

**RESEARCH INSTITUTE FOR INFORMATICS
BUCHAREST**

UNO

**A Package for Unconstrained Optimization
Method using
Direct Searching Techniques**

Neculai Andrei

*Research Institute for Informatics,
Center for Advanced Modeling and Optimization,
8-10, Averescu Avenue, Bucharest 1, Romania,
Academy of Romanian Scientists
E-mail: nandrei@ici.ro*

Raport Tehnic No. 1 / 1991

1 April, 1991

Bucharest



New version: April 20, 2011

CONTENTS

UNOINT.DOC	<p>File with Fortran 77 codes (in DOC format) for UNO package including subroutines for direct searching techniques: <i>Hook-Jeeves (searching of form)</i>, <i>Rosenbrock (rotation of coorodnation system)</i>, <i>Powell (conjugate directions)</i>, <i>Nelder-Mead (simplex)</i> and <i>Parallel with Axes Searching</i>.</p> <p>All these methods are implemented with different one-dimensional optimization methods: <i>Golden Search</i>, <i>Fibonacci</i>, <i>Quadratic Fitting of Powell</i>, or <i>simple steplength equal to 1</i>.</p> <p>This package is in interactive form. The user must specify: <i>the searching method</i>, <i>the one-dimensional searching techniques</i>, <i>the number of variables</i>, <i>the accuracy</i> and <i>the initial point</i>.</p> <p>The optimization results are printed into the output file: <i>print.dat</i>.</p>
UNOINT.FOR	<p>File with Fortran 77 codes including: <i>Main program</i>, <i>HOOKJ</i>, <i>ROSE</i>, <i>POWEL</i>, <i>NELMED</i>, <i>CPA</i>; <i>L1 (golden search)</i>, <i>L2 (Fibonacci)</i>, <i>L3 (Quadratic interpolation)</i>, <i>L4 (Dichotomous search)</i>, <i>EXETIM (CPU time computation)</i>.</p> <p>This version of the package is <u>an interactive one</u>.</p>
UNO.FOR	<p>File with Fortran 77 codes including: <i>Main program</i>, <i>HOOKJ</i>, <i>ROSE</i>, <i>POWEL</i>, <i>NELMED</i>, <i>CPA</i>; <i>L1 (golden search)</i>, <i>L2 (Fibonacci)</i>, <i>L3 (Quadratic interpolation)</i>, <i>L4 (Dichotomous search)</i>, <i>EXETIM (CPU time computation)</i>.</p> <p>A <u>non-interactive</u> version with 78 examples of functions created in August 8, 2007. The name of the functions is in <i>FUNCNAME.TXT</i> file.</p>
CPA.FOR	Fortran subroutine with parallel with axes searching.
HOOKJ.FOR	Fortran subroutine with searching by Hook-Jeeves method.
ROSE.FOR	Fortran subroutine with Rosenbrock method.
NELMED.FOR	Fortran subroutine with Nelder-Mead method.
POWEL.FOR	Fortran subroutine with Powell method.
L1.FOR	Fortran subroutine with golden search method for one-dimensional minimization.
L2.FOR	Fortran subroutine with Fibonacci method for one-dimensional minimization.
L3.FOR	Fortran subroutine with quadratic interpolation method of Powell for one-dimensional minimization.
L4.FOR	Fortran subroutine with dichotomous method for one-dimensional minimization.
EXETIM.FOR	Fortran subroutine for execution time computation.
FUNC1-17.FOR	17 Fortran examples with different functions to be minimized.
EXAMPLES.DOC	File with solution of 6 examples of function minimization with different direct search methods.
FUNCNAME.TXT	File with the name of the function used in UNO package.
UNODEPT.FOR	Interactive version of UNO for solving the Elastic-Plastic Torsion Problem from MINPACK collection with a discretization $n_x=n_y=20$ points.

```

C*****
C
C
C      U      U      N      N      OOOOO
C      U      U      NN     N      O      O
C      U      U      N N    N      O      O
C      U      U      N  N  N      O      O
C      U      U      N   N N      O      O
C      UUUUUUU      N      NN      OOOOO
C      -----
C
C      INTERACTIVE VERSION
C
C      Neculai Andrei,
C      Research Institute for Informatics
C      Advanced Modeling and Optimization Laboratory
C      8-10, Bdl. Maresal Alexandru Averescu
C      71316, Bucharest 1, Romania
C
C
C      Main program for:
C
C      UNCONSTRAINT OPTIMIZATION METHODS
C
C      using
C
C      DIRECT SEARCHING TECHNIQUES
C
C      The following techniques are implemented:
C      - Hook-Jeeves - form searching,
C      - Rosenbrook - rotation of coordinates,
C      - Powell - conjugate directions,
C      - Nelder-Mead - Simplex,
C      - Parallel with Axes Searching
C
C      These methods are implemented with different onedimensional
C      optimization methods like golden section, Fibonacci,
C      quadratic fitting of Powell, or simple lambda =1.0.
C
C      Version A (Interactive version)
C
C
C      April 1991
C*****
C
C
C      The following subroutines are called:
C
C      HOOKJ      Hook-Jeeves.
C      ROSE       Rosenbrook.
C      NELMED     Nelder-Mead.
C      POWEL      Powell.
C      CPA        Parallel with axes.
C
C      L1         Golden searching.
C      L2         Fibonacci searching.
C      L3         Quadratic interpolation of Powell.
C      L4         Dichotomous searching.
C
C*****

```

```

C
C
DOUBLE PRECISION X(40),S(40),H(41),P(1600)
DOUBLE PRECISION X1(40),X2(40),X3(40),Y(1600),Z(1600)
DOUBLE PRECISION EPS

C
COMMON /PREC/EPS
COMMON /TYPE/NO
COMMON /WRIT/NX

integer*4 gh,gm,gs,gc, ght,gmt,gst,gct, timpexp

C
CHARACTER*1 IDC(7),ID,HJC(7),HJ

C
DATA IDC/'J','R','P','S','C','E','H'/
DATA HJC/'U','G','F','P','D','E','H'/

C
C
NX=0
NO=0
WRITE(NX,11)
11  FORMAT(/,14X,'UNO - Package for Unconstraint Optimization'./,
+    14X,'*****'/)
C
10  WRITE(*,1)' Unconstraint Optimization (J,R,P,S,C,End,Help) ?'
1   FORMAT(A,$)

READ(5,2,END=7)ID
2   FORMAT(A1)
IS=1
DO 5 I=1,7
IF(ID.NE.IDC(I)) GO TO 5
IS=0
GO TO 6
5   CONTINUE
6   IF(IS.EQ.1) THEN
WRITE(NX,3)
3   FORMAT(10X,'Error. Please try again'/)
GO TO 10
ENDIF
GO TO 8
7   WRITE(NX,3)
GO TO 10
8   CONTINUE
C
IF(ID.EQ.'H') THEN
WRITE(NX,20)
20  FORMAT(/10X,'Unconstraint Optimization Facilities:')
WRITE(NX,21)
21  FORMAT(10X,'*****')
WRITE(NX,22)
22  FORMAT(10X,'* J = Hook-Jeeves Method. Form searching.  *')
WRITE(NX,23)
23  FORMAT(10X,'* R = Rosenbrock Method. Coordinates rotation.  *')
WRITE(NX,24)
24  FORMAT(10X,'* P = Powell Method. Conjugate Directions.  *')
WRITE(NX,25)
25  FORMAT(10X,'* S = Nelder-Mead Method. Simplex Method.  *')
WRITE(NX,26)
26  FORMAT(10X,'* C = Parallel Searching with Axes Method.  *')

```

```

WRITE(NX,27)
27  FORMAT(10X,'* End = End of this session of optimization.  *')
    WRITE(NX,28)
28  FORMAT(10X,'* Help = Help for this optimization facilities:  *')
    WRITE(NX,21)
    GO TO 10
    ENDIF

C
C
C
C----- Start Hook-Jeeves Method
C
    IF(ID.EQ.'J') THEN
30    WRITE(*,1)' Hook-Jeeves Method (U,G,F,P,D,End,Help) ?'
        READ(5,2,END=33) HJ
        IS=1
        DO 31 I=1,7
            IF(HJ.NE.HJC(I)) GO TO 31
            IS=0
            GO TO 32
31        CONTINUE
32        IF(IS.EQ.1) THEN
            WRITE(NX,3)
            GO TO 30
            ENDIF
            GO TO 34
33        WRITE(NX,3)
            GO TO 30
34        CONTINUE
C
        IF(HJ.EQ.'E') GO TO 10
C
        IF(HJ.EQ.'H') THEN
            WRITE(NX,35)
35        FORMAT(/10X,'Hook-Jeeves Method Facilities:')
            WRITE(NX,36)
36        FORMAT(10X,'*****')
            WRITE(NX,37)
37        FORMAT(10X,'* U : Lambda* = 1.0                *')
            WRITE(NX,38)
38        FORMAT(10X,'* G : Lambda* obtained by Golden Searchnig  *')
            WRITE(NX,39)
39        FORMAT(10X,'* F : Lambda* obtained by Fibonacci Searching *')
            WRITE(NX,40)
40        FORMAT(10X,'* P : Lambda* obtained by Powell Interpolation *')
            WRITE(NX,41)
41        FORMAT(10X,'* D : Lambda* obtained by Dichotomous Searching *')
            WRITE(NX,42)
42        FORMAT(10X,'* End = End of this optimization session      *')
            WRITE(NX,43)
43        FORMAT(10X,'* Help = Help of this optimization facilities  *')
            WRITE(NX,36)
            GO TO 30
            ENDIF
C
        WRITE(*,1)' Number of variables = ? '
        READ(5,50)N
50        FORMAT(I5)
C
        WRITE(*,1)' Accuracy  EPS = ? '

```

```

      READ(5,51)EPS
51      FORMAT(g30.18)
C
      WRITE(NX,54)N
54      FORMAT(3X,'Initial Point (' ,I2,' - real values)')
C
      DO 53 I=1,N
      READ(5,52)X(I)
52      FORMAT(F12.6)
53      CONTINUE
C
      DO 56 I=1,N
56      X1(I)=X(I)
C
      IF(HJ.EQ.'U') THEN
      ITECH=5
      call gettim(gh,gm,gs,gc)
      CALL HOOKJ(N,X1,X2,Y,Z,S,ITECH)
      call gettim(ght,gmt,gst,gct)
      call exetim(gh,gm,gs,gc, ght,gmt,gst,gct)
      timpexp = ght*360000 + gmt*6000 + gst*100 + gct
      write(no,995) timpexp
      GO TO 30
      ENDIF
C
      IF(HJ.EQ.'G') THEN
      ITECH=1
      call gettim(gh,gm,gs,gc)
      CALL HOOKJ(N,X1,X2,Y,Z,S,ITECH)
      call gettim(ght,gmt,gst,gct)
      call exetim(gh,gm,gs,gc, ght,gmt,gst,gct)
      timpexp = ght*360000 + gmt*6000 + gst*100 + gct
      write(no,995) timpexp
      GO TO 30
      ENDIF
C
      IF(HJ.EQ.'F') THEN
      ITECH=2
      call gettim(gh,gm,gs,gc)
      CALL HOOKJ(N,X1,X2,Y,Z,S,ITECH)
      call gettim(ght,gmt,gst,gct)
      call exetim(gh,gm,gs,gc, ght,gmt,gst,gct)
      timpexp = ght*360000 + gmt*6000 + gst*100 + gct
      write(no,995) timpexp
      GO TO 30
      ENDIF
C
      IF(HJ.EQ.'P') THEN
      ITECH=3
      call gettim(gh,gm,gs,gc)
      CALL HOOKJ(N,X1,X2,Y,Z,S,ITECH)
      call gettim(ght,gmt,gst,gct)
      call exetim(gh,gm,gs,gc, ght,gmt,gst,gct)
      timpexp = ght*360000 + gmt*6000 + gst*100 + gct
      write(no,995) timpexp
      GO TO 30
      ENDIF
C
      IF(HJ.EQ.'D') THEN
      ITECH=4

```

```

        call gettim(gh,gm,gs,gc)
        CALL HOOKJ(N,X1,X2,Y,Z,S,ITECH)
    call gettim(ght,gmt,gst,gct)
    call exetim(gh,gm,gs,gc, ght,gmt,gst,gct)
    timpexp = ght*360000 + gmt*6000 + gst*100 + gct
    write(no,995) timpexp
        GO TO 30
    ENDIF
C
    ENDIF

C----- End of Hook-Jeeves Method.
C
C
C----- Start of Rosenbrock's Method.
C
C
    IF(ID.EQ.'R') THEN
        WRITE(NX,90)
90    FORMAT(10X,'Rosenbrock Method of Minimization.')
        WRITE(*,1)' Number of variables = ? '
        READ(5,50)N
        NN=N*N
        WRITE(*,1)' Accuracy EPS = ?'
        READ(5,51)EPS
        WRITE(NX,54)N
        DO 91 I=1,N
            READ(5,52)X(I)
91    CONTINUE
C
        call gettim(gh,gm,gs,gc)
        CALL ROSE(N,NN,X,X1,X2,S,H,Y,Z,P,X3)
        call gettim(ght,gmt,gst,gct)
        call exetim(gh,gm,gs,gc, ght,gmt,gst,gct)
        timpexp = ght*360000 + gmt*6000 + gst*100 + gct
        write(no,995) timpexp
C
        GO TO 10
    ENDIF
C
C----- End of Rosenbrock's Method.
C
C
C----- Start Nelder - Mead Method.
C
C
    IF(ID.EQ.'S') THEN
        WRITE(NX,92)
92    FORMAT(10X,'Simplex Method of Nelder-Mead Minimization')
        WRITE(*,1)' Number of variables = ? '
        READ(5,50)N
        N1=N+1
        NN=N*(N+1)
        WRITE(*,1)' Accuracy EPS = ?'
        READ(5,51)EPS
        WRITE(NX,54)N
        DO 93 I=1,N
            READ(5,52)X(I)
93    CONTINUE
C

