIES TO BE USED IN JACOBSON-OKSMAN METHOD.........ETA( 1)=   1.00000000E-(
                                                            ETA( 2)=   1.00000000E-(
                                                            ETA( 3)=   1.00000000E-1
                                                            ETA( 4)=   1.00000000E-1

EST QUANTITIES TO BE USED IN NEW FLETCHER METHOD............EPS( 1)=   1.00000000E-(
                                                            EPS( 2)=   1.00000000E-(
                                                            EPS( 3)=   1.00000000E-(

EST QUANTITY TO BE USED IN FLETCHER-POWELL METHOD..............EPS1=   1.00000000E-(

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

NITIAL VALUE OF THE PARAMETER ALPHA..............................AO=   1.00000000E+(

NITIAL VALUE OF THE PARAMETER BETA...............................BO=   0.
```

Figure 2

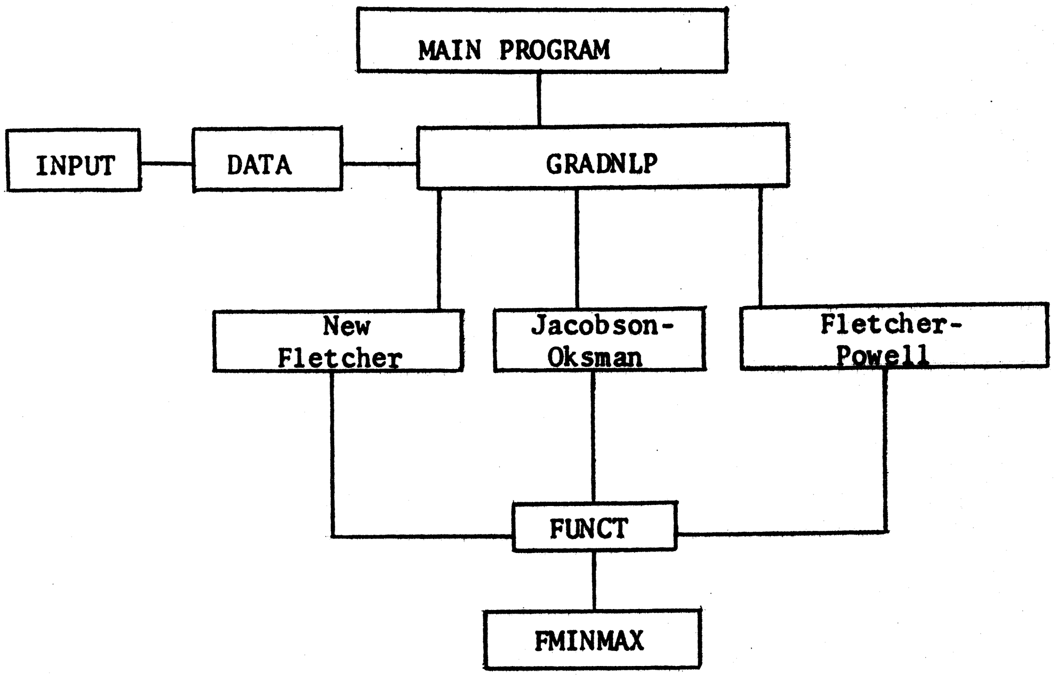
```
                              FOLLOWING IS THE OPTIMUM SOLUTION
                              ---------------------------------

ARTIFICIAL UNCONSTRAINED FUNCTION F =  1.11111704E-01
      ACTUAL OBJECTIVE FUNCTION U =  1.11111472E-01
                            X( 1)=  1.33333387E+00
                            X( 2)=  7.77777490E-01
                            X( 3)=  4.44443506E-01

     INEQUALITY CONSTRAINTS
                          PHI( 1)=  1.33333387E+00
                          PHI( 2)=  7.77777490E-01
                          PHI( 3)=  4.44443506E-01
                          PHI( 4)=  1.62370053E-06
    NUMBER OF FUNCTION EVALUATIONS =   48
 FINAL VALUE OF THE PARAMETER ALPHA =  1.00000000E+00
 FINAL VALUE OF THE PARAMETER BETA  =  0.

          EXECUTION TIME IN SECONDS=     .65600
```

Figure 3

New Fletcher Method

APPENDIX B

The general structure of the package is



Attached is the complete FORTRAN listing of the package.

```
      SUBROUTINE FMINMAX (U,GU,PHI,GPHI,N,NCONS,F,G,P,EPSPHI)
      DIMENSION GU(1), PHI(1), GPHI(N,1), G(1), A(50), QUA(50)
      COMMON /SPL/ ALPHA,BETA,IFLAGA,IFLAGB,ICHEK,AB,KKK
      COMMON /BLK2/ UR,NC,PPHI(100)
      UR=U
      NC=NCONS
      IFLAGA=0
      SUM=0.0
      IFLAGB=1.0
      UU=U+BETA
      IF (UU.GE.0.) IFLAGB=0
      IF (NCONS.EQ.0.OR.ALPHA.EQ.0.) GO TO 8
      DO 1 I=1,NCONS
      A(I)=UU-ALPHA*PHI(I)
1     CONTINUE
      NT=NCONS+1
      A(NT)=UU
      DO 2 I=1,NT
      IF (A(I).GE.0.) IFLAGB=0
2     CONTINUE
      IF (IFLAGB.EQ.1) GO TO 7
      UM=0.
      DO 3 I=1,NT
      UM=AMAX1(UM,A(I))
3     CONTINUE
      DO 4 I=1,NCONS
      QUA(I)=AMAX1(0.,A(I))
      QUAD=QUA(I)/UM
      QUADP=QUAD**P
      SUM=SUM+QUADP
4     CONTINUE
      UU=AMAX1(0.,UU)
      SUMT=SUM+((UU/UM)**P)
      F=UM*((SUMT**(1./P)))
      DO 6 I=1,N
      SUM2=0.0
      DO 5 J=1,NCONS
      TEMP=QUA(J)/UM
      SUM2=SUM2+(TEMP**(P-1.))*(GU(I)-ALPHA*GPHI(I,J))
5     CONTINUE
      G(I)=((SUMT)**(1./P-1.))*(((UU/UM)**(P-1.))*GU(I))+SUM2
6     CONTINUE
      GO TO 11
7     CONTINUE
      BETA=BETA+AB
      GO TO 11
8     IF (IFLAGB.EQ.1) GO TO 10
      F=UU
      DO 9 I=1,N
      G(I)=GU(I)
9     CONTINUE
      GO TO 11
10    BETA=BETA+AB
11    CONTINUE
      IF (ICHEK.EQ.0) GO TO 13
      DO 12 I=1,NCONS
      PPHI(I)=PHI(I)
      PHIT=PHI(I)+EPSPHI
      IF (PHIT.LT.0.) IFLAGA=1
12    CONTINUE
13    CONTINUE
      RETURN
      END
```

