APPENDIX - 2.1

COMPUTER PROGRAM FOR D S C - POWELL ALGORITHM

```
$DEBUG
C  ================================================================
C          AREAOPT     : A General Purpose Optimiser Program
C            BASED ON THE DSC-POWELL COMBINATION ALGORITHM
C                     Programmed by : R.SENGUPTA
C                  CHEMICAL ENGINEERING DEPARTMENT
C          FACULTY OF TECHNOLOGY AND ENGINEERING  , KALABHAVAN
C                                BARODA.
C  ================================================================
C                        NOTICE TO USER
C  ----------------------------------------------------------------
C        AREAOPT  is a general purpose multivariate optimization
C            program. In its present form it is able to optimize
C            upto six variables. Though in a number of cases its
C            convergence has been proved, it can not be guaranteed
C            as it depends upon the system equations. Most important
C            restriction at present is that all the variables must
C            positive values, any -ve sign should be taken care of
C            in the ERROR subroutine which is effectively the
C            application specific part of the program.
C         ERROR  subroutine must be written and included at the
C            place shown below by the user. ERROR should calculate
C            and return error corresponding to the current values of
C            variables K,K2, etc.
C  ----------------------------------------------------------------
C     THE FOLLOWING NOTATION IS USED :
C     K,K2
C     T1,T21
C     T2,T22    these are six variables for which optimization will
C               be done, in two steps : K,T1,T2  and K2,T21,T22.
C     N = always zero here
C     INUM = NUMBER OF INPUT-OUTPUT POINTS
C     U(I) = PROCESS INPUT
C     X(I) = PROCESS OUTPUT
C     T= always 1.0
C     THE PROGRAM CALLS A DATA FILE of initial values NAMED
C     'xxxxx.ini'. ITEMS IN THE FILES ARE SET UP IN THE FOLLOWING ORDER
C     ENTRY 1    K,K2     (INITIAL GUESS)
C     ENTRY 2    T1,T21   (INITIAL GUESS)
C     ENTRY 3    T2,T22   (INITIAL GUESS)
C     ENTRY 4    INUM
C     ENTRIES 6 TO INUM+6  U(I) FOR I = 1 TO INUM+1
C     LAST ENTRIES         X(I) FOR I = 1 TO INUM+1
C     PROGRAM CALLS A DATA BANK TO BE OPTIMISED NAMED xxxxxx.DAT

      REAL K,K2
      COMMON X(350),U1(350),U2(350),U3(350),U4(350),XP(350),F(350)
      COMMON INUM, T, N, J, HGHT ,U5(350),ERR(350),U6(350),U7(350)
      COMMON XAOPT (6) , XBOPT (6) , XCOPT (6) , XSOPT(6)
      COMMON FAOPT (6) , FBOPT (6) , FCOPT (6) , FSOPT(6)
      COMMON XMM2, XMM1, XM, XMP1, FMM2, FMM1, FM
      COMMON FMP1, DK,K,T1,T2                  ,K2, T21,T22
      COMMON PREOPT ,FREOPT
```

```
      OPEN (UNIT = 5, FILE = 'INKLA.INI ')
      READ (5,50) K ,K2,T1,T21,T2,T22
   50 FORMAT ( 2F12.5)
      READ (5,60 ) INUM
   60 FORMAT (I3)
      CLOSE (5)
      OPEN (UNIT = 6, FILE = 'KLA200.DAT')
      DO 100 I = 1,INUM
      READ (6 ,80) U1(I),U2(I),U3(I),U4(I),U5(I),U6(I),X(I)
   80 FORMAT (E12.4,6(E13.4))
  100   CONTINUE
       WRITE (*, 110)
  110  FORMAT ('0COMPUTATION IN PROGRESS DO NOT DISTURB')
C     IC IS CYCLE NUMBER
      N=0
      T=1.0
      IC=1
      WRITE(*, 160)
160   FORMAT ('0STARTING VALUES')
      WRITE(*,165)
165   FORMAT (10X,'K',14X,'T1',13X,'T2')
C
C     IN THE NEXT SECTION OF THE PROGRAM, K,T1,T2,K2,T21 AND T22 ARE
C     OPTIMIZED SEPARATELY USING A DSC-POWELL SEARCH.