```

```

call gettim(gh,gm,gs,gc)
  CALL NELMED(N,N1,NN,X,X1,X2,X3,Y,H)
call gettim(ght,gmt,gst,gct)
call exetim(gh,gm,gs,gc, ght,gmt,gst,gct)
timpexp = ght*360000 + gmt*6000 + gst*100 + gct
write(no,995) timpexp
C
  GO TO 10
ENDIF
C
C----- End Nelder-Mead Method.
C
C
C----- Start Powell Method.
C
C
C
70  IF(ID.EQ.'P') THEN
    WRITE(*,1)' Powell Method (G,F,P,D,End,Help) ?'
    READ(5,2,ERR=73)HJ
    IS=1
    DO 71 I=1,7
      IF(HJ.NE.HJC(I)) GO TO 71
    IS=0
    GO TO 72
71  CONTINUE
72  IF(IS.EQ.1) THEN
    WRITE(NX,3)
    GO TO 70
  ENDIF
    GO TO 74
73  WRITE(NX,3)
    GO TO 70
74  CONTINUE
C
  IF(HJ.EQ.'E') GO TO 10
C
  IF(HJ.EQ.'H') THEN
    WRITE(NX,75)
75  FORMAT(/10X,'Powell Method Facilities:')
    WRITE(NX,36)
    WRITE(NX,38)
    WRITE(NX,39)
    WRITE(NX,40)
    WRITE(NX,41)
    WRITE(NX,42)
    WRITE(NX,43)
    WRITE(NX,36)
    GO TO 70
  ENDIF
C
  WRITE(*,1)' Number of variables = ?'
  READ(5,50)N
C
  NN=N*N
C
  WRITE(*,1)' Accuracy EPS = ?'
  READ(5,51)EPS
C
  WRITE(NX,54)N

```



```

DO 76 I=1,N
READ(5,52)X(I)
76 CONTINUE
C
IF(HJ.EQ.'G') THEN
ITECH=1
call gettim(gh,gm,gs,gc)
CALL POWEL(N,NN,X,X1,X2,X3,S,Y,ITECH)
call gettim(ght,gmt,gst,gct)
call exetim(gh,gm,gs,gc, ght,gmt,gst,gct)
timpexp = ght*360000 + gmt*6000 + gst*100 + gct
write(no,995) timpexp
GO TO 70
ENDIF
C
IF(HJ.EQ.'F') THEN
ITECH=2
call gettim(gh,gm,gs,gc)
CALL POWEL(N,NN,X,X1,X2,X3,S,Y,ITECH)
call gettim(ght,gmt,gst,gct)
call exetim(gh,gm,gs,gc, ght,gmt,gst,gct)
timpexp = ght*360000 + gmt*6000 + gst*100 + gct
write(no,995) timpexp
GO TO 70
ENDIF
C
IF(HJ.EQ.'P') THEN
ITECH=3
call gettim(gh,gm,gs,gc)
CALL POWEL(N,NN,X,X1,X2,X3,S,Y,ITECH)
call gettim(ght,gmt,gst,gct)
call exetim(gh,gm,gs,gc, ght,gmt,gst,gct)
timpexp = ght*360000 + gmt*6000 + gst*100 + gct
write(no,995) timpexp
GO TO 70
ENDIF
C
IF(HJ.EQ.'D') THEN
ITECH=4
call gettim(gh,gm,gs,gc)
CALL POWEL(N,NN,X,X1,X2,X3,S,Y,ITECH)
call gettim(ght,gmt,gst,gct)
call exetim(gh,gm,gs,gc, ght,gmt,gst,gct)
timpexp = ght*360000 + gmt*6000 + gst*100 + gct
write(no,995) timpexp
GO TO 70
ENDIF
C
ENDIF
C
C
C----- End of Powell Method.
C
C
C----- Start Parallel with Axes
C Searching Method.
C
C
IF(ID.EQ.'C') THEN

```

```

60    WRITE(*,1)' Parallel Axes Searching Method (G,F,P,D,End,Help) ?'
      READ(5,2,END=63)HJ
      IS=1
      DO 61 I=1,7
      IF(HJ.NE.HJC(I)) GO TO 61
      IS=0
      GO TO 62
61    CONTINUE
62    IF(IS.EQ.1) THEN
      WRITE(NX,3)
      GO TO 60
    ENDIF
      GO TO 64
63    WRITE(NX,3)
      GO TO 60
64    CONTINUE
C
      IF(HJ.EQ.'E') GO TO 10
C
      IF(HJ.EQ.'H') THEN
      WRITE(NX,65)
65    FORMAT(/10X,'Parallel Searching Method Facilities:')
      WRITE(NX,36)
C      WRITE(NX,37)
      WRITE(NX,38)
      WRITE(NX,39)
      WRITE(NX,40)
      WRITE(NX,41)
      WRITE(NX,42)
      WRITE(NX,43)
      WRITE(NX,36)
      GO TO 60
    ENDIF
C
      WRITE(*,1)' Number of variables = ? '
      READ(5,50)N
C
      WRITE(*,1)' Accuracy EPS = ? '
      READ(5,51)EPS
C
      WRITE(NX,54)N
C
      DO 83 I=1,N
      READ(5,52)X(I)
83    CONTINUE
C
C
      IF(HJ.EQ.'U') THEN
      ITECH=5
      call gettim(gh,gm,gs,gc)
      CALL CPA(N,X,X1,X2,S,ITECH)
      call gettim(ght,gmt,gst,gct)
      call exetim(gh,gm,gs,gc, ght,gmt,gst,gct)
      timpexp = ght*360000 + gmt*6000 + gst*100 + gct
      write(no,995) timpexp
      GO TO 60
    ENDIF
C
      IF(HJ.EQ.'G') THEN
      ITECH=1

```

```

        call gettim(gh,gm,gs,gc)
        CALL CPA(N,X,X1,X2,S,ITECH)
    call gettim(ght,gmt,gst,gct)
    call exetim(gh,gm,gs,gc, ght,gmt,gst,gct)
    timpexp = ght*360000 + gmt*6000 + gst*100 + gct
    write(no,995) timpexp
        GO TO 60
    ENDIF
C
    IF(HJ.EQ.'F') THEN
        ITECH=2
        call gettim(gh,gm,gs,gc)
        CALL CPA(N,X,X1,X2,S,ITECH)
    call gettim(ght,gmt,gst,gct)
    call exetim(gh,gm,gs,gc, ght,gmt,gst,gct)
    timpexp = ght*360000 + gmt*6000 + gst*100 + gct
    write(no,995) timpexp
        GO TO 60
    ENDIF
C
    IF(HJ.EQ.'P') THEN
        ITECH=3
        call gettim(gh,gm,gs,gc)
        CALL CPA(N,X,X1,X2,S,ITECH)
    call gettim(ght,gmt,gst,gct)
    call exetim(gh,gm,gs,gc, ght,gmt,gst,gct)
    timpexp = ght*360000 + gmt*6000 + gst*100 + gct
    write(no,995) timpexp
        GO TO 60
    ENDIF
C
    IF(HJ.EQ.'D') THEN
        ITECH=4
        call gettim(gh,gm,gs,gc)
        CALL CPA(N,X,X1,X2,S,ITECH)
    call gettim(ght,gmt,gst,gct)
    call exetim(gh,gm,gs,gc, ght,gmt,gst,gct)
    timpexp = ght*360000 + gmt*6000 + gst*100 + gct
    write(no,995) timpexp
        GO TO 60
    ENDIF
C
    ENDIF
C-----End of Parallel with Axes Searching Method.
C
C
C
    IF(ID.EQ.'E') GO TO 100
C
C
100 CONTINUE

c   write(no,995) timpexp
995  format(1x,'##### ----- Time=',i9,'c')

    STOP
    END
C
C***** Last card of UNO.FOR.

```



```

      READ(5,112,ERR=113) FD
112  FORMAT(A1)
      IF(FD.NE.'F'.AND.FD.NE.'D') THEN
      WRITE(NX,114)
114  FORMAT(10X,'Error.  Please try again')
      GO TO 110
      ENDIF
      IF(FD.EQ.'F') NO=6
      IF(FD.EQ.'D') NO=6
      GO TO 115
113  WRITE(NX,114)
      GO TO 110
115  CONTINUE
C
      IF(NO.EQ.6) THEN
      OPEN(6,FILE='PRINT.DAT',ACCESS='SEQUENTIAL',
+      FORM='FORMATTED')
      ENDIF
C
      CALL FUNC(N,X,F)
      NFUNC=NFUNC+1
C
      WRITE(NO,50)
50  FORMAT(10X,'***** Parallel with Axes Optimization Method *****')
      IF(ITECH.EQ.1) WRITE(NO,161)
      IF(ITECH.EQ.2) WRITE(NO,162)
      IF(ITECH.EQ.3) WRITE(NO,163)
      IF(ITECH.EQ.4) WRITE(NO,164)
      WRITE(NO,51)
51  FORMAT(10X,'Initial Point :')
      WRITE(NO,52)(X(I),I=1,N)
52  FORMAT(10X,F18.6)
      WRITE(NO,54)EPS
54  FORMAT(10X,'Accuracy EPS = ',g30.15)
      WRITE(NO,55)F
55  FORMAT(10X,'Function Value in Initial point = ',F18.6)
C
1  ITER=ITER+1
C
      IF(ITER.GT.1000) THEN
      WRITE(NX,56)
56  FORMAT(10X,'Too many iterations.',/,
+      10X,'This function is too complicated for the power',/,
+      10X,'of the Parallel with Axes Searching Method.',/,
+      10X,'May be this function is a ridge or a valley.',/,
+      10X,'Please consider another initial point, or',/,
+      10X,'try another optimization method.')
      RETURN
      ENDIF
C
      DO 2 I=1,N
2  D(I)=0.0D0
      J1=ITER/N
      J2=J1*N
      J=ITER-J2
      IF(J.EQ.0) J=N
      D(J)=1.0D0
C
      DO 3 I=1,N
3  X1(I)=X(I)+EP*D(I)

```

```

        CALL FUNC(N,X1,F1)
        NFUNC=NFUNC+1
        DO 4 I=1,N
4       X1(I)=X(I)-EP*D(I)
        CALL FUNC(N,X1,F2)
        NFUNC=NFUNC+1
C
        IF(F1.GT.F2) THEN
        DO 5 I=1,N
5       D(I)=-D(I)
        ENDIF
C
C-----
        GO TO (171,172,173,174) ITECH
171     CALL L1(N,X,D,BL)
        GO TO 180
172     CALL L2(N,X,D,BL)
        GO TO 180
173     CALL L3(N,X,D,BL)
        IF(ICOD.EQ.1) RETURN
        GO TO 180
174     CALL L4(N,X,D,BL)
C
180     CONTINUE
C-----
C
        DO 6 I=1,N
6       X2(I)=X(I)+BL*D(I)
        CALL FUNC(N,X2,FI)
        NFUNC=NFUNC+1
C
        DO 7 I=1,N
7       X(I)=X2(I)
C
        IF(DABS(F-FI).LE.EPS) GO TO 10
        F=FI
C
        GO TO 1
C
10     CONTINUE
C
        WRITE(NO,79)
79     FORMAT(/5X,'-----')
        WRITE(NO,80) ITER
80     FORMAT(10X,'Iterations = ',I4)
        WRITE(NO,85)NFUNC
85     FORMAT(10X,'No.Func. Eval = ',I4)
        WRITE(NO,81) EPS
81     FORMAT(10X,'Precision = ',F18.6)
        WRITE(NO,82)
82     FORMAT(10X,'***** Minimum Point *****')
        WRITE(NO,83)(X(I),I=1,N)
83     FORMAT(10X,F18.6)
        WRITE(NO,84)F
84     FORMAT(10X,'Function Value in Minimum Point =',F18.6)
C
        RETURN
        END
C
C*****Last card of CPA.FOR

```

```

C*****
C
C   Searching method by Hook-Jeeves.
C
C*****
C   Neculai Andrei,
C   Research Institute for Informatics
C   Advanced Modeling and Optimization Laboratory
C*****
C
C       SUBROUTINE HOOKJ(N,X1,X2,Y,Z,S,ITECH)
C
C       DOUBLE PRECISION X1(N),X2(N),Y(N),Z(N),S(N)
C       DOUBLE PRECISION F,FP,FM,BL,FY,F2,H,EPS,A
C
C       CHARACTER*1 FD
C
C       COMMON /PREC/EPS
C       COMMON /TYPE/NO
C       COMMON /WRIT/NX
C       COMMON /L3A/ICOD
C       COMMON /NOF/NBL
C
C       WRITE(NX,150)
150  FORMAT(10X,'Start Optimization process by Hook-Jeeves Method')
C
C       GO TO (151,152,153,154,155)ITECH
151  WRITE(NX,161)
161  FORMAT(10X,'Using Golden Searching Technique')
C       GO TO 170
152  WRITE(NX,162)
162  FORMAT(10X,'Using Fibonacci Searching Technique')
C       GO TO 170
153  WRITE(NX,163)
163  FORMAT(10X,'Using Quadratic Interpolation of Powell')
C       GO TO 170
154  WRITE(NX,164)
164  FORMAT(10X,'Using Dichotomous Searching Technique')
C       GO TO 170
155  WRITE(NX,165)
165  FORMAT(10X,'Using Lambda* = 1.0')
C
170  CONTINUE
110  WRITE(*,111) ' Printing [F=File/D=Display] ? '
111  FORMAT(A,$)
C       READ(5,112,ERR=113) FD
112  FORMAT(A1)
C       IF(FD.NE.'F'.AND.FD.NE.'D') THEN
C       WRITE(NX,114)
114  FORMAT(10X,'Error. Please try again')
C       GO TO 110
C       ENDIF

```

```

        IF(FD.EQ.'F') NO=6
        IF(FD.EQ.'D') NO=6
        GO TO 115
113    WRITE(NX,114)
        GO TO 110
115    CONTINUE
C
        ICOD=0
        JTYPE=0
        H=8.0D-1
        NFUNC=0
        NBL=0
C
        K=1
        CALL FUNC(N,X1,F)
        NFUNC=NFUNC+1
C
        IF(NO.EQ.6) THEN
+      OPEN(6,FILE='PRINT.DAT',ACCESS='SEQUENTIAL',
        FORM='FORMATTED')
        ENDIF
C
        WRITE(NO,50)
50    FORMAT(10X,'*****Optimization Method of HOOK-JEEVES.*****')
        IF(ITECH.EQ.1) WRITE(NO,161)
        IF(ITECH.EQ.2) WRITE(NO,162)
        IF(ITECH.EQ.3) WRITE(NO,163)
        IF(ITECH.EQ.4) WRITE(NO,164)
        IF(ITECH.EQ.5) WRITE(NO,165)
C
        WRITE(NO,51)
51    FORMAT(10X,'Initial Point :')
        WRITE(NO,52)(X1(I),I=1,N)
52    FORMAT(10X,F12.6)
        WRITE(NO,53) H
53    FORMAT(10X,'Step H      = ',F12.6)
        WRITE(NO,54) EPS
54    FORMAT(10X,'Precision EPS = ',g30.15)
        WRITE(NO,55) F
55    FORMAT(10X,'Val. Func. in Initial point = ',F18.6,/)
C
        IF(JTYPE.EQ.1) WRITE(NO,56) K
56    FORMAT(5X,'Iteration K = ',I4)
20    CONTINUE
        IF(JTYPE.EQ.1) WRITE(NO,72)
72    FORMAT(10X,'--- Initialization ---')
        DO 1 J=1,N
1      Y(J)=X1(J)
C
10    DO 5 I=1,N
        IF(JTYPE.EQ.1) WRITE(NO,49) I
49    FORMAT(8X,'I = ',I3)
        DO 2 J=1,N
2      Z(J)=Y(J)
C
        CALL FUNC(N,Z,F)
        NFUNC=NFUNC+1
        IF(JTYPE.EQ.1) WRITE(NO,57)
57    FORMAT(10X,'Bazic Point')
        IF(JTYPE.EQ.1) WRITE(NO,52)(Z(J),J=1,N)