```
      SUBROUTINE GRADNLP (N,X,XSTRT,X2,X1,G,G2,ALFA,H,P,PY,PE,BIGV,EPS,N
     1N,Y,DUM1,DUM2,KR)
      COMMON /BLK/ KO
      COMMON /SPL/ ALPHA,BETA,IFLAGA,IFLAGB,ICHEK,AB,KKK
C
      EXTERNAL FUNCT
      LOGICAL CONV,UNITH
      DIMENSION DUM1(1), DUM2(1)
      DIMENSION X(1), XSTRT(1), X2(1), X1(1), G(1), G2(1), ALFA(1), H(1)
     1, P(NN,1), Y(1), PY(1), PE(1), BIGV(1), EPS(1), ETA(4)
      UNITH=.TRUE.
      AB=BO
      ICHEK=0
      IF (KR.EQ.0) GO TO 1
      CALL DATA (N,NN,XSTRT,EST,EPS1,ETA,MET1,MET2,MET3,M,MAX,MODE,IPRIN
     1T,EPS,AO,BO)
1     DO 2 I=1,NN
      X(I)=XSTRT(I)
2     CONTINUE
      IF (MET1.EQ.0) MET1=4
      IF (MET2.EQ.0) MET2=4
      IF (MET3.EQ.0) MET3=4
      INDEX=0
      GO TO (5,12,20,27), MET1
3     GO TO (5,12,20,27), MET2
4     GO TO (5,12,20,27), MET3
5     IF (IPRINT.EQ.0) GO TO 6
      CALL WRITE1 (1)
6     CONTINUE
      KKK=0
      CALL SECOND (T1)
      IF (KR.NE.0) GO TO 8
      DO 7 I=1,NN
      X(I)=DUM1(I)
7     CONTINUE
8     CONTINUE
      ICHEK=0
      ALPHA=AO
      BETA=BO
9     CONTINUE
      KKK=KKK+1
      CALL VMM01 (N,X,F,G,H,UNITH,EST,EPS,MAXFN,IPRINT,IEXIT,PE)
      ICHEK=1
      IF (IFLAGB.EQ.1) GO TO 9
      CALL FUNCT (N,X,F,G)
      IF (IFLAGA.EQ.0) GO TO 10
      ALPHA=ALPHA*10.
      GO TO 9
10    CONTINUE
      DO 11 I=1,NN
      DUM1(I)=X(I)
11    CONTINUE
      CALL SECOND (T2)
      CALL FINAL (X,F,N)
      T=T2-T1
      WRITE (6,32) T
      GO TO 27
12    IF (IPRINT.EQ.0) GO TO 13
      CALL WRITE1 (2)
13    CONTINUE
      KKK=0
      CALL SECOND (T1)
      IF (KR.NE.0) GO TO 15
      DO 14 I=1,NN
      X(I)=X2(I)
14    CONTINUE
15    CONTINUE
      ICHEK=0
      ALPHA=AO
      BETA=BO
16    CONTINUE
      KKK=KKK+1
      CALL THETA (X,N,ETA,EST,MAX,MODE,X2,X1,G,G2,ALFA,H,P,Y,PY,PE,BIGV,
     1EPS1,NN,IPRINT,F2)
      ICHEK=1
      IF (IFLAGB.EQ.1) GO TO 17
```

```
      CALL FUNCT (N,X,F,G)
      IF (IFLAGA.EQ.0) GO TO 19
      ALPHA=ALPHA*10.
17    DO 18 I=1,NN
      X(I)=X2(I)
18    CONTINUE
      GO TO 16
19    CONTINUE
      CALL SECOND (T2)
      CALL FINAL (X2,F2,N)
      T=T2-T1
      WRITE (6,32) T
      GO TO 27
20    IF (IPRINT.EQ.0) GO TO 21
      CALL WRITE1 (3)
21    CONTINUE
      KKK=0
      CALL SECOND (T1)
      IF (KR.NE.0) GO TO 23
      DO 22 I=1,NN
      X(I)=DUM2(I)
22    CONTINUE
23    CONTINUE
      ICHEK=0
      ALPHA=AO
      BETA=BO
24    CONTINUE
      KKK=KKK+1
      CALL FMFP (FUNCT,N,X,F,G,EST,EPS1,MAX,IER,H,IPRINT)
      ICHEK=1
      IF (IFLAGB.EQ.1) GO TO 24
      CALL FUNCT (N,X,F,G)
      IF (IFLAGA.EQ.0) GO TO 25
      ALPHA=ALPHA*10.
      GO TO 24
25    CONTINUE
      DO 26 I=1,NN
      DUM2(I)=X(I)
26    CONTINUE
      CALL SECOND (T2)
      CALL FINAL (X,F,N)
      T=T2-T1
      WRITE (6,32) T
27    INDEX=INDEX+1
      IF (M.EQ.1) GO TO 28
      GO TO 30
28    DO 29 I=1,NN
      X(I)=XSTRT(I)
29    CONTINUE
30    CONTINUE
      GO TO (3,4,31), INDEX
31    CONTINUE
      RETURN
C
C
C
32    FORMAT (1HO,//25X,*EXECUTION TIME IN SECONDS=*,F10.5)
      END
```

```
      SUBROUTINE DATA (N,NN,XSTRT,EST,EPS1,ETA,MET1,MET2,MET3,M,MAX,MODE
     1,IPRINT,EPS,AO,BO)
      DIMENSION XSTRT(1), EPS(1), ETA(1)
      READ (5,1) MET1,MET2,MET3,M,MAX,MODE,IPRINT,IDATA
      READ (5,2) EPS1,(ETA(I),I=1,4)
      READ (5,2) EST,(EPS(I),I=1,N)
      READ (5,2) (XSTRT(I),I=1,NN)
      READ (5,2) AO,BO
      IF (IDATA.EQ.0) RETURN
      CALL INPUT (MET1,MET2,MET3,M,MAX,MODE,N,IPRINT,IDATA,EPS1,ETA,EST,
     1EPS,XSTRT,AO,BO)
      RETURN
C
C
C
    1 FORMAT (8I5)
    2 FORMAT (5E16.8)
      END
```