C     A PART OF THE DSC SEARCH IS IN A SUBROUTINE.
C     THE POWELL SEARCH IS AT THE END OF THE MAIN PROGRAM.
171   FORMAT( 3(F15.6))
170   P1=K
      P2=T1
      P3=T2
      P21 =K2
      P22 = T21
      P23 = T22
      WRITE(*,175) IC
      WRITE(*,171) K,T1,T2
      WRITE (*,171) K2,T21,T22
      WRITE(*,175) IC
175   FORMAT(' CYCLE',I3)
      Z=ERROR(1,K,T1,T2)
      WRITE(*,*)Z
      CALL ITER(1,K,T1,T2)
      WRITE (*,178)
178   FORMAT ('01 ITER OVER')
      WRITE (*,*) K ,T1 ,T2
      CALL ITER(2,K2,T21,T22)
      WRITE (*,180)
180   FORMAT ('02 ITER OVER')
      WRITE (*,*) K2 ,T21 ,T22
C     THIS SECTION OF THE PROGRAM DETERMINES IF ALL 6 OF
C     THE VARIABLES ARE WITHIN THE PRESCRIBED ACCURACY
C     (YACCUR) OF THE PREVIOUS CYCLE VALUES. IF THEY
C     ARE NOT THEN THE OPTIMIZATION IS REPEATED.
480   IC=IC+1
      YACCUR=.001
      IF(ABS(K-P1).GT.YACCUR) GO TO 170
      IF(ABS(K2-P21).GT.YACCUR) GO TO 170
```

```
      YACCUR=YACCUR*2.
      IF(ABS(T1-P2).GT.YACCUR) GO TO 170
      IF(ABS(T2-P3).GT.YACCUR) GO TO 170
      IF(ABS(T21-P22).GT.YACCUR) GO TO 170
      IF(ABS(T22-P23).GT.YACCUR) GO TO 170
      WRITE(*,490)
490   FORMAT('0OPTIMUM VALUES')
      WRITE(*,500) K,T1,T2
      WRITE(*,500) K2,T21,T22
      Z=ERROR(1,K,T1,T2)
      WRITE(*,*)Z
500   FORMAT( 3(E15.6))
      STOP
      END

C     ------------------------------------------------
      SUBROUTINE ITER(IN,SK,ST1,ST2)
      REAL K,K2
      COMMON X(350),U1(350),U2(350),U3(350),U4(350),XP(350),F(350)
      COMMON INUM, T, N, J, HGHT, U5(350),ERR(350),U6(350),U7(350)
      COMMON XAOPT (6) , XBOPT (6) , XCOPT (6) , XSOPT(6)
      COMMON FAOPT (6) , FBOPT (6) , FCOPT (6) , FSOPT(6)
      COMMON XMM2, XMM1, XM, XMP1, FMM2, FMM1, FM
      COMMON FMP1, DK,K,T1,T2                  ,K2, T21,T22
      COMMON PREOPT ,FREOPT

C     OPTIMIZATION OF K

C     DSC SEARCH

      WRITE (*,190 )
 190  FORMAT ('OK OPTIMIZATION')
      L=1
      DK=.001
      J=1
      PREOPT=SK
      F(J)=ERROR(IN,SK,ST1,ST2)
      IW1=J+1
      F(J+1)=ERROR (IN,SK+DK,ST1,ST2)
      IF(F(J+1).GE.F(J)) DK=-DK
      IF(SK+DK.LE.0.) GO TO 212
200   IW1=J+1
      F(J+1)=ERROR(IN,SK+DK,ST1,ST2)
      IF(F(J+1).GT.F(J)) GOTO 210
      OLDDK=DK
      SK=SK+DK
      PREOPT=SK
      FREOPT=F(J+1)
      DK=2.*DK