```



```

IF(JTYPE.EQ.1) WRITE(NO,58)F
58     FORMAT(10X,'Function value in the basic point = ',F12.6)
Z(I)=Z(I)+H
C
CALL FUNC(N,Z,FP)
NFUNC=NFUNC+1
C
IF(JTYPE.EQ.1) WRITE(NO,59)
59     FORMAT(10X,'Pozitive Point')
IF(JTYPE.EQ.1) WRITE(NO,52)(Z(J),J=1,N)
IF(JTYPE.EQ.1) WRITE(NO,60)FP
60     FORMAT(10X,'Function value in pozitive point = ',F12.6)
IF(FP.LT.F) GO TO 3
Z(I)=Z(I)-2.*H
C
CALL FUNC(N,Z,FM)
NFUNC=NFUNC+1
C
IF(JTYPE.EQ.1) WRITE(NO,61)
61     FORMAT(10X,'Negative Point')
IF(JTYPE.EQ.1) WRITE(NO,52)(Z(J),J=1,N)
IF(JTYPE.EQ.1) WRITE(NO,62)FM
62     FORMAT(10X,'Function value in negative point = ',F12.6)
IF(FM.LT.F.AND.F.LT.FP) GO TO 3
A=FP
IF(FM.LT.A) A=FM
IF(F.LT.A) GO TO 5
3     DO 4 J=1,N
4     Y(J)=Z(J)
5     CONTINUE
C
C
IS=0
DO 6 I=1,N
IF(X1(I).NE.Y(I)) IS=1
6     CONTINUE
IF(IS.EQ.1) THEN
DO 7 I=1,N
7     X2(I)=Y(I)
GO TO 8
ELSE
IF(H.LT.EPS) GO TO 100
H=H/2.
IF(JTYPE.EQ.1) WRITE(NO,71)H
71     FORMAT(10X,'***** Value of H = ',F12.6)
GO TO 10
ENDIF
C
C
DO 9 I=1,N
9     S(I)=X2(I)-X1(I)
IF(JTYPE.EQ.1) WRITE(NO,63)
63     FORMAT(10X,'Direction S :')
IF(JTYPE.EQ.1) WRITE(NO,52)(S(J),J=1,N)
C
C-----
GO TO (171,172,173,174,175) ITECH
171    CALL L1(N,X2,S,BL)
GO TO 180
172    CALL L2(N,X2,S,BL)

```

```

        IF(EPS.EQ.0.0) RETURN
        GO TO 180
173    CALL L3(N,X2,S,BL)
        IF(ICOD.EQ.1) RETURN
        GO TO 180
174    CALL L4(N,X2,S,BL)
        GO TO 180
175    BL=1.0D0
        IF(EPS.EQ.0.0) EPS=0.0000001
C
180    CONTINUE
C-----
C
        IF(JTYPE.EQ.1) WRITE(NO,64)BL
64      FORMAT(12X,'Steplength =',F12.6)
C
        NFUNC=NFUNC+NBL
C
        DO 11 I=1,N
11      Y(I)=X2(I)+BL*S(I)
        IF(JTYPE.EQ.1)WRITE(NO,65)
65      FORMAT(10X,'New Point: Y = X + lambda*S')
        IF(JTYPE.EQ.1) WRITE(NO,52)(Y(J),J=1,N)

        K=K+1
        IF(K.GT.1000) THEN
        WRITE(NX,201)
201    FORMAT(10X,'Too many iterations.',/,
+      10X,'This function is too complicated for the power',/,
+      10X,'of Hook-Jeeves Optimization Method.',/,
+      10X,'May be this function is a ridge or a valley.',/,
+      10X,'Please consider another initial point, or',/,
+      10X,'try another optimization method.')
        RETURN
        ENDIF
C
        IF(JTYPE.EQ.1) WRITE(NO,66) K
66      FORMAT(5X,'Iteration K = ',I4)
C
        CALL FUNC(N,Y,F)
        NFUNC=NFUNC+1
C
        IF(JTYPE.EQ.1) WRITE(NO,67)F
67      FORMAT(10X,'Function value in the new point = ',F12.6)
        DO 15 I=1,N
        IF(JTYPE.EQ.1) WRITE(NO,68)I
68      FORMAT(8X,'I = ',I3)
        DO 12 J=1,N
12      Z(J)=Y(J)
C
        CALL FUNC(N,Z,F)
        NFUNC=NFUNC+1
C
        IF(JTYPE.EQ.1) WRITE(NO,57)
        IF(JTYPE.EQ.1) WRITE(NO,52)(Z(J),J=1,N)
        IF(JTYPE.EQ.1) WRITE(NO,58)F
        Z(I)=Z(I)+H
C
        CALL FUNC(N,Z,FP)
        NFUNC=NFUNC+1

```

```

C      IF(JTYPE.EQ.1) WRITE(NO,59)
      IF(JTYPE.EQ.1) WRITE(NO,52)(Z(J),J=1,N)
      IF(JTYPE.EQ.1) WRITE(NO,60)FP
      IF(FP.LT.F) GO TO 13
      Z(I)=Z(I)-2.*H

C      CALL FUNC(N,Z,FM)
      NFUNC=NFUNC+1

C      IF(JTYPE.EQ.1) WRITE(NO,61)
      IF(JTYPE.EQ.1) WRITE(NO,52)(Z(J),J=1,N)
      IF(JTYPE.EQ.1) WRITE(NO,62)FM
      IF(FM.LT.F.AND.F.LT.FP) GO TO 13
      A=FP
      IF(FM.LT.A) A=FM
      IF(F.LT.A) GO TO 15
13     DO 14 J=1,N
14     Y(J)=Z(J)
15     CONTINUE
C
C      CALL FUNC(N,X2,F2)
      CALL FUNC(N,Y,FY)
      NFUNC=NFUNC+2

C      IF(JTYPE.EQ.1) WRITE(NO,69)F2
69     FORMAT(12X,'Function value in X =',F12.6)
      IF(JTYPE.EQ.1) WRITE(NO,70)FY
70     FORMAT(12X,'Function value in Y =',F12.6)
C
C      IF(FY.LT.F2) THEN
      DO 16 I=1,N
16     X1(I)=X2(I)
      DO 17 I=1,N
17     X2(I)=Y(I)
C
      IF(DABS(FY-F2).LE.EPS) THEN
      IF(H.LE.EPS) GO TO 100
      H=H/2.
      ENDIF

C      GO TO 8
      ELSE
      IF(H.LE.EPS) GO TO 100
      H=H/2.
      IF(JTYPE.EQ.1) WRITE(NO,71)H
      DO 18 I=1,N
18     X1(I)=X2(I)
      GO TO 20
      ENDIF

C
C      CONTINUE
100
C      WRITE(NO,79)
79     FORMAT(/5X,'-----')
      WRITE(NO,80)K,NFUNC
80     FORMAT(10X,'Iterations = ',I4/,

```

```

      +      10X,'No. Function Evaluations = ',I5)
      WRITE(NO,81)EPS
81      FORMAT(10X,'Precision = ',F12.9)
      WRITE(NO,82)
82      FORMAT(5X,'***** Minimum Point *****')
      WRITE(NO,83)(Z(J),J=1,N)
83      FORMAT(10X,F12.6)
      WRITE(NO,84)F
84      FORMAT(10X,'Val. Func. in Minimum Point =',F12.6)
C
C
      RETURN
      END
C
C***** Last card of HOOKJ.FOR

```

```

C*****
C
C  Rosenbrock's method for unconstrained minimization.
C
C*****
C  Neculai Andrei,
C  Research Institute for Informatics
C  Advanced Modeling and Optimization Laboratory
C*****
C
      SUBROUTINE ROSE(N,NN,X,X1,X2,Z,H,HM,XD,P,Y)
C
      DOUBLE PRECISION X(N),X1(N),X2(N),Z(N),H(N),Y(N)
      DOUBLE PRECISION HM(NN),XD(NN),P(NN)
C
      DOUBLE PRECISION EPS,FX,FY,S,T,ALFA,BETA
C
      CHARACTER*1 FD
C
      COMMON /PREC/EPS
      COMMON /TYPE/NO
      COMMON /WRIT/NX
C
      NFUNC=0
      JTYPE=0
      IR=0
C
      WRITE(NX,150)
150      FORMAT(10X,'Start Optimization Process by Rosenbrock',
      +      ' Method')
C
110      WRITE(*,111)' Printing [F=File/D=Display] ? '
111      FORMAT(A,$)
      READ(5,112,ERR=113) FD
112      FORMAT(A1)
      IF(FD.NE.'F'.AND.FD.NE.'D') THEN
      WRITE(NX,114)
114      FORMAT(10X,'Error. Please try again')
      GO TO 110
      ENDIF
      IF(FD.EQ.'F') NO=6

```

```

        IF(FD.EQ.'D') NO=6
        GO TO 115
113    WRITE(NX,114)
        GO TO 110
115    CONTINUE
C
        IF(NO.EQ.6) THEN
+      OPEN(6,FILE='PRINT.DAT',ACCESS='SEQUENTIAL',
        FORM='FORMATTED')
        ENDIF
C
        ALFA=3.0D0
        BETA=5.0D-1
C
        DO 1 I=1,N
1      H(I)=8.0D-1
C
        ITER=1
C
        CALL FUNC(N,X,FX)
        NFUNC=NFUNC+1
C
        WRITE(NO,50)
50     FORMAT(10X,'***** Optimization Method of Rosenbrock *****')
        WRITE(NO,51)
51     FORMAT(10X,'Initial Point : ')
        WRITE(NO,52)(X(I),I=1,N)
52     FORMAT(10X,F18.6)
        WRITE(NO,53)EPS
53     FORMAT(10X,'Accuracy EPS = ',F18.6)
        WRITE(NO,54)FX
54     FORMAT(10X,'Val. Func. in Initial Point = ',F18.6)
C
        DO 3 I=1,N
        DO 2 J=1,N
        L=J+(I-1)*N
2      XD(L)=0.0
        L=I+(I-1)*N
3      XD(L)=1.0
C
        DO 4 I=1,N
        X1(I)=0.0
        X2(I)=0.0
4      Z(I)=0.0
C
5      CONTINUE
C
        IF(JTYPE.EQ.1) THEN
        WRITE(NO,100) ITER
100    FORMAT(10X,'ITERATION = ',I3)
        WRITE(NO,101)
101    FORMAT(10X,'Current Point: ')
        WRITE(NO,102)(I,X(I),I=1,N)
102    FORMAT(10X,I4,5X,F18.6)
        WRITE(NO,103)FX
103    FORMAT(10X,'Func. Value in current point = ',F18.6)
        ENDIF
C
        I=1
6      DO 7 J=1,N

```

```

      L=J+(I-1)*N
7      Y(J)=X(J)+H(I)*XD(L)
C
      CALL FUNC(N,Y,FY)
      NFUNC=NFUNC+1
C
      IF(DABS(FX-FY).LE.EPS) IR=IR+1
      IF(IR.GT.20) GO TO 30
C
      IF(JTYPE.EQ.1) THEN
        WRITE(NO,104)
104      FORMAT(10X,'Point Y(I)')
        WRITE(NO,102)(I1,Y(I1),I1=1,N)
        WRITE(NO,105)FY
105      FORMAT(10X,'Val func in point Y=',F18.6)
      ENDIF
C
      IF(FY.LT.FX) THEN
C
C-----Succes.
C
      IF(JTYPE.EQ.1) WRITE(NO,106)I
106      FORMAT(10X,'I=',I3,' Success')
C
      Z(I)=Z(I)+H(I)
      H(I)=ALFA*H(I)
      X1(I)=1.0
      FX=FY
      DO 8 J=1,N
8      X(J)=Y(J)
      ITER=ITER+1
      IF(ITER.GT.1001) THEN
        WRITE(NX,40)
40      FORMAT(10X,'Too many iterations.',/,
+      10X,'This function is too complicated for the',/,
+      10X,'power of the Rosenbrock Optimization Method.',/,
+      10X,'May be this function is a ridge or a valley.',/,
+      10X,'Please consider another initial point, or',/,
+      10X,'try another optimization method')
        RETURN
      ENDIF
      GO TO 9
C
      ELSE
C
C-----Failure.
C
      IF(JTYPE.EQ.1) WRITE(NO,107) I
107      FORMAT(10X,'I=',I3,' Failure')
C
      H(I)=-BETA*H(I)
      X2(I)=1.0
      GO TO 9
C
      ENDIF
C
9      I=I+1
      IF(I.LE.N) GO TO 6
C
      IF(JTYPE.EQ.1) THEN

```

```

        WRITE(NO,108)
108      FORMAT(10X,'Arrays X1 and X2')
        WRITE(NO,109)(I1,X1(I1),X2(I1),I1=1,N)
109      FORMAT(10X,I2,3X,F5.2,3X,F5.2)
      ENDIF
C
      IS=1
      DO 10 J=1,N
      IF(X1(J)*X2(J).NE.0.0) GO TO 10
      IS=0
      GO TO 11
10      CONTINUE
11      IF(IS.EQ.0) GO TO 5
C
      IF(JTYPE.EQ.1) THEN
        WRITE(NO,210)
210      FORMAT(10X,'Array Z(I)')
        WRITE(NO,102)(I1,Z(I1),I1=1,N)
      ENDIF
C
      IS=1
      DO 12 J=1,N
      IF(DABS(Z(J)).LE.EPS) GO TO 12
      IS=0
      GO TO 13
12      CONTINUE
13      IF(IS.EQ.1) GO TO 30
C
      DO 15 I=1,N
      DO 14 J=1,N
      L=J+(I-1)*N
14      HM(L)=Z(J)
      Z(I)=0.0
15      CONTINUE
C
      IF(JTYPE.EQ.1) THEN
        WRITE(NO,211)
211      FORMAT(10X,'Array HM:')
        WRITE(NO,102)(I1,HM(I1),I1=1,NN)
C
        WRITE(NO,213)
213      FORMAT(10X,'Array XD: ')
        WRITE(NO,102)(I1,XD(I1),I1=1,NN)
      ENDIF
C
      DO 16 I=1,N
      DO 16 J=1,N
      L=I+(J-1)*N
      P(L)=0.0
      DO 16 K=1,N
      L1=I+(K-1)*N
      L2=K+(J-1)*N
16      P(L)=P(L)+XD(L1)*HM(L2)
C
      IF(JTYPE.EQ.1) THEN
        WRITE(NO,212)
212      FORMAT(10X,'Array P:')
        WRITE(NO,102)(I1,P(I1),I1=1,NN)
      ENDIF
C