```
      SUBROUTINE FINAL (X,F,N)
      DIMENSION X(1)
      COMMON /SPL/ ALPHA,BETA,IFLAGA,IFLAGB,ICHEK,AB,KKK
      COMMON /BLK/ KO
      COMMON /BLK1/ NUMF
      COMMON /BLK2/ UR,NC,PPHI(100)
      WRITE (6,9)
      IF (KO.EQ.0) GO TO 1
      WRITE (6,10)
      GO TO 2
    1 WRITE (6,11)
    2 CONTINUE
      WRITE (6,12) F
      WRITE (6,3) UR
      WRITE (6,13) (I,X(I),I=1,N)
      WRITE (6,4)
      WRITE (6,5) (I,PPHI(I),I=1,NC)
      WRITE (6,6) NUMF
      WRITE (6,7) ALPHA
      WRITE (6,8) BETA
      RETURN
C
C
C
    3 FORMAT (1H0,21X,*ACTUAL OBJECTIVE FUNCTION U =*,E16.8,/)
    4 FORMAT (1H0,//,21X,*INEQUALITY CONSTRAINTS*,/)
    5 FORMAT (43X,*PHI(*,I2,*)=*,E16.8)
    6 FORMAT (1H0,/,19X,*NUMBER OF FUNCTION EVALUATIONS =*,I5)
    7 FORMAT (1H0,/,15X,*FINAL VALUE OF THE PARAMETER ALPHA =*,E16.8)
    8 FORMAT (1H0,/,15X,*FINAL VALUE OF THE PARAMETER BETA  =*,E16.8)
    9 FORMAT (1H1)
   10 FORMAT (1H0,40X,*FOLLOWING IS THE OPTIMUM SOLUTION*,/,41X,*-------
     1---------------------------*)
   11 FORMAT (1H0,44X,*RESULTS AT LAST ITERATION*/,45X,*----------------
     1---------*)
   12 FORMAT (1H0,//,14X,*ARTIFICIAL UNCONSTRAINED FUNCTION F =*,E16.8,/
     1)
   13 FORMAT (1H0,44X,*X(*,I2,*)=*,E16.8)
      END
```

```
      SUBROUTINE VMM01 (N,X,F,G,H,UNITH,FEST,EPS,MAXFN,IPRINT,IEXIT,PE)
      DIMENSION PE(1)
      DIMENSION X(1), G(1), H(1), EPS(1)
      COMMON /BLK1/ NFNS
      COMMON /SPL/ ALPHA,BETA,IFLAGA,IFLAGB,ICHEK,AB,KKK
      LOGICAL CONV,UNITH
      COMMON /BLK/ KO
      KO=0
      CALL FUNCT (N,X,F,G)
      IF (IFLAGB.EQ.1) RETURN
      IF (F.LT.FEST) GO TO 26
      IF (KKK.NE.1) GO TO 1
      NFNS=1
      ITN=0
      CALL SECOND (T3)
    1 CONTINUE
      STEP=1.
      IDX=N
      IDG=N+N
      IH=IDG+N
      IF (.NOT.UNITH) GO TO 3
      IJ=IH+1
      DO 2 I=1,N
      DO 2 J=I,N
      H(IJ)=0.
      IF (I.EQ.J) H(IJ)=1.0
    2 IJ=IJ+1
    3 CONV=.TRUE.
      GDX=0.
      DO 7 I=1,N
      Z=0.
      IJ=IH+I
      IF (I.EQ.1) GO TO 5
      II=I-1
      DO 4 J=1,II
      Z=Z-H(IJ)*G(J)
      IJ=IJ+N-J
    4 CONTINUE
    5 DO 6 J=I,N
      Z=Z-H(IJ)*G(J)
      IJ=IJ+1
    6 CONTINUE
      IF (ABS(Z).GT.EPS(I)) CONV=.FALSE.
      H(IDX+I)=Z
      GDX=GDX+G(I)*Z
    7 CONTINUE
      DO 8 I=1,N
      PE(I)=X(I)
    8 CONTINUE
C
      IF (IPRINT.EQ.0) GO TO 10
      IF (MOD(ITN,IPRINT).NE.0) GO TO 10
      CALL SECOND (T4)
      TIME=T4-T3
      IF (KKK.NE.1) GO TO 9
      CALL WRITE2 (X,N,G,F,NFNS,ITN,TIME)
    9 KKK=1
   10 IEXIT=1
      IF (CONV) GO TO 27
      IEXIT=2
      IF (GDX.GE.0.) GO TO 27
      Z=1.
      IF (ITN.LT.N.AND.UNITH) Z=STEP
      W=2.*(FEST-F)/GDX
      IF (W.LT.Z) Z=W
      STEP=Z
   11 GDX=GDX*Z
      DO 12 I=1,N
      H(IDX+I)=H(IDX+I)*Z
      X(I)=X(I)+H(IDX+I)
   12 CONTINUE
      CALL FUNCT (N,X,FP,H)
      IF (IFLAGB.EQ.1) RETURN
      IF (FP.LT.FEST) GO TO 26
      NFNS=NFNS+1
      IEXIT=3
```

```
      IF (ITN.EQ.MAXFN) GO TO 27
      GPDX=0.
      DO 13 I=1,N
      H(IDG+I)=H(I)-G(I)
      GPDX=GPDX+H(I)*H(IDX+I)
13    CONTINUE
      DGDX=GPDX-GDX
      IF (F.GT.FP-.0001*GDX) GO TO 15
      IEXIT=4
      IF (GPDX.LT.0..AND.ITN.GT.N) GO TO 27
      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 14 I=1,N
      X(I)=X(I)-H(IDX+I)
14    CONTINUE
      GO TO 17
15    F=FP
      DO 16 I=1,N
      G(I)=H(I)
16    CONTINUE
      IF (DGDX.GT.0.) GO TO 18
      GDX=GPDX
      Z=4.
17    STEP=Z*STEP
      GO TO 11
18    IF (GPDX.LT.0.5*GDX) STEP=2.*STEP
      DGHDG=0.
      DO 22 I=1,N
      Z=0.
      IJ=IH+I
      IF (I.EQ.1) GO TO 20
      II=I-1
      DO 19 J=1,II
      Z=Z+H(IJ)*H(IDG+J)
      IJ=IJ+N-J
19    CONTINUE
20    DO 21 J=I,N
      Z=Z+H(IJ)*H(IDG+J)
      IJ=IJ+1
21    CONTINUE
      DGHDG=DGHDG+Z*H(IDG+I)
      H(I)=Z
22    CONTINUE
      IF (DGHDG.LT.0.0) DGHDG=DGDX*0.01
      IF (DGDX.LT.DGHDG) GO TO 24
      W=1.0+DGHDG/DGDX
      DO 23 I=1,N
      H(IDX+I)=W*H(IDX+I)-H(I)
23    CONTINUE
      DGDX=DGDX+DGHDG
      DGHDG=DGDX
24    IJ=IH
      DO 25 I=1,N
      W=H(IDX+I)/DGDX
      Z=H(I)/DGHDG
      DO 25 J=I,N
      IJ=IJ+1
25    H(IJ)=H(IJ)+W*H(IDX+J)-Z*H(J)
      ITN=ITN+1
      GO TO 3
26    IEXIT=5
27    IF (IEXIT.EQ.1) KO=1
      IF (IEXIT.NE.4) GO TO 29
      DO 28 I=1,N
      X(I)=PE(I)
28    CONTINUE
29    CONTINUE
      IF (IPRINT.EQ.0) RETURN
      GO TO (30,31,32,31,33), IEXIT
30    WRITE (6,35) IEXIT
      GO TO 34
31    WRITE (6,36) IEXIT
```