      IF(DK.GT.0.5) DK=.5
      IF(DK.LT.-0.5) DK=-.5
      IF(SK+DK.LT.0.) GO TO 205
      J=J+1
      GO TO 200
205   DK=OLDDK
      SK=SK-DK
```

```
210    XM=SK+DK
       XMM1=SK
       XMM2=SK-.5*DK
       XMP1=SK+.5*DK
       FM=F(J+1)
       FMM1=F(J)
       FMM2=ERROR (IN,XMM2,ST1,ST2)
       FMP1=ERROR (IN,XMP1,ST1,ST2)
       CALL DSC(IN,1)
       SK=XSOPT(L)
        GO TO 248

C      OPTIMIZATION OF T1
C      DSC SEARCH
212    WRITE (*,215 )
215    FORMAT ('OT1 OPTIMIZATION')
218    L=2
       DK=.001
       J=1
       PREOPT=ST1
       F(J)=ERROR(IN,SK,ST1,ST2)
       IW1=J+1
       F(J+1)=ERROR(IN,SK,ST1+DK,ST2)
       IF(F(J+1).GE.F(J)) DK=-DK
       IF((ST1+DK)/T.LT.0.001) GO TO 231
225    IW1=J+1
       F(J+1)=ERROR(IN,SK,ST1+DK,ST2)
       IF(F(J+1).GT.F(J)) GO TO 230
       OLDDK=DK
       ST1=ST1+DK
       PREOPT=ST1
       FREOPT=F(J+1)
       DK=2.*DK
       IF(DK.LT.-0.5) DK=-.5
       IF(ST1+DK.LT.0.) GO TO 227
       IF((ST1+DK)/T.LT.0.001) GO TO 227
       J=J+1
       GO TO 225
227    DK=OLDDK
       ST1=ST1-DK
230    XM=ST1+DK
       XMM1=ST1
       XMM2=ST1-.5*DK
       XMP1=ST1+.5*DK
       FM=F(J+1)
       FMM1=F(J)
       FMM2=ERROR(IN,SK,XMM2,ST2)
       FMP1=ERROR(IN,SK,XMP1,ST2)
       CALL DSC(IN,2)
       ST1=XSOPT(L)
       GO TO 248

C      OPTIMIZATION OF T2
231    WRITE (*,232 )
232    FORMAT ('OT2 OPTIMIZATION')
235    L=3
       DK=.001
```

```
      J=1
      PREOPT=ST2
      F(J)=ERROR(IN,SK,ST1,ST2)
      IW1=J+1
      F(J+1)=ERROR(IN,SK,ST1,ST2+DK)
      IF(F(J+1).GE.F(J)) DK=-DK
      IF((ST2+DK)/T.LT.0.001) GO TO 480
245   IW1=J+1
      F(J+1)=ERROR(IN,SK,ST1,ST2+DK)
      IF(F(J+1).GE.F(J)) GO TO 247
      OLDDK=DK
      ST2=ST2+DK
      PREOPT=ST2
      DK=2.*DK
      FREOPT=F(J+1)
      IF(DK.GT.0.5)  DK=.5
      IF(DK.LT.-0.5)  DK=-.5
      IF(ST2+DK.LT.0.0) GO TO 246
      IF((ST2+DK)/T.LT.0.001) GO TO 246
      J=J+1
      GO TO 245
246   DK=OLDDK
      ST2=ST2-DK
247   XM=ST2+DK
      XMM1=ST2
      XMM2=ST2-.5*DK
      XMP1=ST2+.5*DK
      FM=F(J+1)
      FMM1=F(J)
      FMM2=ERROR(IN,SK,ST1,XMM2)
      FMP1=ERROR(IN,SK,ST1,XMP1)
      CALL DSC(N,3)
      ST2=XSOPT(L)

C     POWELL SEARCH

C     THIS SECTION OF THE PROGRAM CALCULATES "XSTAR" WHICH
C     IS THE OPTIMUM VALUE OF ONE OF THE VARIABLES.