```

```

        S=0.0
        DO 17 J=1,N
17      S=S+P(J)*P(J)
        T=SQRT(S)
        DO 18 J=1,N
18      XD(J)=P(J)/T
C
        DO 25 I=2,N
        DO 19 K=1,N
        L=K+(I-1)*N
19      Y(K)=P(L)
C
        IF(JTYPE.EQ.1) THEN
            WRITE(NO,220)
220      FORMAT(15X,'Array Y: cycle 19')
            WRITE(NO,102)(I1,Y(I1),I1=1,N)
        ENDIF
C
        DO 22 J=1,I-1
        S=0.0
        DO 20 K=1,N
        L1=K+(I-1)*N
        L2=K+(J-1)*N
20      S=S+P(L1)*XD(L2)
C
        IF(JTYPE.EQ.1) WRITE(NO,221)S
221      FORMAT(10X,'S = (cycle 20)',F18.6)
C
        DO 21 K=1,N
        L=K+(J-1)*N
21      Y(K)=Y(K)-S*XD(L)
C
        IF(JTYPE.EQ.1) THEN
            WRITE(NO,222)
222      FORMAT(10X,'Array Y: cycle 21')
            WRITE(NO,102)(I1,Y(I1),I1=1,N)
        ENDIF
C
22      CONTINUE
C
        S=0.0
        DO 23 K=1,N
23      S=S+Y(K)*Y(K)
C
        IF(JTYPE.EQ.1) WRITE(NO,223) S
223      FORMAT(10X,'S = (cycle 23)',F18.6)
C
        T=SQRT(S)
        DO 24 K=1,N
        L=K+(I-1)*N
24      XD(L)=Y(K)/T
25      CONTINUE
C
        IF(JTYPE.EQ.1) THEN
            WRITE(NO,216)
216      FORMAT(10X,'Array XD - New directions')
            WRITE(NO,102)(I1,XD(I1),I1=1,NN)
        ENDIF
C
        DO 26 I=1,N

```



```

      X1(I)=0.0D0
      X2(I)=0.0D0
      Z(I)=0.0D0
26      H(I)=8.0D-1
      C
      GO TO 5
      C
30      CONTINUE
      C
      WRITE(NO,31)
31      FORMAT(/5X,'-----')
      WRITE(NO,32)ITER,NFUNC
32      FORMAT(10X,'Iterations = ',I4/,
+      10X,'No. Function Evaluations = ',I5)
      WRITE(NO,33)EPS
33      FORMAT(10X,'Accuracy = ',F12.9)
      WRITE(NO,34)
34      FORMAT(5X,'***** Minimum Point *****')
      WRITE(NO,35)(X(J),J=1,N)
35      FORMAT(10X,F18.6)
      WRITE(NO,36)FX
36      FORMAT(10X,'Val. Func. in Minimum Point =',F18.6)
      C
      RETURN
      END
      C
C*****Last card of ROSE.FOR

```

```

C*****
C
C  Nelder - Mead Optimization method.
C  Simplex Method.
C
C*****
C  Neculai Andrei,
C  Research Institute for Informatics
C  Advanced Modeling and Optimization Laboratory
C*****
C
      SUBROUTINE NELMED(N,N1,NN,X,X0,XR,XE,XD,VF)
      C
      DOUBLE PRECISION X(N),X0(N),XR(N),XE(N)
      DOUBLE PRECISION XD(NN),VF(N1),VF1(100)
      C
      DOUBLE PRECISION ALFA,BETA,GAMA,EPS
      DOUBLE PRECISION FX,F0,FR,FE,VFI,VFM,S,T
      C
      CHARACTER*1 FD
      C
      COMMON /PREC/EPS
      COMMON /TYPE/NO
      COMMON /WRIT/NX
      C
      NFUNC=0
      ITER=0
      ALFA=1.0
      BETA=0.5

```

```

        GAMA=2.0
        JTYPE=0
C
        WRITE(NX,150)
150      FORMAT(10X,'Start Optimization Process by Nelder-Mead',
+        ' Method')
C
110      WRITE(*,111)' Printing [F=File/D=Display] ? '
111      FORMAT(A,$)
        READ(5,112,ERR=113) FD
112      FORMAT(A1)
        IF(FD.NE.'F'.AND.FD.NE.'D') THEN
            WRITE(NX,114)
114      FORMAT(10X,'Error. Please try again')
            GO TO 110
        ENDIF
        IF(FD.EQ.'F') NO=6
        IF(FD.EQ.'D') NO=6
        GO TO 115
113      WRITE(NX,114)
        GO TO 110
115      CONTINUE
C
        IF(NO.EQ.6) THEN
            OPEN(6,FILE='PRINT.DAT',ACCESS='SEQUENTIAL',
+            FORM='FORMATTED')
            ENDIF
C
        CALL FUNC(N,X,FX)
        NFUNC=NFUNC+1
C
        WRITE(NO,50)
50      FORMAT(10X,'***** Optimization Method of Nelder-Mead *****')
        WRITE(NO,51)
51      FORMAT(10X,'Initial Point : ')
        WRITE(NO,52)(X(I),I=1,N)
52      FORMAT(10X,F18.6)
        WRITE(NO,53)EPS
53      FORMAT(10X,'Accuracy EPS = ',g30.15)
        WRITE(NO,54)FX
54      FORMAT(10X,'Val. Func. in Initial Point = ',F18.6)
C
C----- Build up the initial simplex
C      using the initial point X.
C
        N2=N*N
        DO 1 J=1,N
            L=J+N2
1      XD(L)=X(J)
C
        DO 3 I=1,N
            DO 2 J=1,N
                L=J+(I-1)*N
2      XD(L)=X(J)
                L=I+(I-1)*N
3      XD(L)=XD(L)+0.5
C
        IF(JTYPE.EQ.1) THEN
            WRITE(NO,201)
201      FORMAT(10X,'Initial simplex')

```

```

        WRITE(NO,202)(I1,XD(I1),I1=1,NN)
202    FORMAT(10X,I4,5X,F18.6)
    ENDIF
C
C-----Compute the values of the objective
C      function in vertices of the simplex
C
        DO 5 I=1,N1
        DO 4 J=1,N
        L=J+(I-1)*N
4      X(J)=XD(L)
        CALL FUNC(N,X,FX)
        NFUNC=NFUNC+1
        VF(I)=FX
5      CONTINUE
C
    IF(JTYPE.EQ.1) THEN
        WRITE(NO,203)
203    FORMAT(10X,'Val. Func in points of initial simplex')
        WRITE(NO,202)(I1,VF(I1),I1=1,N1)
    ENDIF
C
6      ITER=ITER+1
C
        IF(ITER.GT.1001)THEN
        IF(EPS.EQ.0.0) THEN
        WRITE(NX,41)
41    FORMAT(10X,'Too many iterations.',/
+      10X,'The accuracy is too high, EPS=0.0',//,
+      10X,'Please consider another finite, small',/
+      10X,'positive value for accuracy EPS.')
        RETURN
        ENDIF
C
        WRITE(NX,40)
40    FORMAT(10X,'Too many iterations.',/
+      10X,'This function is too complicated for the',/
+      10X,'power of the Nelder-Mead Optimization Method.',/
+      10X,'May be this function is a ridge or a valley.',//,
+      10X,'Please consider another initial point, or',/
+      10X,'try another optimization method')
        RETURN
        ENDIF
C
    IF(JTYPE.EQ.1) THEN
        WRITE(NO,204)ITER
204    FORMAT(5X,'ITERATION = ',I4)
        write(no,220)
220    format(3x,'Current simplex')
        WRITE(NO,202)(I1,XD(I1),I1=1,NN)
        WRITE(NO,223)
223    FORMAT(10X,'Func. value in points of the current simplex')
    ENDIF
*
    do i=1,n1
    do j=1,n
        l=j+(i-1)*n
        x(j)=xd(l)
    end do
    call func(n,x,fx)

```

```

        vf1(i)=fx
    end do
C
    IF(JTYPE.EQ.1) WRITE(NO,202)(I1,VF1(I1),I1=1,N1)
C
    VFI=VF1(1)
    VFM=VFI
    IMI=1
    IMA=1
    DO 7 I=2,N1
    IF(VF1(I).LT.VFI) THEN
        VFI=VF1(I)
        IMI=I
    ENDIF
    IF(VF1(I).GT.VFM) THEN
        VFM=VF1(I)
        IMA=I
    ENDIF
7    CONTINUE
C
    IF(JTYPE.EQ.1) THEN
        WRITE(NO,205)IMI,VFI
205    FORMAT(10X,'IMI = ',I4,4X,'VFI = ',F18.6)
        WRITE(NO,206)IMA,VFM
206    FORMAT(10X,'IMA = ',I4,4X,'VFM = ',F18.6)
    ENDIF
C
C
C-----Compute the centroid of the simplex.
C
    DO 9 J=1,N
    X0(J)=0.0
    DO 8 I=1,N1
    IF(I.EQ.IMA) GO TO 8
    L=J+(I-1)*N
    X0(J)=X0(J)+XD(L)
8    CONTINUE
9    CONTINUE
C
    DO 10 J=1,N
10    X0(J)=X0(J)/FLOAT(N)
    CALL FUNC(N,X0,F0)
    NFUNC=NFUNC+1
C
    IF(JTYPE.EQ.1) THEN
        WRITE(NO,208)
208    FORMAT(10X,'Centroid Point')
        WRITE(NO,202)(I1,X0(I1),I1=1,N)
        WRITE(NO,211)F0
211    FORMAT(10X,'Val. Func. in Centr. Point = ',F18.6)
    ENDIF
C
C-----Compute the reflection point.
C
    DO 11 J=1,N
    L=J+(IMA-1)*N
11    XR(J)=(1.+ALFA)*X0(J)-ALFA*XD(L)
    CALL FUNC(N,XR,FR)
    NFUNC=NFUNC+1
C

```

```

        IF(JTYPE.EQ.1) THEN
            WRITE(NO,209)
209      FORMAT(10X,'Reflection Point')
            WRITE(NO,202)(I1,XR(I1),I1=1,N)
            WRITE(NO,210) FR
210      FORMAT(10X,'Func. Val. in Ref. Point = ',F18.6)
        ENDIF
C
C-----Decision on the way.
C
        IF(FR.LT.VFI) THEN
C
            DO 12 J=1,N
12      XE(J)=GAMA*XR(J)+(1.-GAMA)*X0(J)
            CALL FUNC(N,XE,FE)
            NFUNC=NFUNC+1
C
            IF(JTYPE.EQ.1) THEN
                WRITE(NO,212)
212      FORMAT(10X,'FR.LT.VFI -- Expansion point')
                WRITE(NO,202)(I1,XE(I1),I1=1,N)
                WRITE(NO,213)FE
213      FORMAT(10X,'Func. Val. in Expan. Point = ',F18.6)
            ENDIF
C
            IF(FE.LT.VFI) THEN
                DO 13 J=1,N
                L=J+(IMA-1)*N
13      XD(L)=XE(J)
                VF1(IMA)=FE
                GO TO 23
            ELSE
                DO 16 J=1,N
                L=J+(IMA-1)*N
16      XD(L)=XR(J)
                VF1(IMA)=FR
                GO TO 23
            ENDIF
C
            ELSE
C
                IS=1
                DO 14 I=1,N1
                IF(I.EQ.IMA) GO TO 14
                IF(FR.GT.VF1(I)) GO TO 14
                IS=0
                GO TO 15
14      CONTINUE
15      IF(IS.EQ.1) GO TO 18
                DO 17 J=1,N
                L=J+(IMA-1)*N
17      XD(L)=XR(J)
                VF1(IMA)=FR
                GO TO 23
C
                ENDIF
C
C-----Compute the contraction point.
C
18      CONTINUE

```

```

        IF(FR.GT.VFM) THEN
        DO 19 J=1,N
        L=J+(IMA-1)*N
19      X(J)=BETA*XD(L)+(1.-BETA)*X0(J)
        ELSE
        DO 20 J=1,N
20      X(J)=BETA*XR(J)+(1.-BETA)*X0(J)
        ENDIF
C
        CALL FUNC(N,X,FX)
        NFUNC=NFUNC+1
C
        IF(JTYPE.EQ.1) THEN
        WRITE(NO,214)
214      FORMAT(10X,'Contraction Point')
        WRITE(NO,202)(I1,X(I1),I1=1,N)
        WRITE(NO,215)FX
215      FORMAT(10X,'Func. Val. in Contr. Point = ',F18.6)
        ENDIF
C
C-----Reduce the simplex.
C
        IF(FX.GT.VFM) THEN
        DO 21 I=1,N1
        DO 21 J=1,N
        L=J+(I-1)*N
        K=J+(IMI-1)*N
21      XD(L)=(XD(L)+XD(K))/2.
C
        IF(JTYPE.EQ.1) THEN
        WRITE(NO,216)
216      FORMAT(10X,'New Reduced Simplex ')
        WRITE(NO,202)(I1,XD(I1),I1=1,NN)
        ENDIF
C
        GO TO 23
C
        ELSE
C
        DO 22 J=1,N
        L=J+(IMA-1)*N
22      XD(L)=X(J)
        GO TO 23
C
        ENDIF
C
C-----Test of continuation (accuracy).
C
23      CONTINUE
        S=0.0
        DO 24 I=1,N1
24      S=S+(VF1(I)-F0)**2
        S=S/FLOAT(N+1)
        T=SQRT(S)
C
        IF(JTYPE.EQ.1) WRITE(NO,218)T
218      FORMAT(10X,'*** Accuracy test T = ',F18.6)
C
        IF(T.LE.EPS) GO TO 30
        GO TO 6

```

```

C
30    CONTINUE
      DO 31 J=1,N
      L=J+(IMI-1)*N
31    X(J)=XD(L)
C
      WRITE(NO,32)
32    FORMAT(/5X,'-----')
      WRITE(NO,33)ITER,NFUNC
33    FORMAT(10X,'Iterations = ',I4/,
+      10X,'No. Function Evaluations = ',I5)
      WRITE(NO,34)EPS
34    FORMAT(10X,'Accuracy = ',F12.9)
      WRITE(NO,35)
35    FORMAT(5X,'***** Minimum Point *****')
      WRITE(NO,36)(X(J),J=1,N)
36    FORMAT(10X,F18.6)
      WRITE(NO,37) VFI
37    FORMAT(10X,'Val. Func. in Minimum Point =',F18.6)
C
      RETURN
      END
C
C*****Last card of NELMED.FOR