```
      GO TO 34
   32 WRITE (6,37) IEXIT
      GO TO 34
   33 WRITE (6,38) IEXIT
   34 CONTINUE
      RETURN
C
C
C
   35 FORMAT (/,1HO,*IEXIT=*,I2,*CRITERION FOR OPTIMUM (CHANGE IN VECTOR
     1 X .LT.EPS) HAS BEEN SATISFIED*)
   36 FORMAT (/,1HO,*IEXIT=*,I2,*EITHER OF THE FOLLOWING THINGS HAS HAPP
     1ENED*,/,9X,*1. EPS CHOSEN IS TOO SMALL*,/,9X,*2. GRADIENTS ARE NOT
     2 CORRECT*,/,9X,*3. MATRIX H GOES SINGULAR*)
   37 FORMAT (/,1HO,*IEXIT=*,I2,*MAXIMUM NUMBER OF ALLOWABLE ITERATION H
     1AS BEEN EXCEEDED*)
   38 FORMAT (/,1HO,*IEXIT=*,I2,*FUNCTION VALUE LESS THAN MINIMUM ESTIMA
     1TED HAS BEEN DETECTED*)
      END
```

```
      SUBROUTINE THETA (X,NDIM,ETA,EST,MAX,MODE,X2,X1,G1,G2,ALFA,H,P,Y,P
     1Y,PE,BIGV,EPS1,NN,IPRINT,F2)
      DIMENSION X(1), X1(1), ETA(1), X2(1), G1(1), G2(1), ALFA(1), H(1),
     1 P(NN,1), Y(1), PY(1), PE(1), BIGV(1)
      COMMON /BLK1/ NUMF
      COMMON /SPL/ ALPHA,BETA,IFLAGA,IFLAGB,ICHEK,AB,KKK
      COMMON /BLK/ KO
      KO=0
      IFLAG=0
      KK=0
      N2=NDIM+1
      N1=NDIM+2
      IF (KKK.NE.1) GO TO 1
      CALL SECOND (T3)
      M=0
      NUMF=0
    1 CONTINUE
      IER=0
      DO 2 I=1,N1
      X1(I)=X(I)
    2 CONTINUE
      CALL FUNCT (NDIM,X1,F1,G1)
      NUMF=NUMF+1
      IF (IFLAGB.EQ.1) RETURN
      CALL SECOND (T4)
      TIME=T4-T3
      IF (IPRINT.EQ.0) GO TO 4
      IF (KKK.NE.1) GO TO 3
      CALL WRITE2 (X1,NDIM,G1,F1,NUMF,M,TIME)
    3 KKK=1
    4 CONTINUE
      DO 5 I=1,NDIM
      X2(I)=X1(I)
      G2(I)=G1(I)
      H(I)=-G1(I)
    5 CONTINUE
      F2=F1
      X2(N2)=X1(N2)
      X2(N1)=X1(N1)
    6 CONTINUE
      KOUNT=0
      EPS=EPS1
      CALL MIN1D (FUNCT,X2,H,RO,NDIM,F2,G2,NUMF,IER,EPS,EST)
      IF (IER.NE.0) GO TO 32
      DO 7 I=1,N1
      BIGV(I)=X2(I)
      ALFA(I)=X2(I)
    7 CONTINUE
      RO=-RO
      GG=0.
      DO 8 I=1,NDIM
      GG=GG+G2(I)*G2(I)
    8 CONTINUE
      GG=SQRT(GG)
      IF (IPRINT.EQ.0) GO TO 9
      IF (KK.NE.IPRINT) GO TO 9
      CALL SECOND (T4)
      TIME=T4-T3
      KK=0
      CALL WRITE2 (X2,NDIM,G2,F2,NUMF,M,TIME)
    9 CONTINUE
      DO 11 I=1,N1
      DO 10 J=1,N1
      P(I,J)=0.
   10 CONTINUE
      P(I,I)=1.
   11 CONTINUE
   12 CONTINUE
      KOUNT=0
      KOUNT=KOUNT+1
   13 DO 14 I=1,NDIM
      Y(I)=G2(I)
   14 CONTINUE
      Y(N2)=F2
      Y(N1)=ETA(1)
      V=0.
```

```
      DO 15 I=1,NDIM
      V=V+X2(I)*G2(I)
   15 CONTINUE
      YA=0.
      DO 16 I=1,N1
      YA=YA+Y(I)*ALFA(I)
   16 CONTINUE
      VYA=V-YA
      BIGV(KOUNT)=V
      DO 17 I=1,N1
      PY(I)=0.
      PE(I)=P(I,KOUNT)
      DO 17 J=1,N1
   17 PY(I)=PY(I)+P(J,I)*Y(J)
      EPY=PY(KOUNT)
      IF (ABS(EPY).LT.ETA(3)) GO TO 33
      PY(KOUNT)=PY(KOUNT)-1.
      DO 18 I=1,N1
      DO 18 J=1,N1
   18 P(I,J)=P(I,J)-PE(I)*PY(J)/EPY
      DO 19 I=1,N1
      ALFA(I)=0.
      DO 19 J=1,N1
   19 ALFA(I)=ALFA(I)+P(I,J)*BIGV(J)
      DEL=0.
      DO 20 I=1,NDIM
      DEL=DEL+G2(I)*(X2(I)-ALFA(I))
   20 CONTINUE
      IF (ABS(DEL).GT.ETA(4)) GO TO 21
      IF (IFLAG.EQ.1) GO TO 31
      IFLAG=1
      GO TO 33
   21 IFLAG=0
      DO 22 I=1,N1
      H(I)=X2(I)-ALFA(I)
      IF (DEL.GT.0) H(I)=-H(I)
   22 CONTINUE
      DO 23 I=1,NDIM
      X1(I)=X2(I)
      G1(I)=G2(I)
   23 CONTINUE
      F1=F2
      X1(N2)=X2(N2)
      X1(N1)=X2(N1)
      X2(N2)=ALFA(N2)
      X2(N1)=ALFA(N1)
      CALL MIN1D (FUNCT,X2,H,RO,NDIM,F2,G2,NUMF,IER,EPS,EST)
      IF (IER.NE.0) GO TO 32
      IF (DEL.GT.0) RO=-RO
      GG=0.
      DO 24 I=1,NDIM
      GG=GG+G2(I)*G2(I)
   24 CONTINUE
      GG=SQRT(GG)
      KOUNT=KOUNT+1
      M=M+1
      KK=KK+1
      IF (KK.NE.IPRINT) GO TO 25
      KK=0
      CALL SECOND (T4)
      TIME=T4-T3
      CALL WRITE2 (X2,NDIM,G2,F2,NUMF,M,TIME)
   25 CONTINUE
      IF (M.GT.MAX) GO TO 28
      IF (MODE.EQ.2) GO TO 26
      IF (((F1-F2).LE.ETA(2))) GO TO 29
      GO TO 27
   26 IF ((GG.LT.ETA(1))) GO TO 30
   27 CONTINUE
      IF (KOUNT.LE.N1) GO TO 13
      GO TO 12
   28 WRITE (6,37)
      GO TO 32
   29 KO=1
```