248   WRITE (*,249)
249   FORMAT ('OPOWELL SEARCH')
250   IF(XSOPT(L).EQ.PREOPT) GO TO 470
      PREOPT=XSOPT(L)
      FREOPT=FSOPT(L)
      XA=XAOPT(L)
      XB=XBOPT(L)
      XC=XCOPT(L)
      XSTAR=XSOPT(L)
      FA=FAOPT(L)
      FB=FBOPT(L)
      FC=FCOPT(L)
      FSTAR=FSOPT(L)
255   XACCUR=.0005
      I1=INT(FA*1.E4+.5)
      I2=INT(FB*1.E4+.5)
      I3=INT(FC*1.E4+.5)
      I4=INT(FSTAR*1.E4+.5)
      IF(I1.EQ.I2.AND.I2.EQ.I3.AND.I3.EQ.I4) GO TO 470
```

```
290   IF(FA.GE.FB.AND.FA.GE.FC) GO TO 300
      IF(FB.GE.FA.AND.FB.GE.FC) GO TO 310
      IF(FC.GE.FA.AND.FC.GE.FB) GO TO 320
300   FBIG=FA
      XBIG=XA
      IF(FB.LE.FC) GO TO 307
305   FSMALL=FC
      XSMALL=XC
      GO TO 330
307   FSMALL=FB
      XSMALL=XB
      GO TO 330
310   FBIG=FB
      XBIG=XB
      IF(FA.GT.FC) GO TO 305
315   FSMALL=FA
      XSMALL=XA
      GO TO 330
320   FBIG=FC
      XBIG=XC
      IF(FA.LE.FB) GO TO 315
      GO TO 307
330   ABEROR=ABS(XSTAR-XSMALL)
      IF(ABEROR.LE.XACCUR) GO TO 470
      IF(FSTAR.LT.FBIG) GO TO 350
      XSTAR=XSMALL
      GO TO 470
350   IF(FBIG.NE.FB) GO TO 370
      GO TO 470
370   IF(FBIG.EQ.FC) GO TO 380
      AB1=ABS(XC-XB)-ABS(XC-XSTAR)
      IF(AB1) 390,400,400
380   AB1=ABS(XA-XB)-ABS(XA-XSTAR)
      IF(AB1) 400,390,390
390   XC=XB
      FC=FB
      XB=XSTAR
      FB=FSTAR
      GO TO 410
400   XA=XB
      FA=FB
      XB=XSTAR
      FB=FSTAR
410   ANUM=(XB**2-XC**2)*FA+(XC**2-XA**2)*FB+(XA**2-XB**2)*FC
      DENOM=(XB-XC)*FA+(XC-XA)*FB+(XA-XB)*FC
      XSTAR=(ANUM/DENOM)/2.
      IF(XSTAR.LT.0.) XSTAR=PREOPT
      IF(L-2.GE.0.AND.XSTAR/T.LT.0.001) XSTAR=PREOPT
      IF(L-2) 430,440,450
430   FSTAR=ERROR(IN,XSTAR,ST1,ST2)
      SK=XSTAR
      IF(FSTAR.GE.FREOPT) SK=PREOPT
      GO TO 460
440   FSTAR=ERROR(IN,SK,XSTAR,ST2)
      ST1=XSTAR
      IF(FSTAR.GE.FREOPT) ST1=PREOPT
      GO TO 460
```

```
 450    FSTAR=ERROR(IN,SK,ST1,XSTAR)
        ST2=XSTAR
        IF(FSTAR.GE.FREOPT) ST2=PREOPT
 460    IF(SK.EQ.PREOPT) GO TO 470
        IF(ST1.EQ.PREOPT) GO TO 470
        IF(ST2.EQ.PREOPT) GO TO 470
        GO TO 255
 470    IF(L-2) 212,231,480
 480    RETURN
        END

C       THIS SUBPROGRAM CALCULATES "ERROR" WHICH IS A MEASURE OF
C       HOW WELL THE VALUES OF K,T1, AND T2 FIT THE DATA.
C       XP(I)= THE PREDICTED PROCESS OUTPUT
        FUNCTION ERROR(IN,SK,ST1,ST2)
        REAL K,K2
        COMMON X(350),U1(350),U2(350),U3(350),U4(350),XP(350),F(350)
        COMMON INUM, T, N, J, HGHT, U5(350),ERR(350),U6(350),U7(350)
        COMMON XAOPT (6) , XBOPT (6) , XCOPT (6) , XSOPT(6)
        COMMON FAOPT (6) , FBOPT (6) , FCOPT (6) , FSOPT(6)
        COMMON XMM2, XMM1, XM, XMP1, FMM2, FMM1, FM
        COMMON FMP1, DK,K,T1,T2                   ,K2, T21,T22
        COMMON PREOPT ,FREOPT
C       Calculate the ERROR for the current values of K,K2, ...
C       as per your system equations. Here we are working out a
C       example for illustration . ERROR will depend upon input data,
C       current values of K,K2 etc. and your system equations.