```

```

C*****
C
C  Powell's Method.
C
C*****
C  Neculai Andrei,
C  Research Institute for Informatics
C  Advanced Modeling and Optimization Laboratory
C*****
C
      SUBROUTINE POWEL(N,NN,X,X1,X2,X3,S,XD,ITECH)
C
      DOUBLE PRECISION X(N),X1(N),X2(N),X3(N),S(N),XD(NN)
      DOUBLE PRECISION BL,EP,EPS,F1,F2,F3,FS,T,V,DELTA,DEL
      DOUBLE PRECISION FA,FB
C
      CHARACTER*1 FD
C
      COMMON /PREC/EPS
      COMMON /TYPE/NO
      COMMON /WRIT/NX
      COMMON /L3A/ICOD
      COMMON /NOF/NBL
C
      WRITE(NX,150)
150    FORMAT(10X,'Start Optimization Process by Powell Method')
C
      GO TO(151,152,153,154) ITECH
151    WRITE(NX,161)
161    FORMAT(10X,'Using Golden Searching Technique')
      GO TO 170
152    WRITE(NX,162)

```

```

162  FORMAT(10X,'Using Fibonacci Searching Technique')
      GO TO 170
153  WRITE(NX,163)
163  FORMAT(10X,'Using Quadratic Interpolation of Powell')
      GO TO 170
154  WRITE(NX,164)
164  FORMAT(10X,'Using Dichotomous Searching Technique')
C
170  CONTINUE
110  WRITE(*,111)' Printing [F=File/D=Display] ? '
111  FORMAT(A,$)
      READ(5,112,ERR=113) FD
112  FORMAT(A1)
      IF(FD.NE.'F'.AND.FD.NE.'D') THEN
          WRITE(NX,114)
114  FORMAT(10X,'Error. Please try again')
          GO TO 110
      ENDIF
      IF(FD.EQ.'F') NO=6
      IF(FD.EQ.'D') NO=6
      GO TO 115
113  WRITE(NX,114)
      GO TO 110
115  CONTINUE
C
      IF(NO.EQ.6) THEN
          OPEN(6,FILE='PRINT.DAT',ACCESS='SEQUENTIAL',
+         FORM='FORMATTED')
      ENDIF
C
      NFUNC=0
      NBL=0
      JTYPE=0
      EP=1.0D-3
C
      ITER=0
      CALL FUNC(N,X,F1)
      NFUNC=NFUNC+1
C
      WRITE(NO,50)
50  FORMAT(10X,'***** Optimization Method of Powell *****')
      IF(ITECH.EQ.1) WRITE(NO,161)
      IF(ITECH.EQ.2) WRITE(NO,162)
      IF(ITECH.EQ.3) WRITE(NO,163)
      IF(ITECH.EQ.4) WRITE(NO,164)
C
      WRITE(NO,51)
51  FORMAT(10X,'Initial Point : ')
      WRITE(NO,52)(X(I),I=1,N)
52  FORMAT(10X,F18.6)
      WRITE(NO,53)EPS
53  FORMAT(10X,'Accuracy EPS = ',g30.15)
      WRITE(NO,54)F1
54  FORMAT(10X,'Function Value in Initial Point = ',F18.6)
C-----
C
      DO 3 I=1,N
      DO 2 J=1,N
          L=J+(I-1)*N
2      XD(L)=0.0

```



```

        L=I+(I-1)*N
3      XD(L)=1.0
C
4      CONTINUE
      ITER=ITER+1
C
      IF(ITER.GT.1000) THEN
        WRITE(NX,56)
56      FORMAT(10X,'Too many iterations.',/,
+        10X,'This function is too complicated for the power',/,
+        10X,'of the Powell Optimization Method.',/,
+        10X,'May be this function is a ridge or a valley.',/,
+        10X,'Please consider another initial point, or',/,
+        10X,'try another optimization method.')
        RETURN
      ENDIF
C
      IF(JTYPE.EQ.1) WRITE(NO,300) ITER
300     FORMAT(10X,'ITER = ',I4)
C
      if(jtype.eq.1) then
        write(no,360)
360     format(4x,'The Current Point:')
        WRITE(NO,302)(I1,X(I1),I1=1,N)
*
        WRITE(NO,301)
301     FORMAT(10X,'Directions:')
        WRITE(NO,302)(I1,XD(I1),I1=1,NN)
302     FORMAT(10X,I4,4X,F18.6)
      endif
C
      I=1
      DO 5 J=1,N
5      S(J)=XD(J)
C-----
      DO 25 J=1,N
25     X1(J)=X(J)+EP*S(J)
C
      CALL FUNC(N,X1,FA)
      NFUNC=NFUNC+1
C-----
      DO 26 J=1,N
26     X1(J)=X(J)-EP*S(J)
C
      CALL FUNC(N,X1,FB)
      NFUNC=NFUNC+1
C
      IF(FA.GT.FB) THEN
        DO 27 J=1,N
27     S(J)=-S(J)
        ENDIF
C
C-----
      GO TO(171,172,173,174) ITECH
171    CALL L1(N,X,S,BL)
      GO TO 180
172    CALL L2(N,X,S,BL)
      GO TO 180
173    CALL L3(N,X,S,BL)
      IF(ICOD.EQ.1) RETURN

```

```

        GO TO 180
174    CALL L4(N,X,S,BL)
C
180    CONTINUE
C-----
C
        NFUNC=NFUNC+NBL
C
        DO 6 J=1,N
6      X2(J)=X(J)+BL*S(J)
C
        CALL FUNC(N,X2,F2)
        NFUNC=NFUNC+1
C
        IF(JTYPE.EQ.1) WRITE(NO,304)I,BL
        IF(JTYPE.EQ.1) WRITE(NO,302)(I1,X2(I1),I1=1,N)
C
        DELTA=(F1-F2)
        M=1
C
7      I=I+1
        IF(I.GT.N) GO TO 11
C
        DO 8 J=1,N
        L=J+(I-1)*N
8      S(J)=XD(L)
C
        DO 30 J=1,N
30     X1(J)=X(J)+EP*S(J)
C
        CALL FUNC(N,X1,FA)
        NFUNC=NFUNC+1
C
        DO 31 J=1,N
31     X1(J)=X(J)-EP*S(J)
C
        CALL FUNC(N,X1,FB)
        NFUNC=NFUNC+1
C
        IF(FA.GT.FB) THEN
        DO 32 J=1,N
32     S(J)=-S(J)
        ENDIF
C
C-----
        GO TO(181,182,183,184) ITECH
181    CALL L1(N,X2,S,BL)
        GO TO 190
182    CALL L2(N,X2,S,BL)
        GO TO 190
183    CALL L3(N,X2,S,BL)
        IF(ICOD.EQ.1) RETURN
        GO TO 190
184    CALL L4(N,X2,S,BL)
C
190    CONTINUE
C-----
C
        NFUNC=NFUNC+NBL
C

```

```

DO 9 J=1,N
9   X3(J)=X2(J)+BL*S(J)
C
CALL FUNC(N,X3,F3)
NFUNC=NFUNC+1
C
IF(JTYPE.EQ.1) WRITE(NO,304)I,BL
304  FORMAT(10X,'I = ',I2,4X,'BL=',F18.6,' New Point')
IF(JTYPE.EQ.1) WRITE(NO,302)(I1,X3(I1),I1=1,N)
C
DEL=(F2-F3)
IF(DEL.GT.DELTA) THEN
DELTA=DEL
M=I
ENDIF
C
IF(JTYPE.EQ.1) WRITE(NO,305)M,DELTA
305  FORMAT(10X,'M = ',I3,4X,'DELTA =',F18.8)
C
DO 10 J=1,N
10   X2(J)=X3(J)
F2=F3
GO TO 7
C
11   CONTINUE
C
IF(JTYPE.EQ.1) THEN
WRITE(NO,313)
313  FORMAT(10X,'End of a cycle of N Searchings')
WRITE(NO,314)
314  FORMAT(10X,'New Directions:')
WRITE(NO,302)(I1,XD(I1),I1=1,NN)
ENDIF
C
IS=0
DO 12 I=1,N
IF(DABS(X3(I)-X(I)).LE.EPS) GO TO 12
IS=1
C
IF(JTYPE.EQ.1) WRITE(NO,307)X3(I)-X(I)
307  FORMAT(2X,'X3(I)-X(I) =',F18.8)
C
GO TO 121
12   CONTINUE
C
121  CONTINUE
C
IF(IS.EQ.0) GO TO 100
C
DO 13 I=1,N
13   S(I)=2.*X3(I)-X(I)
C
CALL FUNC(N,S,FS)
NFUNC=NFUNC+1
C
IF(JTYPE.EQ.1) THEN
WRITE(NO,320)
320  FORMAT(/,10X,'Point : 2.*X3(I)-X(I)')
WRITE(NO,302)(I1,S(I1),I1=1,N)
WRITE(NO,321)FS

```

```

321     FORMAT(/,10X,'Function Value in 2*X3(I)-X(I)=' ,F18.6)
      ENDIF
C
      IF(FS.GE.F1) THEN
      DO 14 I=1,N
14      X(I)=X3(I)
          F1=F3
          GO TO 4
      ENDIF
C
      T=(F1-2.*F3+FS)*(F1-F3-DELTA)**2
      V=0.5*DELTA*(F1-FS)**2
C
      if(jtype.eq.1) then
          write(no,3221)f1,f3,delta
3221      format(10x,'f1  =' ,f18.6/,
1          10x,'f2  =' ,f18.6/,
1          10x,'delta=' ,f18.6)
          WRITE(NO,322)T,V
322      FORMAT(/,10X,'T=(f1-2f2+f3)*(f1-f2-delta)**2 = ',F18.6/,
+          10X,'V=0.5*(f1-f3)**2      = ',F18.6)
      endif
C
      IF(T.GE.V) THEN
          if(jtype.eq.1) write(no,355)
355      format(/,4x,'Here T > V: Consider the old directions')
C-----
          DO 15 I=1,N
15      X(I)=X3(I)
          F1=F3
          GO TO 4
      ENDIF
C
      DO 16 I=1,N
16      S(I)=X3(I)-X(I)
C
      if(jtype.eq.1) write(no,350)
350      format(4x,'Here T < V: Compute the direction X3-X')
C-----
          GO TO(191,192,193,194) ITECH
191      CALL L1(N,X3,S,BL)
          GO TO 200
192      CALL L2(N,X3,S,BL)
          GO TO 200
193      CALL L3(N,X3,S,BL)
          IF(ICOD.EQ.1) RETURN
          GO TO 200
194      CALL L4(N,X3,S,BL)
C
200      CONTINUE
C-----
C
      NFUNC=NFUNC+NBL
C
      IF(JTYPE.EQ.1) THEN
          WRITE(NO,323)
323      FORMAT(10X,'Direction : X3(I)-X(I)')
          WRITE(NO,302)(I1,S(I1),I1=1,N)
          WRITE(NO,324)BL
324      FORMAT(10X,'----- BL in Direction X3-X = ',F18.6)

```

```

ENDIF
C
DO 17 I=1,N
17 X(I)=X3(I)+BL*S(I)
C
CALL FUNC(N,X,F1)
NFUNC=NFUNC+1
C
IF(JTYPE.EQ.1) THEN
WRITE(NO,325)
325 FORMAT(/,10X,'New point X=X3+BL*(X3-X)')
WRITE(NO,302)(I1,X(I1),I1=1,N)
WRITE(NO,326)F1
326 FORMAT(/,10X,'Function Value in X = ',F18.6)
ENDIF
C
IS=0
DO 18 I=1,N
IF(DABS(BL*S(I)).LE.EPS) GO TO 18
IS=1
C
IF(JTYPE.EQ.1) WRITE(NO,308)BL*S(I)
308 FORMAT(2X,'BL*S(I) =',F18.8)
C
GO TO 118
18 CONTINUE
C
118 CONTINUE
C
IF(IS.EQ.0) GO TO 100
C
IF(M.EQ.N) THEN
DO 19 J=1,N
L=J+(N-1)*N
19 XD(L)=S(J)
GO TO 4
ENDIF
C
DO 20 I=M+1,N
DO 20 J=1,N
J1=J+(I-2)*N
J2=J+(I-1)*N
20 XD(J1)=XD(J2)
C
DO 21 J=1,N
L=J+(N-1)*N
21 XD(L)=S(J)
GO TO 4
C
100 CONTINUE
C
WRITE(NO,79)
79 FORMAT(/5X,'-----')
WRITE(NO,80)ITER,NFUNC
80 FORMAT(10X,'Iterations = ',I4,/,
+ 10X,'No. of Function Evaluations = ',I5)
WRITE(NO,81) EPS
81 FORMAT(10X,'Accuracy = ',F12.9)
WRITE(NO,82)
82 FORMAT(5X,'***** Minimum Point *****')

```

```

      WRITE(NO,83)(X(I),I=1,N)
83    FORMAT(10X,F18.6)
      WRITE(NO,84) F1
84    FORMAT(10X,'Function Value in Minimum Point =',F18.6)
C
C
      RETURN
      END
C
C*****Last card of POWEL.FOR

```

```

C*****
C
C  Subroutine for one dimensional searching.
C  Golden section with interval searching.
C  =====
C*****
C  Neculai Andrei,
C  Research Institute for Informatics
C  Advanced Modeling and Optimization Laboratory
C*****
C
      SUBROUTINE L1(N,X,S,BL)
      DOUBLE PRECISION X(N),S(N),BL
      DOUBLE PRECISION F0,FY,FY1,SL,SL1,G1,G2,G2P,D1,D2
      DOUBLE PRECISION R,R1,EPS,EPS1,Y(40)

C
      COMMON /PREC/EPS
      COMMON /TYPE/NO
      COMMON /NOF/NBL

C
      EPS1=0.001

C
      R=0.618034
      R1=1.0-R
      SL=1.
      NBL=0
      JTYPE=0

C
      IF(NO.EQ.6)THEN
+      OPEN(6,FILE='PRINT.DAT',ACCESS='SEQUENTIAL',
      FORM='FORMATTED')
      ENDIF

C
      CALL FUNC(N,X,F0)
      NBL=NBL+1

C
      IF(JTYPE.EQ.1) THEN
      WRITE(NO,20)
20    FORMAT(10X,'Basic Point')
      WRITE(NO,21)(I,X(I),I=1,N)
21    FORMAT(10X,I3,5X,F12.6)
      WRITE(NO,22)F0
22    FORMAT(10X,'Function Value in Basic Point = ',F12.6)
      ENDIF
C
      DO 1 I=1,N