```
      IF (IPRINT.EQ.0) RETURN
      WRITE (6,38)
      GO TO 32
30    KO=1
      IF (IPRINT.EQ.0) RETURN
      WRITE (6,39)
      GO TO 32
31    WRITE (6,36)
      KO=1
32    RETURN
33    IF (IPRINT.EQ.0) GO TO 34
      PRINT 40
34    CONTINUE
      DO 35 I=1,NDIM
      X1(I)=X2(I)
      G1(I)=G2(I)
      H(I)=-G1(I)
35    CONTINUE
      F1=F2
      X1(N2)=X(N2)
      X1(N1)=X(N1)
      X2(N2)=X(N2)
      X2(N1)=X(N1)
      GO TO 6
C
C
C
36    FORMAT (1HO,*RESTART COULD NOT YIELD SIGNIFICANT IMPROVEMENT,OPTIM
     1UM HAS BEEN REACHED*)
37    FORMAT (1HO,*MAXIMUM NUMBER OF ALLOWABLE ITERATIONS HAS BEEN EXCEE
     1DED*)
38    FORMAT (1HO,*CRITERION FOR OPTIMUM (FUNCTION VALUE DOES NOT CHANGE
     1  SIGNIFICANTLY ,MODE=1) HAS BEEN SATISFIED*)
39    FORMAT (1HO,*CRITERION FOR OPTIMUM (GRADIENTS HAVE BECOME TOO SMAL
     1L MODE=2) HAS BEEN SATISFIED*)
40    FORMAT (///20X,*A RESTART HAS OCCURRED*///)
      END
```

•
•
•

```
      SUBROUTINE INPUT (MET1,MET2,MET3,M,MAX,MODE,N,IPRINT,IDATA,EPS1,ET
     1A,EST,EPS,XSTRT,AO,BO)
      DIMENSION XSTRT(1), ETA(1), EPS(1)
      WRITE (6,7)
      IF (MET1.EQ.0) MET1=4
      IF (MET2.EQ.0) MET2=4
      IF (MET3.EQ.0) MET3=4
      INDEX=0
      GO TO (3,4,5,6), MET1
    1 GO TO (3,4,5,6), MET2
    2 GO TO (3,4,5,6), MET3
    3 WRITE (6,8)
      GO TO 6
    4 WRITE (6,9)
      GO TO 6
    5 WRITE (6,10)
    6 CONTINUE
      INDEX=INDEX+1
      IF (INDEX.EQ.1) GO TO 1
      IF (INDEX.EQ.2) GO TO 2
      WRITE (6,11) N
      WRITE (6,12) MAX
      WRITE (6,13) IPRINT
      WRITE (6,14) M
      WRITE (6,15) XSTRT(1)
      WRITE (6,16) (I,XSTRT(I),I=2,N)
      WRITE (6,17) ETA(1),ETA(2),ETA(3),ETA(4)
      WRITE (6,18) EPS(1)
      WRITE (6,19) (I,EPS(I),I=2,N)
      WRITE (6,20) EPS1
      WRITE (6,21) EST
      WRITE (6,22) AO
      WRITE (6,23) BO
      RETURN
C
C
    7 FORMAT (1HO,*INPUT DATA*,/,1X,10(*-*),//,1X,*FOLLOWING METHODS HAV
     1E BEEN CALLED*,/)
    8 FORMAT (1HO,*NEW FLETCHER METHOD*)
    9 FORMAT (1HO,*JACOBSON-OKSMAN METHOD*)
   10 FORMAT (1HO,*FLETCHER-POWELL METHOD*)
   11 FORMAT (1HO,/1X,*NUMBER OF INDEPENDENT VARIABLES*,36(*.*),*N=*,I5,
     1/)
   12 FORMAT (1HO,*MAXIMUM NUMBER OF ALLOWABLE ITERATIONS*,27(*.*),*MAX=
     1*,I5,/)
   13 FORMAT (1HO,*INTERMEDIATE OUTPUT TO BE PRINTED EVERY IPRINT ITERAT
     1IONS*,5(*.*),*IPRINT=*,I5,/)
   14 FORMAT (1HO,*STARTING POINT TO BE SAME FOR ALL THE METHODS IF M=1*
     1,15(*.*),*M=*,I5,/)
   15 FORMAT (1HO,*STARTING VALUE FOR VECTOR X(I)*,29(*.*),*XSTRT( 1)=*,
     1E16.8)
   16 FORMAT (1HO,59X,*XSTRT(*,I2,*)=*,E16.8)
   17 FORMAT (1HO,/,1HO,*TEST QUANTITIES TO BE USED IN JACOBSON-OKSMAN M
     1ETHOD*,9(*.*),*ETA( 1)=*,E16.8,/,62X,*ETA( 2)=*,E16.8,/,62X,*ETA(
     23)=*,E16.8,/62X,*ETA( 4)=*,E16.8)
   18 FORMAT (1HO,/,1HO,*TEST QUANTITIES TO BE USED IN NEW FLETCHER METH
     1OD*,12(*.*),*EPS( 1)=*,E16.8)
   19 FORMAT (1HO,61X,*EPS(*,I2,*)=*,E16.8)
   20 FORMAT (1HO,/,1HO,*TEST QUANTITY TO BE USED IN FLETCHER-POWELL MET
     1HOD*,14(*.*),*EPS1=*,E16.8)
   21 FORMAT (1HO,/,1HO,*ESTIMATE OF LOWER BOUND ON FUNCTION TO BE MINIM
     1IZED*,14(*.*),*EST=*,E16.8)
   22 FORMAT (1HO,/,1HO,*INITIAL VALUE OF THE PARAMETER ALPHA*,30(*.*),*
     1AO=*,E16.8)
   23 FORMAT (1HO,/,1HO,*INITIAL VALUE OF THE PARAMETER BETA*,31(*.*),*B
     1O=*,E16.8)
      END
```