C        ERROR=0.0
         SUM = 0.0
         IF (IN-1) 510 ,510 ,560
 510      DO 550 I = 1,INUM

C        ------------------------------------------------------------
         XP(I) = T22*(U1(I)**SK)*((1/U4(I))**ST1)*(U2(I)**ST2)*
     1    ((1/U5(I))**K2)*(U3(I)**T21)*U6(I)
C        ------------------------------------------------------------

          ERR(I) = ABS ((X(I) - XP(I))/X(I))
C         ERR IS USER SPECIFIC AND CAN BE DEFINED IN VARIOUS WAYS.
          SUM = SUM + ERR(I)
 550      CONTINUE
          GO TO 590
 560      DO 580 I = 1,INUM

C        ------------------------------------------------------------
         XP(I) = ST2*(U1(I)**K)*((1/U4(I))**T1)*(U2(I)**T2)*
     1    ((1/U5(I))**SK)*(U3(I)**ST1)*U6(I)
C        ------------------------------------------------------------

          ERR(I) = ABS ((X(I) - XP(I))/X(I))
          SUM = SUM + ERR(I)
 580      CONTINUE
          GO TO 590
 590      ERROR = SUM
          WRITE(*,*)ERROR
          RETURN
          END
```

```
C      DSC SEARCH (CONTINUED)

C      THIS SUBROUTINE FINDS "XSOPT(L)" WHICH IS THE PSEUDO-OPTIMUM
C      VALUE OF THE PARTICULAR VARIABLE BEING SEARCHED.
C      L=1   K(SEARCHED VARIABLE)
C      L=2   T1(SEARCHED VARIABLE)
C      L=3   T2(SEARCHED VARIABLE)

       SUBROUTINE DSC(IN,L)
       REAL K,K2
       COMMON X(350),U1(350),U2(350),U3(350),U4(350),XP(350),F(350)
       COMMON INUM, T, N, J, HGHT, U5(350),ERR(350),U6(350),U7(350)
       COMMON XAOPT (6) , XBOPT (6) , XCOPT (6) , XSOPT(6)
       COMMON FAOPT (6) , FBOPT (6) , FCOPT (6) , FSOPT(6)
       COMMON XMM2, XMM1, XM, XMP1, FMM2, FMM1, FM
       COMMON FMP1, DK,K,T1,T2                    ,K2, T21,T22
       COMMON PREOPT ,FREOPT

       IF(FMM1.GE.FMP1) GO TO 600
       XAOPT(L)=XMM2
       XBOPT(L)=XMM1
       XCOPT(L)=XMP1
       FAOPT(L)=FMM2
       FBOPT(L)=FMM1
       FCOPT(L)=FMP1
       GO TO 610
600    XAOPT(L)=XMM1
       XBOPT(L)=XMP1
       XCOPT(L)=XM
       FAOPT(L)=FMM1
       FBOPT(L)=FMP1
       FCOPT(L)=FM
610    ANUM=(DK/2.)*(FAOPT(L)-FCOPT(L))
       WRITE (*,612)
612    FORMAT ('0ANUM =')
       WRITE (*,*) ANUM
       DENOM=2.*(FAOPT(L)-2.*FBOPT(L)+FCOPT(L))
       WRITE (*,614)
614    FORMAT ('0DENOM = ')
       WRITE (*,* ) DENOM
       IF(DENOM.EQ.0.0)DENOM=1E-8
       XSOPT(L)=XBOPT(L)+(ANUM/DENOM)
       WRITE (*,616)
616    FORMAT ('0XSOPT =')
       WRITE (*,* ) XSOPT(L)
       IF(XSOPT(L).LE.0.) GO TO 670
       IF(L-2.GE.0.AND.XSOPT(L)/T.LT.0.001) GO TO 670
       IF(L-2) 630,640,650
630    IF(IN-1)631,631,632
631    FSOPT(L)=ERROR(IN,XSOPT(L),T1,T2)
       GO TO 660
632    FSOPT(L)=ERROR(IN,XSOPT(L),T21,T22)
       GO TO 660
640    IF(IN-1)641,641,642
       WRITE (*,616)
641    FSOPT(L)=ERROR(IN,K,XSOPT(L),T2)
       GO TO 660
```

```
642   FSOPT(L)=ERROR(IN,K2,XSOPT(L),T22)
      GO TO 660
650   IF(IN-1)651,651,652
651   FSOPT(L)=ERROR(IN,K,T1,XSOPT(L))
      GO TO 660
652   FSOPT(L)=ERROR(IN,K2,T21,XSOPT(L))
660   IF(FSOPT(L).LE.FREOPT) GO TO 690
670   XSOPT(L)=PREOPT
690   RETURN
      END
```

APPENDIX - 2.2

COMPUTER PROGRAM FOR MODIFIED SIMPLEX ALGORITHM OF NELDER AND MEAD.