```

```

1      Y(I)=X(I)+SL*S(I)
      CALL FUNC(N,Y,FY)
      NBL=NBL+1
C
      IF(JTYPE.EQ.1) THEN
        WRITE(NO,23)
23      FORMAT(10X,'Point: X+SL*S')
        WRITE(NO,21)(I,Y(I),I=1,N)
        WRITE(NO,24)FY
24      FORMAT(10X,'Function value in X+SL*S = ',F12.6)
      ENDIF
C
      SL1=SL-EPS1
      DO 2 I=1,N
2      Y(I)=X(I)+SL1*S(I)
      CALL FUNC(N,Y,FY1)
      NBL=NBL+1
C
      IF(JTYPE.EQ.1) THEN
        WRITE(NO,25)
25      FORMAT(10X,'Point: X+(SL-EPS1)*S')
        WRITE(NO,21)(I,Y(I),I=1,N)
        WRITE(NO,26)FY1
26      FORMAT(10X,'Function value in X+(SL-EPS1)*S = ',F12.6)
      ENDIF
C
      IF(FY.LT.FY1) GO TO 6
C
      D1=0.0
      D2=SL
      GO TO 7
C
6      K=1
      G1=SL
5      G2=G1+SL
C
15     CONTINUE
C
      IF(JTYPE.EQ.1) THEN
        WRITE(NO,27)K
27      FORMAT(5X,'K = ',I3)
        WRITE(NO,28)G1,G2
28      FORMAT(5X,'G1 = ',F12.6,5X,'G2 = ',F12.6)
      ENDIF
C
      DO 3 I=1,N
3      Y(I)=X(I)+G2*S(I)
      CALL FUNC(N,Y,FY)
      NBL=NBL+1
C
      IF(JTYPE.EQ.1) THEN
        WRITE(NO,29)
29      FORMAT(5X,'Point: X+G2*S')
        WRITE(NO,21)(I,Y(I),I=1,N)
        WRITE(NO,30)FY
30      FORMAT(5X,'Function value in X+G2*S = ',F12.6)
      ENDIF
C
      G2P=G2-EPS1
      DO 4 I=1,N

```

```

4      Y(I)=X(I)+G2P*S(I)
      CALL FUNC(N,Y,FY1)
      NBL=NBL+1
C
      IF(JTYPE.EQ.1) THEN
        WRITE(NO,31)
31      FORMAT(5X,'Point: X+(G2-EPS1)*S')
        WRITE(NO,21)(I,Y(I),I=1,N)
        WRITE(NO,32)FY1
32      FORMAT(5X,'Function value in X+(G2-EPS1)*S = ',F12.6)
      ENDIF
C
      IF(FY.LE.FY1) THEN
        IF(K.GT.10) THEN
          K=1
          EPS1=-EPS1
          G1=0.
          G2=SL
          GO TO 15
        ENDIF
        K=K+1
        G1=G2
        GO TO 5
      ELSE
        D1=G1
        D2=G2
        GO TO 7
      ENDIF
C
7      SL=D2-D1
C
      IF(JTYPE.EQ.1) WRITE(NO,33)SL
33      FORMAT(5X,'----- SL = ',F12.6)
C
      IF(SL.LE.EPS) GO TO 10
C
      G1=D1+R1*SL
      G2=D1+R*SL
C
      DO 8 I=1,N
8      Y(I)=X(I)+G1*S(I)
      CALL FUNC(N,Y,FY)
      NBL=NBL+1
C
      IF(JTYPE.EQ.1) THEN
        WRITE(NO,34)
34      FORMAT(10X,'Point: X+G1*S')
        WRITE(NO,21)(I,Y(I),I=1,N)
        WRITE(NO,35)FY
35      FORMAT(10X,'Function value in X+G1*S = ',F12.6)
      ENDIF
C
      DO 9 I=1,N
9      Y(I)=X(I)+G2*S(I)
      CALL FUNC(N,Y,FY1)
      NBL=NBL+1
C
      IF(JTYPE.EQ.1) THEN
        WRITE(NO,36)
36      FORMAT(10X,'Point: X+G2*S')

```



```

        WRITE(NO,21)(I,Y(I),I=1,N)
        WRITE(NO,37)FY1
37      FORMAT(10X,'Function value in X+G2*S = ',F12.6)
      ENDIF
      C
        IF(FY.LT.FY1) THEN
          D2=G2
        ELSE
          D1=G1
        ENDIF
      C
        IF(JTYPE.EQ.1) WRITE(NO,38)D1,D2
38      FORMAT(10X,'D1 = ',F12.6,5X,'D2 = ',F12.6)
        GO TO 7
      C
10      BL=(D1+D2)/2
      C
        IF(JTYPE.EQ.1) THEN
          WRITE(NO,39)BL
39      FORMAT(20X,'Minimum Point = ',F12.6)
          WRITE(NO,40)NBL
40      FORMAT(20X,'Number of Function Evaluation =',I5)
        ENDIF
      C
        RETURN
        END
      C
C***** Last card of L1.FOR

```

```

C*****
C
C  Subroutine for one dimensional searching.
C  Fibonacci searching technique.
C  =====
C*****
C  Neculai Andrei,
C  Research Institute for Informatics
C  Advanced Modeling and Optimization Laboratory
C*****
C
      SUBROUTINE L2(N,X,S,BL)
      DOUBLE PRECISION X(N),S(N),BL
      DOUBLE PRECISION X1(50),FIB(50)
      DOUBLE PRECISION D1,D2,R1,R2,F1,F2
      DOUBLE PRECISION EPS,EPSR
      C
      COMMON /PREC/EPS
      COMMON /TYPE/NO
      COMMON /NOF/NBL
      C
      NBL=0
      JTYPE=0
      FIB(1)=1.0
      FIB(2)=1.0
      C
      DO 1 I=3,50
1      FIB(I)=FIB(I-1)+FIB(I-2)
      C

```

```

      IF(NO.EQ.6) THEN
      OPEN(6,FILE='PRINT.DAT',ACCESS='SEQUENTIAL',
+     FORM='FORMATTED')
      ENDIF
C
      IF(EPS.EQ.0.0) THEN
      WRITE(5,20)
20     FORMAT(10X,'Warning: EPS = 0.0.',/,
+     10X,'The Fibonacci searching method does not run',/,
+     10X,'with this value of EPS !',/,
+     10X,'Please enter a finite, positive, small value',/,
+     10X,'of the parameter EPS.')
      RETURN
      ENDIF
C
      EPSR=1.0/EPS
C
      IF(JTYPE.EQ.1) WRITE(NO,12)EPSR
12     FORMAT(10X,'EPSR =',F20.9)
      I=2
2     IF(FIB(I).GE.EPSR) GO TO 3
      I=I+1
      GO TO 2
C
3     D1=0.0
      D2=FIB(I)
C
      IF(JTYPE.EQ.1) WRITE(NO,13)I
13     FORMAT(10X,'Number of Fibonacci numbers =',I5)
C
4     R1=D1+FIB(I-2)
      R2=D1+FIB(I-1)
C
      IF(JTYPE.EQ.1) WRITE(NO,15)D1,D2,R1,R2
15     FORMAT(10X,'D1=',F12.6,5X,'D2=',F12.6,5X,'R1=',F12.6,5X,
+     'R2=',F12.6)
C
      R11=R1*EPS
      R22=R2*EPS
C
      IF(JTYPE.EQ.1) WRITE(NO,11)R11,R22
11     FORMAT(10X,'Lambda1=',F12.6,5X,'Lambda2=',F12.6)
C
      DO 5 J=1,N
5     X1(J)=X(J)+R11*S(J)
      CALL FUNC(N,X1,F1)
      NBL=NBL+1
C
      DO 6 J=1,N
6     X1(J)=X(J)+R22*S(J)
      CALL FUNC(N,X1,F2)
      NBL=NBL+1
C
      IF(JTYPE.EQ.1) WRITE(NO,16)F1,F2
16     FORMAT(10X,'F1=',F12.6,5X,'F2=',F12.6)
C
      IF(F2.GT.F1) THEN
      D2=R2
      ELSE
      D1=R1

```

```

        ENDIF
C
        IF(R1.EQ.R2) GO TO 7
        I=I-1
        GO TO 4
C
7       BL=R11
C
        IF(JTYPE.EQ.1) WRITE(NO,17)BL,NBL
17      FORMAT(10X,'BL = ',F12.6/,
+         10X,'Number of Func. Eval. = ',I5)
        RETURN
        END
C
C*****Last card of L2.FOR

C*****
C
C  Subroutine for one dimensional searching.
C  Quadratic interpolation of Powell technique.
C  =====
C*****
C  Neculai Andrei,
C  Research Institute for Informatics
C  Advanced Modeling and Optimization Laboratory
C*****
C
        SUBROUTINE L3(N,X,S,BL)
        DOUBLE PRECISION X(N),S(N),BL
        DOUBLE PRECISION V(3),P(3),XT(50)
        DOUBLE PRECISION H,HL,F0,E,F,D,R,DM,DMM,BLM,EPS
C
        COMMON /PREC/EPS
        COMMON /TYPE/NO
        COMMON /WRIT/NX
        COMMON /L3A/ICOD
        COMMON /NOF/NBL
C
        IF(NO.EQ.6) THEN
+         OPEN(6,FILE='PRINT.DAT',ACCESS='SEQUENTIAL',
+         FORM='FORMATTED')
        ENDIF
C
        NBL=0
        ICOD=0
        JTYPE=0
        ITER=0
        H=2.0
        HL=0.5
C
        CALL FUNC(N,X,F0)
        NBL=NBL+1
        P(1)=0.0
        V(1)=F0
C
        DO 1 I=1,N
1       XT(I)=X(I)+HL*S(I)

```

```

        CALL FUNC(N,XT,F0)
        NBL=NBL+1
        P(2)=HL
        V(2)=F0
C
        IF(V(1).LT.V(2)) THEN
        DO 2 I=1,N
2       XT(I)=X(I)-HL*S(I)
        CALL FUNC(N,XT,F0)
        NBL=NBL+1
        P(3)=-HL
        V(3)=F0
C
        ELSE
C
        DO 3 I=1,N
3       XT(I)=X(I)+2.*HL*S(I)
        CALL FUNC(N,XT,F0)
        NBL=NBL+1
        P(3)=2.*HL
        V(3)=F0
        ENDIF
C
4      CONTINUE
C
        ITER=ITER+1
        IF(ITER.GT.1001) THEN
        WRITE(NX,100)
100     FORMAT(10X,'Too high accuracy for the quadratic interpolation',/,
+          10X,'technique for one dimensional searching method',/,
+          10X,'of Powell.',/,
+          10X,'Please enter another accuracy EPS')
        ICOD=1
        RETURN
        ENDIF
C
        IF(JTYPE.EQ.1) THEN
        WRITE(NO,200)
200     FORMAT(10X,'Puncts and Values')
        WRITE(NO,201) (P(I),I=1,3)
201     FORMAT(3F20.10)
        WRITE(NO,201)(V(I),I=1,3)
        ENDIF
C
        E=(P(2)*P(2)-P(3)*P(3))*V(1) +
+ (P(3)*P(3)-P(1)*P(1))*V(2) +
+ (P(1)*P(1)-P(2)*P(2))*V(3)
C
        F=(P(2)-P(3))*V(1)+(P(3)-P(1))*V(2)+(P(1)-P(2))*V(3)
        IF(F.EQ.0.0) THEN
        BL=1.0
        GO TO 10
        ENDIF
C
        BLM=(0.5*E)/F
C
        D=(P(1)-P(2))*(P(2)-P(3))*(P(3)-P(1))
C
        IF(JTYPE.EQ.1) WRITE(NO,210) E,F,BLM,D
210     FORMAT(10X,'E,F,BLM,D =',4F16.8)

```

```

C      R=F/D
C
211  IF(JTYPE.EQ.1) WRITE(NO,211)R
C      FORMAT(10X,'R =',F12.6)
C
      DM=DABS(BLM-P(1))
      DMM=DM
      IMI=1
      IMA=1
      DO 5 I=2,3
      D=DABS(BLM-P(I))
      IF(DM.GT.D) THEN
      DM=D
      IMI=I
      ENDIF
      IF(DMM.LT.D) THEN
      DMM=D
      IMA=I
      ENDIF
5      CONTINUE
C
      IF(JTYPE.EQ.1) THEN
      WRITE(NO,202)
202      FORMAT(10X,'Min Value And Max Value')
      WRITE(NO,203)DM,IMI
203      FORMAT(10X,'DM = ',F12.6,6X,'IMI = ',I4)
      WRITE(NO,204) DMM,IMA
204      FORMAT(10X,'DMM= ',F12.6,6X,'IMA = ',I4)
      ENDIF
C
      IF(DM.GT.H) THEN
      IF(R.LT.0.0) THEN
      IF(BLM.GT.P(IMI)) THEN
      P(IMA)=P(IMI)+H
      ELSE
      P(IMA)=P(IMI)-H
      ENDIF
C
      ELSE
C
      IF(BLM.LT.P(IMA)) THEN
      P(IMA)=P(IMA)-H
      ELSE
      P(IMA)=P(IMA)+H
      ENDIF
C
      ENDIF
C
      DO 6 I=1,N
6      XT(I)=X(I)+P(IMA)*S(I)
      CALL FUNC(N,XT,F0)
      NBL=NBL+1
      V(IMA)=F0
      GO TO 4
C
      ENDIF
C
      DO 7 I=1,N
7      XT(I)=X(I)+BLM*S(I)

```

```

      CALL FUNC(N,XT,F0)
      NBL=NBL+1
C
      E=V(1)
      F=E
      IMI=1
      IMA=1
      DO 8 I=2,3
      IF(E.GT.V(I)) THEN
      E=V(I)
      IMI=I
      ENDIF
      IF(F.LT.V(I)) THEN
      F=V(I)
      IMA=I
      ENDIF
8      CONTINUE
C
      IF(R.LT.0.0.AND.DM.LT.EPS) THEN
      BL=BLM
      IF(E.LT.F0) BL=P(IMI)
      GO TO 10
      ENDIF
C
      IF(R.LT.0.0.AND.(DM.GT.EPS.OR.DM.LT.H)) THEN
      P(IMA)=BLM
      V(IMA)=F0
      GO TO 4
      ENDIF
C
10      CONTINUE
C
      IF(JTYPE.EQ.1) WRITE(NO,300) BL,NBL
300      FORMAT(10X,'BL = ',F12.6,/,
      +        10X,'Number of Func. Eval. = ',I5)
      RETURN
      END
C
C***** Last card of L3.FOR

```

```

C*****
C
C  Dichotomous search technique for
C  one-dimensional minimization.
C
C*****
C  Neculai Andrei,
C  Research Institute for Informatics
C  Advanced Modeling and Optimization Laboratory
C*****
C
      SUBROUTINE L4(N,X,S,BL)
      DOUBLE PRECISION X(N),S(N),BL
      DOUBLE PRECISION EPS,A,B,C,C1,C2
      DOUBLE PRECISION XT(40),F1,F2
C
      COMMON /PREC/EPS
      COMMON /TYPE/NO