```
      SUBROUTINE MIN1D (FUNCT,X,H,AMBDA,N,F,G,NUMF,IER,EPS,EST)
      DIMENSION H(1), X(1), G(1)
      COMMON /SPL/ ALPHA,BETA,IFLAGA,IFLAGB,ICHEK,AB,KKK
      IER=0
      DY=0.
      HNRM=0.
      GNRM=0.
      DO 1 J=1,N
      HNRM=HNRM+ABS(H(J))
      GNRM=GNRM+ABS(G(J))
      DY=DY+H(J)*G(J)
    1 CONTINUE
      IF (DY) 2,31,31
    2 IF (HNRM/GNRM-EPS) 31,31,3
    3 FY=F
      ALFA=2.*(EST-F)/DY
      IF (X(N+1).GT.0.) ALFA=X(N+1)*ALFA/2.
      AMBDA=1.
      IF (ALFA) 6,6,4
    4 IF (ALFA-AMBDA) 5,6,6
    5 AMBDA=ALFA
    6 ALFA=0.
    7 FX=FY
      DX=DY
      DO 8 I=1,N
      X(I)=X(I)+AMBDA*H(I)
    8 CONTINUE
      CALL FUNCT (N,X,F,G)
      NUMF=NUMF+1
      IF (IFLAGB.EQ.1) RETURN
      IF (F.LT.FX) RETURN
      FY=F
      DY=0.
      DO 9 I=1,N
      DY=DY+G(I)*H(I)
    9 CONTINUE
      IF (DY) 10,30,13
   10 IF (FY-FX) 11,13,13
   11 AMBDA=AMBDA+ALFA
      ALFA=AMBDA
      IF (HNRM*AMBDA-1.E10) 7,7,12
   12 IER=2
      GO TO 31
   13 T=0.
   14 IF (AMBDA) 15,30,15
   15 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) 31,16,16
   16 W=ALFA*SQRT(DALFA)
      ALFA=DY-DX+W+W
      IF (ALFA) 17,18,17
   17 ALFA=(DY-Z+W)/ALFA
      GO TO 19
   18 ALFA=(Z+DY-W)/(Z+DX+Z+DY)
   19 ALFA=ALFA*AMBDA
      DO 20 I=1,N
      X(I)=X(I)+(T-ALFA)*H(I)
   20 CONTINUE
      CALL FUNCT (N,X,F,G)
      NUMF=NUMF+1
      IF (IFLAGB.EQ.1) RETURN
      IF (F.LT.FX) GO TO 30
      IF (F-FX) 21,21,22
   21 IF (F-FY) 30,30,22
   22 DALFA=0.
      DO 23 I=1,N
      DALFA=DALFA+G(I)*H(I)
   23 CONTINUE
      IF (DALFA) 24,27,27
   24 IF (F-FX) 26,25,27
   25 IF (DX-DALFA) 26,30,26
   26 FX=F
      DX=DALFA
      T=ALFA
```

```
      AMBDA=ALFA
      GO TO 14
27    IF (FY-F) 29,28,29
28    IF (DY-DALFA) 29,30,29
29    FY=F
      DY=DALFA
      AMBDA=AMBDA-ALFA
      GO TO 13
30    AMBDA=AMBDA-ALFA
      RETURN
31    CONTINUE
      IF (DY.GE.0.) IER=-2
      IF (HNRM/GNRM.LE.EPS) IER=-3
      IF (DALFA.LT.0.) IER=-1
      II=IABS(IER)
      GO TO (32,33,34), II
32    WRITE (6,36) IER
      GO TO 35
33    WRITE (6,37) IER
      GO TO 35
34    WRITE (6,38) IER
35    RETURN
C
C
C
36    FORMAT (1HO,*IER=*,I2,*THERE IS AN ERROR IN GRADIENTS CALCULATION*
     1)
37    FORMAT (1HO,*IER=*,I2,*ERROR HAS OCCURED, SEARCH DIRECTION IS NOT
     1A DESCENT DIRECTION*)
38    FORMAT (1HO,*IER=*,I2,*ERROR HAS OCCURED, SEARCH DIRECTION VECTOR
     1 IS TOO SMALL IN COMPARISON TO GRADIENT VECTOR*)
      END
```

```
      SUBROUTINE WRITE1 (N)
      WRITE (6,5)
      GO TO (1,2,3), N
1     WRITE (6,6)
      GO TO 4
2     WRITE (6,7)
      GO TO 4
3     WRITE (6,8)
4     CONTINUE
      WRITE (6,9)
      RETURN
C
C
C
5     FORMAT (1H1)
6     FORMAT (1HO,*OPTIMIZATION BY NEW FLETCHER METHOD*,/,1HO,*---------
     1---------------------------*)
7     FORMAT (1HO,*OPTIMIZATION BY JACOBSON-OKSMAN METHOD*,/,1HO,*------
     1------------------------------------*)
8     FORMAT (1HO,*OPTIMIZATION BY FLETCHER-POWELL METHOD*,/,1HO,*------
     1------------------------------------*)
9     FORMAT (1HO,*ITERATION*,2X,*FUNCTION*,6X,*TIME ELAPSED*,8X,*OBJECT
     1IVE*,9X,*ALPHA AND BETA*,5X,*VARIABLE VECTOR X(I)*,5X,*GRADIENT VE
     2CTOR G(I)*,/1HO,*NUMBER*,5X,*EVALUATIONS*,3X,*(SECONDS)*,11X,*FUNC
     3TION*,/)
      END
```

```
      SUBROUTINE WRITE2 (X,N,G,F,NUMF,ITER,TIME)
      DIMENSION X(1), G(1)
      COMMON /SPL/ ALPHA,BETA,IFLAGA,IFLAGB,ICHEK,AB,KKK
      WRITE (6,1) ITER,NUMF,TIME,F,ALPHA,X(1),G(1),BETA,((X(I),G(I)),I=2
     1,N)
      RETURN
C
C
C
    1 FORMAT (1H0,I5,7X,I5,5X,E16.8,3X,E16.8,4X,E16.8,6X,E16.8,9X,E16.8,
     1/,62X,E16.8,6X,E16.8,9X,E16.8,/,98(84X,E16.8,9X,E16.8,/)
      END
```

```
C        COMPUTE FUNCTION VALUE AND GRADIENT VECTOR FOR INITIAL ARGUMENT
      KO=0
      CALL SECOND (T3)
      CALL FUNCT (N,X,F,G)
      IF (IFLAGB.EQ.1) RETURN
      IF (KKK.NE.1) GO TO 1
      KOUNT=0
      NUMF=1
    1 CONTINUE
      CALL SECOND (T4)
      TIME=T4-T3
      IF (IPRINT.EQ.0) GO TO 2
      CALL WRITE2 (X,N,G,F,NUMF,KOUNT,TIME)
    2 CONTINUE
C
C        RESET ITERATION COUNTER AND GENERATE IDENTITY MATRIX
      IER=0
      KK=0
      N2=N+N
      N3=N2+N
      N31=N3+1
    3 K=N31
      DO 6 J=1,N
      H(K)=1.
      NJ=N-J
      IF (NJ) 7,7,4
    4 DO 5 L=1,NJ
      KL=K+L
      H(KL)=0.
    5 CONTINUE
      K=KL+1
    6 CONTINUE
C
C        START ITERATION LOOP
    7 IF (KOUNT.EQ.0) GO TO 8
      IF (KK.NE.IPRINT) GO TO 8
      KK=0
      CALL SECOND (T4)
      TIME=T4-T3
      CALL WRITE2 (X,N,G,F,NUMF,KOUNT,TIME)
    8 CONTINUE
      IF (KAML.EQ.0) GO TO 11
      FTE=ABS(F-OLDF)
      FQE=EPS*ABS(F)
      IF (FTE.LT.FQE) GO TO 9
      KQ=0
      GO TO 10
    9 KQ=KQ+1
   10 CONTINUE
   11 KAML=KAML+1
      IF (KQ.EQ.4) GO TO 71
      KOUNT=KOUNT+1
      KK=KK+1
C
C        SAVE FUNCTION VALUE, ARGUMENT VECTOR AND GRADIENT VECTOR
      OLDF=F
      DO 15 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 14 L=1,N
      T=T-G(L)*H(K)
      IF (L-J) 12,13,13
   12 K=K+N-L
      GO TO 14
   13 K=K+1
   14 CONTINUE
      H(J)=T
   15 CONTINUE
```

```
      SUBROUTINE FMFP (FUNCT,N,X,F,G,EST,EPS,LIMIT,IER,H,IPRINT)
      COMMON /BLK1/ NUMF
      COMMON /BLK/ KO
      COMMON /SPL/ ALPHA,BETA,IFLAGA,IFLAGB,ICHEK,AB,KKK
C        SUBROUTINE FMFP
C
C        PURPOSE
C           TO FIND A LOCAL MINIMUM OF A FUNCTION OF SEVERAL VARIABLES
C           BY THE METHOD OF FLETCHER AND POWELL
C
C        USAGE
C           CALL FMFP(FUNCT,N,X,F,G,EST,EPS,LIMIT,IER,H)
C
C        DESCRIPTION OF PARAMETERS
C           FUNCT  - USER-WRITTEN SUBROUTINE CONCERNING THE FUNCTION TO
C                    BE MINIMIZED. IT MUST BE OF THE FORM
C                    SUBROUTINE FUNCT(N,ARG,VAL,GRAD)
C                    AND MUST SERVE THE FOLLOWING PURPOSE
C                    FOR EACH N-DIMENSIONAL ARGUMENT VECTOR  ARG,
C                    FUNCTION VALUE AND GRADIENT VECTOR MUST BE COMPUTED
C                    AND, ON RETURN, STORED IN VAL AND GRAD RESPECTIVELY
C           N      - NUMBER OF VARIABLES
C           X      - VECTOR OF DIMENSION N CONTAINING THE INITIAL
C                    ARGUMENT WHERE THE ITERATION STARTS. ON RETURN,
C                    X HOLDS THE ARGUMENT CORRESPONDING TO THE
C                    COMPUTED MINIMUM FUNCTION VALUE
C           F      - SINGLE VARIABLE CONTAINING THE MINIMUM FUNCTION
C                    VALUE ON RETURN, I.E. F=F(X).
C           G      - VECTOR OF DIMENSION N CONTAINING THE GRADIENT
C                    VECTOR CORRESPONDING TO THE MINIMUM ON RETURN,
C                    I.E. G=G(X).
C           EST    - IS AN ESTIMATE OF THE MINIMUM FUNCTION VALUE.
C           EPS    - TESTVALUE REPRESENTING THE EXPECTED ABSOLUTE ERROR.
C                    A REASONABLE CHOICE IS 10**(-6), I.E.
C                    SOMEWHAT GREATER THAN 10**(-D), WHERE D IS THE
C                    NUMBER OF SIGNIFICANT DIGITS IN FLOATING POINT
C                    REPRESENTATION.
C           LIMIT  - MAXIMUM NUMBER OF ITERATIONS.
C           IER    - ERROR PARAMETER
C                    IER = 0 MEANS CONVERGENCE WAS OBTAINED
C                    IER = 1 MEANS NO CONVERGENCE IN LIMIT ITERATIONS
C                    IER =-1 MEANS ERRORS IN GRADIENT CALCULATION
C                    IER = 2 MEANS LINEAR SEARCH TECHNIQUE INDICATES
C                    IT IS LIKELY THAT THERE EXISTS NO MINIMUM.
C           H      - WORKING STORAGE OF DIMENSION N*(N+7)/2.
C
C        REMARKS
C            I) THE SUBROUTINE NAME REPLACING THE DUMMY ARGUMENT  FUNCT
C               MUST BE DECLARED AS EXTERNAL IN THE CALLING PROGRAM.
C           II) IER IS SET TO 2 IF , STEPPING IN ONE OF THE COMPUTED
C               DIRECTIONS, THE FUNCTION WILL NEVER INCREASE WITHIN
C               A TOLERABLE RANGE OF ARGUMENT.
C               IER = 2 MAY OCCUR ALSO IF THE INTERVAL WHERE F
C               INCREASES IS SMALL AND THE INITIAL ARGUMENT WAS
C               RELATIVELY FAR AWAY FROM THE MINIMUM SUCH THAT THE
C               MINIMUM WAS OVERLEAPED. THIS IS DUE TO THE SEARCH
C               TECHNIQUE WHICH DOUBLES THE STEPSIZE UNTIL A POINT
C               IS FOUND WHERE THE FUNCTION INCREASES.
C
C        SUBROUTINES AND FUNCTION SUBPROGRAMS REQUIRED
C           FUNCT
C
C        METHOD
C           THE METHOD IS DESCRIBED IN THE FOLLOWING ARTICLE
C           R. FLETCHER AND M.J.D. POWELL, A RAPID DESCENT METHOD FOR
C           MINIMIZATION,
C           COMPUTER JOURNAL VOL.6, ISS. 2, 1963, PP.163-168.
C
C     ..................................................................
C
C
C        DIMENSIONED DUMMY VARIABLES
      DIMENSION H(1), X(1), G(1)
      KQ=0
      KAML=0
C
```