```

```

COMMON /NOF/NBL
C
  IF(NO.EQ.6) THEN
    OPEN(6,FILE='PRINT.DAT',ACCESS='SEQUENTIAL',
+     FORM='FORMATTED')
    ENDIF
C
  NBL=0
  JTYPE=0
  EPS1=EPS/10
  A=0.0
  B=1.0
C
1  C=A+(B-A)/2.
C
  IF(JTYPE.EQ.1) WRITE(NO,14)A,B
14  FORMAT(10X,'Interval: A=',F12.6,5X,'B = ',F12.6)
C
  C1=C+EPS1/2.
  DO 2 I=1,N
2   XT(I)=X(I)+C1*S(I)
  CALL FUNC(N,XT,F1)
  NBL=NBL+1
C
  IF(JTYPE.EQ.1) WRITE(NO,10)C1,F1
10  FORMAT(10X,'C1 = ',F12.6,5X,'F1 = ',F18.6)
C
  C2=C-EPS1/2.
  DO 3 I=1,N
3   XT(I)=X(I)+C2*S(I)
  CALL FUNC(N,XT,F2)
  NBL=NBL+1
C
  IF(JTYPE.EQ.1) WRITE(NO,11)C2,F2
11  FORMAT(10X,'C2 = ',F12.6,5X,'F2 = ',F18.6)
C
  IF(F2.LT.F1) THEN
    B=C1
  ELSE
    A=C2
  ENDIF
C
  IF(DABS(B-A).LE.EPS) GO TO 4
  GO TO 1
4   BL=C
C
  IF(JTYPE.EQ.1) WRITE(NO,100)BL
100  FORMAT(10X,'BL = ',F18.6,/,
+     10X,'Number of Func. Eval. = ',I5)
C
  RETURN
END
C
C*****Last card of L4.FOR

```

```

*-----
*
* Subroutine for execution time computation.
*-----
*
      subroutine exetim(tih,tim,tis,tic, tfh,tfm,tfs,tfc)
*
      integer*4 tih,tim,tis,tic
      integer*4 tfh,tfm,tfs,tfc
*
      integer*4 ti,tf
      integer*4 ch,cm,cs
      data ch,cm,cs/360000,6000,100/
*
      ti=tih*ch+tim*cm+tis*cs+tic
      tf=tfh*ch+tfm*cm+tfs*cs+tfc
      tf=tf-ti
      tfh=tf/ch
      tf=tf-tfh*ch
      tfm=tf/cm
      tf=tf-tfm*cm
      tfs=tf/cs
      tfc=tf-tfs*cs
*
      return
      end
*----- End of EXETIM

```

***** FUNCTION TO BE MINIMIZED *****

```

C
C
C      FUNCTION FUNC8.FOR
C      =====
C
C      Date created:   April 9, 1991
C      Date last modified:
C
C      Remark:
c      This is the function NEC.
C*****
      SUBROUTINE FUNC(N,X,F)
      DOUBLE PRECISION X(N),F
C
      F=2.*X(1)**2+X(2)**2+2.*X(1)*X(2)+X(1)-X(2)
C
      RETURN
      END
C
C      Remark:
C      X* = [-1.0  1.5] ; F(X*) = -1.25
C
C*****Last card of FUNC8.FOR

```


UNO Examples

Example 1

```

C*****
C
      SUBROUTINE FUNC(N,X,F)
      DOUBLE PRECISION X(N),F
      DOUBLE PRECISION A,B,C,D
C
      A= 28.*X(1)+36.*X(2)-11.*X(3)-42.*X(4)-56.
      B=-18.*X(1)-41.*X(2)+78.*X(3)-12.*X(4)-26.
      C= 43.*X(1)- 3.*X(2)-24.*X(3)+18.*X(4)-62.
      D= 29.*X(1)+45.*X(2)-10.*X(3)- 7.*X(4)-18.
C
      F=A*A+B*B+C*C+D*D
C
      RETURN
      END
C
C***** Last card of FUNC.FOR

      *****Optimization Method of HOOK-JEEVES.*****
      Using Lambda* = 1.0
      Initial Point :
          1.000000
          1.000000
          1.000000
          1.000000
      Step H      =      0.800000
      Precision EPS =      0.1000000000000000E-05
      Val. Func. in Initial point =      4691.000000

      -----
      Iterations =      36
      No. Function Evaluations =      684
      Precision = 0.000001000
      ***** Minimum Point *****
          1.843000
          -0.896929
          0.146880
          -0.911932
      Val. Func. in Minimum Point =      0.000000
      ##### ----- Time=      193c

      *****Optimization Method of HOOK-JEEVES.*****
      Using Golden Searching Technique
      Initial Point :
          1.000000
          1.000000
          1.000000
          1.000000
      Step H      =      0.800000
      Precision EPS =      0.1000000000000000E-05
      Val. Func. in Initial point =      4691.000000

```

```

-----
Iterations = 43
No. Function Evaluations = 3353
Precision = 0.000001000
***** Minimum Point *****
1.843000
-0.896916
0.146893
-0.911918
Val. Func. in Minimum Point = 0.000000
##### ----- Time= 154c

****Optimization Method of HOOK-JEEVES.****
Using Fibonacci Searching Technique
Initial Point :
1.000000
1.000000
1.000000
1.000000
Step H = 0.800000
Precision EPS = 0.1000000000000000E-05
Val. Func. in Initial point = 4691.000000

-----
Iterations = 40
No. Function Evaluations = 3013
Precision = 0.000001000
***** Minimum Point *****
1.843001
-0.896924
0.146884
-0.911927
Val. Func. in Minimum Point = 0.000000
##### ----- Time= 158c

****Optimization Method of HOOK-JEEVES.****
Using Quadratic Interpolation of Powell
Initial Point :
1.000000
1.000000
1.000000
1.000000
Step H = 0.800000
Precision EPS = 0.1000000000000000E-05
Val. Func. in Initial point = 4691.000000

-----
Iterations = 57
No. Function Evaluations = 1210
Precision = 0.000001000
***** Minimum Point *****
1.843001
-0.896927
0.146882
-0.911930

```

```

Val. Func. in Minimum Point = 0.000000
##### ----- Time= 171c

**** Optimization Method of Rosenbrock ****
Initial Point :
    1.000000
    1.000000
    1.000000
    1.000000
Accuracy EPS = 0.000001
Val. Func. in Initial Point = 4691.000000

-----
Iterations = 78
No. Function Evaluations = 781
Accuracy = 0.000001000
***** Minimum Point *****
    1.843008
   -0.896933
    0.146882
   -0.911930
Val. Func. in Minimum Point = 0.000000
##### ----- Time= 163c

**** Optimization Method of Powell ****
Using Golden Searching Technique
Initial Point :
    1.000000
    1.000000
    1.000000
    1.000000
Accuracy EPS = 0.1000000000000000E-05
Function Value in Initial Point = 4691.000000

-----
Iterations = 14
No. of Function Evaluations = 4097
Accuracy = 0.000001000
***** Minimum Point *****
    1.843000
   -0.896848
    0.146918
   -0.911801
Function Value in Minimum Point = 0.000019
##### ----- Time= 244c

**** Optimization Method of Powell ****
Using Fibonacci Searching Technique
Initial Point :
    1.000000
    1.000000
    1.000000
    1.000000
Accuracy EPS = 0.1000000000000000E-05
Function Value in Initial Point = 4691.000000

-----
Iterations = 14

```

```

        No. of Function Evaluations = 3725
        Accuracy = 0.000001000
        ***** Minimum Point *****
                1.843003
                -0.896928
                0.146883
                -0.911929
        Function Value in Minimum Point = 0.000000
##### ----- Time= 198c

        ***** Optimization Method of Powell *****
        Using Quadratic Interpolation of Powell
        Initial Point :
                1.000000
                1.000000
                1.000000
                1.000000
        Accuracy EPS = 0.1000000000000000E-05
        Function Value in Initial Point = 4691.000000

        -----
        Iterations = 6
        No. of Function Evaluations = 218
        Accuracy = 0.000001000
        ***** Minimum Point *****
                1.843003
                -0.896928
                0.146883
                -0.911929
        Function Value in Minimum Point = 0.000000
##### ----- Time= 370c

        ***** Optimization Method of Powell *****
        Using Dichotomous Searching Technique
        Initial Point :
                1.000000
                1.000000
                1.000000
                1.000000
        Accuracy EPS = 0.1000000000000000E-05
        Function Value in Initial Point = 4691.000000

        -----
        Iterations = 12
        No. of Function Evaluations = 2430
        Accuracy = 0.000001000
        ***** Minimum Point *****
                1.843003
                -0.896929
                0.146882
                -0.911930
        Function Value in Minimum Point = 0.000000
##### ----- Time= 102c

        ***** Optimization Method of Nelder-Mead *****
        Initial Point :
                1.000000
                1.000000

```

```

1.000000
1.000000
Accuracy EPS = 0.1000000000000000E-05
Val. Func. in Initial Point = 4691.000000

-----
Iterations = 105
No. Function Evaluations = 293
Accuracy = 0.000001000
***** Minimum Point *****
1.842990
-0.896909
0.146889
-0.911929
Val. Func. in Minimum Point = 0.000001
##### ----- Time= 142c

**** Parallel with Axes Optimization Method ****
Using Golden Seraching Technique
Initial Point :
1.000000
1.000000
1.000000
1.000000
Accuracy EPS = 0.1000000000000000E-05
Function Value in Initial point = 4691.000000

-----
Iterations = 161
No.Func. Eval = 484
Precision = 0.000001
***** Minimum Point *****
1.842952
-0.896720
0.146998
-0.911771
Function Value in Minimum Point = 0.000043
##### ----- Time= 271c

**** Parallel with Axes Optimization Method ****
Using Fibonacci Searching Technique
Initial Point :
1.000000
1.000000
1.000000
1.000000
Accuracy EPS = 0.1000000000000000E-05
Function Value in Initial point = 4691.000000

-----
Iterations = 161
No.Func. Eval = 484
Precision = 0.000001
***** Minimum Point *****
1.842952
-0.896720
0.146999
-0.911771
Function Value in Minimum Point = 0.000043

```

----- Time= 161c

Example 2

```
C      Date created:      April 2, 1991
C      Date last modified:
C
C*****
C      SUBROUTINE FUNC(N,X,F)
C      DOUBLE PRECISION X(N),F
C
C      F=4.*(X(1)-5.)**2+(X(2)-6.)**2
C
C      RETURN
C      END
C
C*****Last card of FUNC.FOR
```

```
*****Optimization Method of HOOK-JEEVES.*****
Using Lambda* = 1.0
Initial Point :
    1.000000
    1.000000
Step H          =    0.800000
Precision EPS =    0.0000000000000000
Val. Func. in Initial point =    89.000000
```

```
-----
    Iterations =    5
    No. Function Evaluations =    175
    Precision =  0.000000100
***** Minimum Point *****
    5.000000
    6.000000
    Val. Func. in Minimum Point =    0.000000
##### ----- Time= 164c
```

```
*****Optimization Method of HOOK-JEEVES.*****
Using Golden Searching Technique
Initial Point :
    1.000000
    1.000000
Step H          =    0.800000
Precision EPS =    0.1000000000000000E-05
Val. Func. in Initial point =    89.000000
```

```
-----
    Iterations =    4
    No. Function Evaluations =    342
    Precision =  0.000001000
***** Minimum Point *****
    5.000000
    5.999999
    Val. Func. in Minimum Point =    0.000000
```

```

##### ----- Time=          74c

      ***** Optimization Method of Rosenbrock *****
      Initial Point :
            1.000000
            1.000000
      Accuracy EPS =          0.000001
      Val. Func. in Initial Point =          89.000000

-----
      Iterations =      25
      No. Function Evaluations =      186
      Accuracy =      0.000001000
      ***** Minimum Point *****
            5.000006
            6.000007
      Val. Func. in Minimum Point =          0.000000
##### ----- Time=          139c

      ***** Optimization Method of Powell *****
      Using Quadratic Interpolation of Powell
      Initial Point :
            1.000000
            1.000000
      Accuracy EPS =          0.1000000000000000E-05
      Function Value in Initial Point =          89.000000

-----
      Iterations =      2
      No. of Function Evaluations =      34
      Accuracy =      0.000001000
      ***** Minimum Point *****
            5.000000
            6.000000
      Function Value in Minimum Point =          0.000000
##### ----- Time=          82c

```

Example 3 (Banana of Rosenbrock)

```

C      Date created:          April 2,1991
C      Date last modified:
C
C*****
C      SUBROUTINE FUNC(N,X,F)
C      DOUBLE PRECISION X(N),F
C      F=100.*(X(2)-X(1)*X(1))**2+(1.-X(1))**2
C      RETURN
C      END

C Initial Point:
C x(1) = -1.2
C x(2) = 1.0
C
C*****Last card of FUNC.FOR

```

```

*****Optimization Method of HOOK-JEEVES.*****
Using Golden Searching Technique
Initial Point :
    -1.200000
     1.000000
Step H      =      0.800000
Precision EPS =      0.1000000000000000E-05
Val. Func. in Initial point =      24.200000

-----
Iterations =    53
No. Function Evaluations =   3880
Precision =  0.000001000
***** Minimum Point *****
    1.000191
    1.000382
Val. Func. in Minimum Point =    0.000000
##### ----- Time=    428c

*****Optimization Method of HOOK-JEEVES.*****
Using Fibanacchi Searching Technique
Initial Point :
    -1.200000
     1.000000
Step H      =      0.800000
Precision EPS =      0.1000000000000000E-05
Val. Func. in Initial point =      24.200000

-----
Iterations =    63
No. Function Evaluations =   4211
Precision =  0.000001000
***** Minimum Point *****
    1.000113
    1.000226
Val. Func. in Minimum Point =    0.000000
##### ----- Time=    176c

***** Optimization Method of Rosenbrock *****
Initial Point :
    -1.200000
     1.000000
Accuracy EPS =      0.000001
Val. Func. in Initial Point =      24.200000

-----
Iterations =    82
No. Function Evaluations =    582
Accuracy =  0.000001000
***** Minimum Point *****
    0.999992
    0.999983
Val. Func. in Minimum Point =    0.000000
##### ----- Time=    135c

```


Example 4 (Wood's function)

```
C      Remark:
C      Wood's function.
C*****
      SUBROUTINE FUNC(N,X,F)
      DOUBLE PRECISION X(N),F
      F=100.*(X(2)-X(1)*X(1))**2+(1.-X(1))**2+
+      90.*(X(4)-X(3)*X(3))**2+(1.-X(3))**2+
+      10.1*((X(2)-1.)**2+(X(4)-1.)**2)+
+      19.8*(X(2)-1.)*(X(4)-1.)
      RETURN
      END
C
C
C      Remark:
C      X* = (1.0  1.0  1.0  1.0)
C      F(X*) = 0.0
C
C*****Last card of FUNC.FOR

      *****Optimization Method of HOOK-JEEVES.*****
      Using Quadratic Interpolation of Powell
      Initial Point :
          2.000000
          2.000000
          2.000000
          2.000000
      Step H      =      0.800000
      Precision EPS =      0.10000000000000000E-05
      Val. Func. in Initial point =      802.000000

-----
      Iterations =      3
      No. Function Evaluations =      302
      Precision =  0.000001000
      ***** Minimum Point *****
          1.000000
          1.000000
          1.000000
          0.999999
      Val. Func. in Minimum Point =      0.000000
      ##### ----- Time=      149c

      ***** Optimization Method of Rosenbrock *****
      Initial Point :
          2.000000
          2.000000
          2.000000
          2.000000
      Accuracy EPS =      0.000001
      Val. Func. in Initial Point =      802.000000

-----
      Iterations =  140
      No. Function Evaluations =  1075
      Accuracy =  0.000001000
      ***** Minimum Point *****
```

```

0.999421
0.998814
1.000626
1.001208
Val. Func. in Minimum Point = 0.000002
##### ----- Time= 182c

```

******* Optimization Method of Nelder-Mead *******

```

Initial Point :
2.000000
2.000000
2.000000
2.000000
Accuracy EPS = 0.1000000000000000E-05
Val. Func. in Initial Point = 802.000000

```

```

-----
Iterations = 196
No. Function Evaluations = 534
Accuracy = 0.000001000
***** Minimum Point *****
0.999861
0.999819
1.000102
1.000177
Val. Func. in Minimum Point = 0.000001
##### ----- Time= 270c

```

******* Parallel with Axes Optimization Method *******
Using Fibonacci Searching Technique

```

Initial Point :
2.000000
2.000000
2.000000
2.000000
Accuracy EPS = 0.1000000000000000E-05
Function Value in Initial point = 802.000000

```

```

-----
Iterations = 152
No.Func. Eval = 457
Precision = 0.000001
***** Minimum Point *****
1.000487
1.000875
1.000057
1.000016
Function Value in Minimum Point = 0.000011
##### ----- Time= 146c

```

Example 5 (Reactor) (UNO) (non-interactive version)

cF78
*

REACTOR

```

*
78      continue

      k1d=31.24d0
      k2d=0.272d0
      k3d=303.03d0
      r1=2.062d0
      r2=0.02d0

      fx(1)=1.d0 - x(1) - k1d*x(1)*x(6) + r1*x(4)
      fx(2)=1.d0 - x(2) - k2d*x(2)*x(6) + r2*x(5)
      fx(3)=-x(3) + 2.d0*k3d*x(4)*x(5)
      fx(4)=k1d*x(1)*x(6) - r1*x(4) - k3d*x(4)*x(5)
      fx(5)=1.5d0*k2d*x(2)*x(6)-1.5d0*r2*x(5)-k3d*x(4)*x(5)
      fx(6)=1.d0 - x(4) - x(5) - x(6)

      f=0.d0
      do i=1,6
        f = f + fx(i)**2
      end do

      return
*****

```

78 UNO Algorithm. Function: REACTOR
Nelder - Mead method

n	iter	nfunc	time(c)	f

Initial value: 0.1961733675060E+08				
1	0.7621259215851E-01			
2	0.3344399186419E+01			
3	-0.1589175762111E+01			
4	0.7497734772275E-03			
5	0.1924846471132E+01			
6	0.3086517549602E+00			
Final value: 0.1224004112154E+02				
6	1001	2586	0	0.1224004112154E+02

TOTAL	1001	2586	0.00(seconds)	

78 UNO Algorithm. Function: REACTOR
Rosenbrock method

n	iter	nfunc	time(c)	f

Initial value: 0.1961733675060E+08				
1	0.2856656208381E+00			
2	-0.8280152891094E+00			
3	0.7893227536243E+01			
4	0.1197713574015E+00			
5	0.7268740531020E-01			
6	0.2222657979826E+00			
Final value: 0.0000000000000E+00				
6	1001	6262	1	0.0000000000000E+00

TOTAL	1001	6262	0.01(seconds)	

Example 6 (Propan) (UNO) (non-interactive version)

cF77

PROPAN

```

77      continue
      r5=0.193d0
      r6=0.4106217541d-3
      r7=0.5451766686d-3
      r8=0.44975d-6
      r9=0.3407354178d-4
      r10=0.9615d-6
      r=10.d0

      fx(1)=x(1)*x(2) + x(1) - 3.d0*x(5)
      fx(2)=2.d0*x(1)*x(2) + x(1) + 2.d0*r10*x(2)**2 +
            x(2)*x(3)**2 +
1      r7*x(2)*x(3)+r9*x(2)*x(4)+r8*x(2) - r*x(5)
      fx(3)=2.d0*x(2)*x(3)**2 + r7*x(2)*x(3) +
            2.d0*r5*x(3)**2 +
1      r6*x(3) - 8.d0*x(5)
      fx(4)=r9*x(2)*x(4) + 2.d0*x(4)**2 - 4.d0*r*x(5)
      fx(5)=x(1)*x(2) + x(1) + r10*x(2)**2 + x(2)*x(3)**2 +
1      r7*x(2)*x(3) + r9*x(2)*x(4) + r8*x(2) +
            r5*x(3)**2 +
1      r6*x(3) + x(4)**2 - 1.d0

      f=0.d0
      do i=1,5
        f = f + fx(i)**2
      end do
      return

```

77 UNO Algorithm. Function: PROPAN
Nelder - Mead method

n	iter	nfunc	time(c)	f

Initial value: 0.2502293538479E+01				
1	0.3575641244465E-02			
2	0.3013621797270E+02			
3	0.6967916318585E-01			
4	0.8593444895874E+00			
5	0.3694564597281E-01			
Final value: 0.3282712795731E-06				
5	131	365	0	0.3282712795731E-06

TOTAL	131	365	0.00(seconds)	

77 UNO Algorithm. Function: PROPAN
Rosenbrock method

n	iter	nfunc	time(c)	f

Initial value: 0.2502293538479E+01				
1	0.3557047101060E-02			

2	0.3029435485404E+02			
3	0.6949834690195E-01			
4	0.8593335580487E+00			
5	0.3694505366551E-01			
Final value: 0.3047133834219E-06				
5	110	1057	0	0.3047133834219E-06

TOTAL	110	1057	0.00(seconds)	

77 UNO Algorithm. Function: PROPAN
Hook - Jeeves method with Golden search

n	iter	nfunc	time(c)	f

1	0.6870885373936E-02			
2	0.1551259400171E+02			
3	0.9689770616690E-01			
4	0.8587765673185E+00			
5	0.3688609219850E-01			
Final value: 0.1034639605992E-04				
5	230	27853	1	0.1034639605992E-04

TOTAL	230	27853	0.01(seconds)	

Example 7 Elastic-plastic torsion problem. **MINPACK-2 Project**

The Elastic-Plastic Torsion problem is from MINPACK collection of optimization problems. For this problem we present a solution for the discretization $n_x=n_y=20$ points (400 variables). Figure 1 presents the solution obtained with Hook-Jeeves method and Golden searching line search in the program UNODEPT.FOR (interactive version)

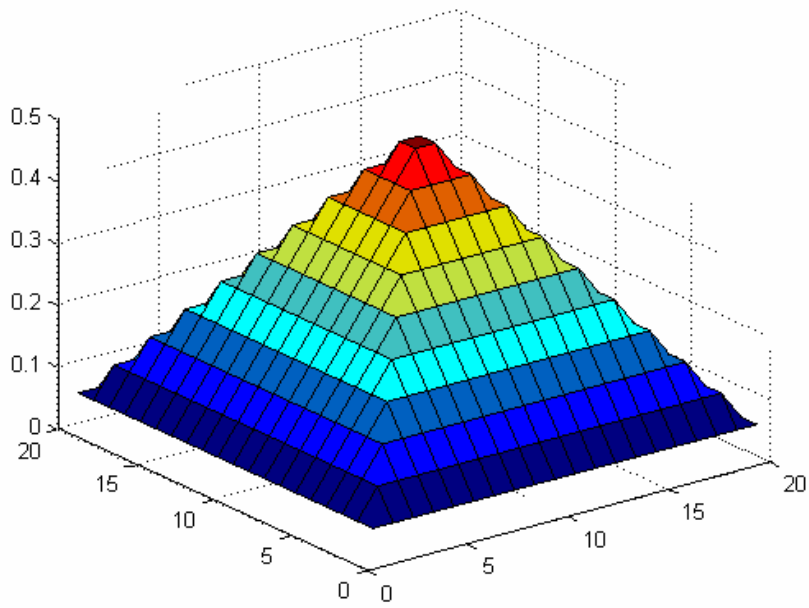


Fig. 1. Solution given by Hook-Jeeves method and Golden Searchnig line search.

Figure 2 presents the solution of this problem given by Hook-Jeeves method and Fibonacci searching line search.

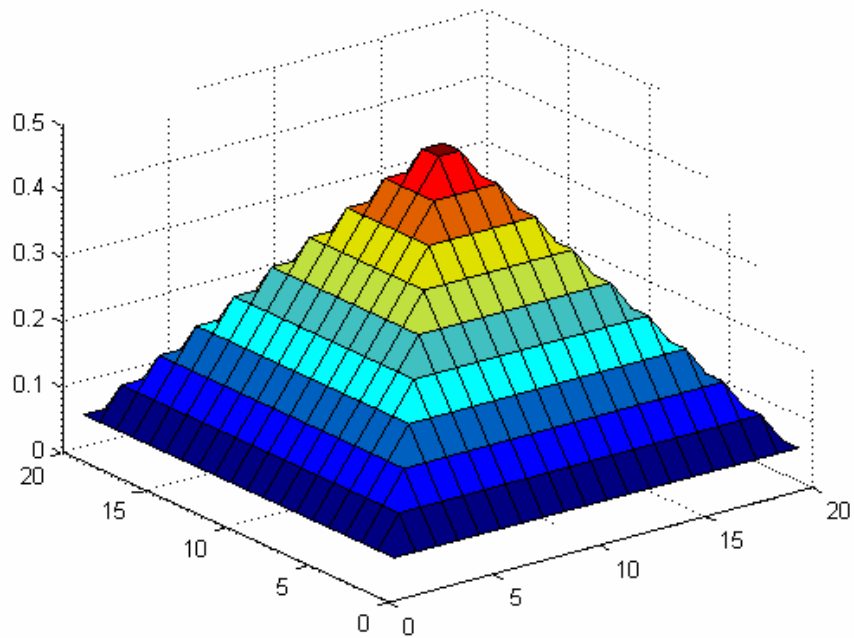


Fig. 2. Solution given by Hook-Jeeves method and Fibonacci Searchnig line search.

Figure 3 illustrate the solution given by UNODEPT.FOR using the method of ROSENBROCK.

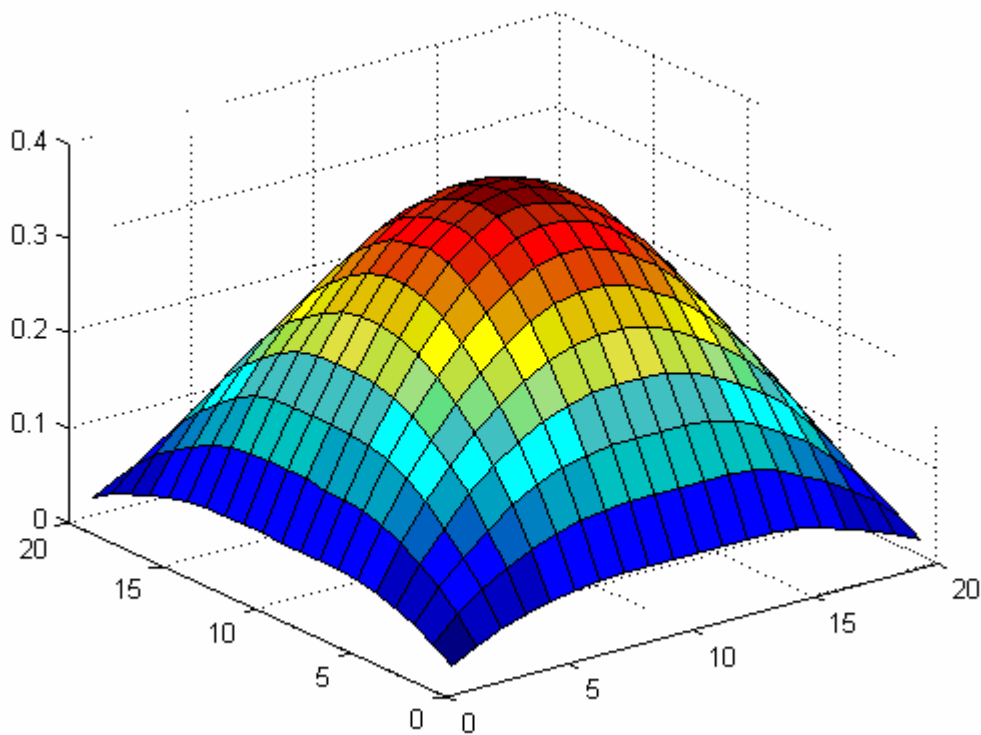


Fig. 3. Solution given by Rosenbrock method.

Observe that the solution given by Rosenbrock method is better than that given by Hook-Jeeves.

References

Andrei, N., (1999) *Programarea Matematică Avansată. Teorie, Metode Computaționale, Aplicații*. Editura Tehnică, București, 1999. (Cap. 18)

Andrei, N., (2009) *Critica Rațiunii Algoritmilor de Optimizare fără Restricții*. Editura Academiei Române, București, 2009. (Cap. 16)

Hooke, R., Jeeves, T.A., (1961) *Direct search solution of numerical and statistical problems*. Journal of the Association for Computing Machinery, vol. 8, 1961, pp.212-229.

Nelder, J.A, Mead, R., (1965) *A simplex method for function minimization*. The Computer Journal, vol. 7, 1965, pp. 308-313.

Powell, M.J.D., (1964) *An efficient method of finding the minimum of a function of several variables without calculating derivatives*. The Computer Journal, vol. 7, 1964, pp.155-162.

Rosenbrock, H.H., (1960) *An automatic method for finding the greatest or least value of a function*. The Computer Journal, vol. 3, 1960, pp. 175-184.

Note:

UNOINT package has been elaborated in April 1991 during my stay in Duisburg University under a very generous financial support given by the **Alexander von Humboldt Foundation**.

This package and its non-interactive version (UNO) have been used in 2007 for the numerical experiments included in Chapter 16 of the book: **N. Andrei**, “**Criticism of the unconstrained optimization algorithms reasoning**”, published in Publishing House of Romanian Academy, 2009. (ISBN 978-973-27-1669-4)

New version: April 20, 2011