```
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 16 J=1,N
      HNRM=HNRM+ABS(H(J))
      GNRM=GNRM+ABS(G(J))
      DY=DY+H(J)*G(J)
   16 CONTINUE
C
C        REPEAT SEARCH IN DIRECTION OF STEEPEST DESCENT IF DIRECTIONAL
C        DERIVATIVE APPEARS TO BE POSITIVE OR ZERO.
      IF (DY) 17,61,61
C
C        REPEAT SEARCH IN DIRECTION OF STEEPEST DESCENT IF DIRECTION
C        VECTOR H IS SMALL COMPARED TO GRADIENT VECTOR G.
   17 IF (HNRM/GNRM-EPS) 61,61,18
C
C        SEARCH MINIMUM ALONG DIRECTION H
C
C        SEARCH ALONG H FOR POSITIVE DIRECTIONAL DERIVATIVE
   18 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) 21,21,19
   19 IF (ALFA-AMBDA) 20,21,21
   20 AMBDA=ALFA
   21 ALFA=0.
C
C        SAVE FUNCTION AND DERIVATIVE VALUES FOR OLD ARGUMENT
   22 FX=FY
      DX=DY
C
C        STEP ARGUMENT ALONG H
      DO 23 I=1,N
      X(I)=X(I)+AMBDA*H(I)
   23 CONTINUE
C
C        COMPUTE FUNCTION VALUE AND GRADIENT FOR NEW ARGUMENT
      CALL FUNCT (N,X,F,G)
      IF (IFLAGB.EQ.1) RETURN
      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 24 I=1,N
      DY=DY+G(I)*H(I)
   24 CONTINUE
      IF (DY) 25,45,28
C
C        TERMINATE SEARCH ALSO IF THE FUNCTION VALUE INDICATES THAT
C        A MINIMUM HAS BEEN PASSED
   25 IF (FY-FX) 26,28,28
C
C        REPEAT SEARCH AND DOUBLE STEPSIZE FOR FURTHER SEARCHES
   26 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) 22,22,27
C
C        LINEAR SEARCH TECHNIQUE INDICATES THAT NO MINIMUM EXISTS
   27 IER=2
      GO TO 66
```

```
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
   28 T=0.
   29 IF (AMBDA) 30,45,30
   30 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) 61,31,31
   31 W=ALFA*SQRT(DALFA)
      ALFA=DY-DX+W+W
      IF (ALFA) 32,33,32
   32 ALFA=(DY-Z+W)/ALFA
      GO TO 34
   33 ALFA=(Z+DY-W)/(Z+DX+Z+DY)
   34 ALFA=ALFA*AMBDA
      DO 35 I=1,N
      X(I)=X(I)+(T-ALFA)*H(I)
   35 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)
      IF (IFLAGB.EQ.1) RETURN
      IF (F-FX) 36,36,37
   36 IF (F-FY) 45,45,37
   37 DALFA=0.
      DO 38 I=1,N
      DALFA=DALFA+G(I)*H(I)
   38 CONTINUE
      IF (DALFA) 39,42,42
   39 IF (F-FX) 41,40,42
   40 IF (DX-DALFA) 41,45,41
   41 FX=F
      DX=DALFA
      T=ALFA
      AMBDA=ALFA
      GO TO 29
   42 IF (FY-F) 44,43,44
   43 IF (DY-DALFA) 44,45,44
   44 FY=F
      DY=DALFA
      AMBDA=AMBDA-ALFA
      GO TO 28
C
C        TERMINATE, IF FUNCTION HAS NOT DECREASED DURING LAST ITERATION
   45 IF (OLDF-F+EPS) 61,46,46
C
C        COMPUTE DIFFERENCE VECTORS OF ARGUMENT AND GRADIENT FROM
C        TWO CONSECUTIVE ITERATIONS
   46 DO 47 J=1,N
      K=N+J
      H(K)=G(J)-H(K)
      K=N+K
      H(K)=X(J)-H(K)
   47 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) 51,48,48
   48 T=0.
      Z=0.
      DO 49 J=1,N
      K=N+J
      W=H(K)
```

```
      K=K+N
      T=T+ABS(H(K))
      Z=Z+W*H(K)
   49 CONTINUE
      IF (HNRM-EPS) 50,50,51
   50 IF (T-EPS) 66,66,51
C
C        TERMINATE, IF NUMBER OF ITERATIONS WOULD EXCEED  LIMIT
   51 IF (KOUNT-LIMIT) 52,59,59
C
C        PREPARE UPDATING OF MATRIX H
   52 ALFA=0.
      DO 56 J=1,N
      K=J+N3
      W=0.
      DO 55 L=1,N
      KL=N+L
      W=W+H(KL)*H(K)
      IF (L-J) 53,54,54
   53 K=K+N-L
      GO TO 55
   54 K=K+1
   55 CONTINUE
      K=N+J
      ALFA=ALFA+W*H(K)
      H(J)=W
   56 CONTINUE
C
C        REPEAT SEARCH IN DIRECTION OF STEEPEST DESCENT IF RESULTS
C        ARE NOT SATISFACTORY
      IF (Z*ALFA) 57,3,57
C
C        UPDATE MATRIX H
   57 K=N31
      DO 58 L=1,N
      KL=N2+L
      DO 58 J=L,N
      NJ=N2+J
      H(K)=H(K)+H(KL)*H(NJ)/Z-H(L)*H(J)/ALFA
   58 K=K+1
      GO TO 7
C        END OF ITERATION LOOP
C
C        NO CONVERGENCE AFTER  LIMIT  ITERATIONS
   59 IER=1
      IF (KK.NE.IPRINT) GO TO 60
      CALL WRITE2 (X,N,G,F,NUMF,KOUNT)
   60 CONTINUE
      GO TO 66
C
C        RESTORE OLD VALUES OF FUNCTION AND ARGUMENTS
   61 DO 62 J=1,N
      K=N2+J
      X(J)=H(K)
   62 CONTINUE
      CALL FUNCT (N,X,F,G)
      IF (IFLAGB.EQ.1) RETURN
      NUMF=NUMF+1
C
C        REPEAT SEARCH IN DIRECTION OF STEEPEST DESCENT IF DERIVATIVE
C        FAILS TO BE SUFFICIENTLY SMALL
      IF (GNRM-EPS) 65,65,63
C
C        TEST FOR REPEATED FAILURE OF ITERATION
   63 IF (IER) 66,64,64
   64 IER=-1
      GO TO 3
   65 IER=0
   66 II=IER+2
      IF (II.EQ.2) KO=1
      IF (IPRINT.EQ.0) RETURN
      GO TO (67,68,69,70), II
   67 WRITE (6,74) IER
      GO TO 72
```

```
68    WRITE (6,75) IER
      GO TO 72
69    WRITE (6,76) IER
      GO TO 72
70    WRITE (6,77) IER
71    WRITE (6,73)
      KO=1
72    RETURN
C
C
C
73    FORMAT (1HO,*THERE IS NO SIGNIFICANT DECREASE IN THE FUNCTION VALU
     1E OPTIMUM IS ASSUMED TO HAVE BEEN REACHED*)
74    FORMAT (1HO,*IER=*,I2,* ERROR IN GRADIENTS CALCULATIONS*)
75    FORMAT (1HO,*IER=*,I2,* CRITERION FOR OPTIMUM HAS BEEN SATISFIED*)
76    FORMAT (1HO,*IER=*,I2,* MAXIMUM NUMBER OF ALLOWABLE ITERATIONS HAS
     1 BEEN EXCEEDED*)
77    FORMAT (1HO,*IER=*,I2,* CHANGE IN ARGUMENTS GETS TOO LARGE, LINEAR
     1 SEARCH INDICATES THAT NO MINIMUM EXISTS*)
      END
```