```
05    REM ------------------------------------------------------------
10    REM  OPTIMIZATION BY THE MODIFIED SIMPLEX METHOD OF NELDER & MEAD
20    REM      Adopted from  Valko,.P and Vajda,.S, Ref (143).
30    REM   MODIFIED FOR RANDOM AND SEQUENTIAL GENERATION OF VERTICES
35    REM                              by
40    REM                         R.SENGUPTA
50    REM   CHEMICAL ENGINEERING-DEPARTMENT , FACULTY OF TECH. & ENGG.
60    REM            M.S. UNIVERSITY OF BARODA,  BARODA.
65    REM ------------------------------------------------------------

67    REM  The main body of the program starts from subroutine 3400
68    REM  Function evaluation is done in subroutine 900
69    REM  Centroid calculations are highlighted in subroutine 1200

70    REM  NOTATION

80    REM  INPUT
81    REM  N     Number of Variables
82    REM  S (N+N, N)    Initial Simplex Coordinates
83    REM  EP    Threshold of Norm of the Centroid correction
84    REM  IM    Maximum Number of Iterations

85    REM  OUTPUT
86    REM  ER  STATUS  FLAG
87    REM            0     SUCCESSFUL SEARCH
88    REM            1     Unadmissible Point in Initial Simplex
89    REM            2     Threshold not attained
90    REM  X(N)  Estimate of the minimum point
95    REM  F     Function value of Final Estimate

100   REM FUNCTION MINIMISATION BY SIMPLEX METHOD
110   REM   PROBLEM SIZE

120   INPUT "PROBLEM SIZE  N=";N : LPRINT "N = "; N
130   DIM  X(N), S(N+N, N), R(3,N+N)

140   REM CONTROL PARAMETERS
150   EP = 0.0005  : INPUT "IM = ";IM    : LPRINT "IM = "; IM

160   DIM  RE(200),WE(200),FR(200),DST(200),EAR(200),DAR(200),AE(200)
170   DIM  DIFF(200),SUMF(200),F(200)

180   INPUT "NAME OF THE INPUT DATA FILE (XXXXXXXNNNN)";ID$
190   LPRINT "INPUT DATA FILE IS "; ID$
200   OPEN ID$ FOR INPUT AS #1
210   INPUT "CALCULATIONS BEGIN FROM B=";B
220   INPUT "CALCULATIONS UPTO E =";E
```

```
230   FOR I = B TO E
240   INPUT #1,RE(I),WE(I),FR(I),DST(I),DAR(I)
250   NEXT I
260   CLOSE 1
270   REM INITIAL SIMPLEX
280   PRINT "ENTER INITIAL GUESS VALUES OF COEFFICIENTS IN THE DESIRED
      DATA FILE"
290   PRINT " X2 TO X5 ARE INDICES FOR NUMBERS & X1 IS CONSTANT"
300   FOR J = 1 TO N+N
310   INPUT " X(1),X(2),X(3),X(4),X(5) ";X(1),X(2),X(3),X(4),X(5)
320   FOR I = 1 TO N
330   S(J,I) = X(I)
340   NEXT I
350   NEXT J
360   P$ = " #.#####^^^^  "
370   V$ = STRING$ (75, "-")

380   LPRINT "SIMPLEX METHOD OF NELDER AND MEAD" :LPRINT
390   LPRINT V$
400   GOSUB 3400
410   LPRINT : LPRINT "MINIMUM";
420   LPRINT TAB(10);"X(1)="X(1);TAB(25);"X(2)=";X(2);TAB(40);
      "X3=";X(3);TAB(55);"X(4)=";X(4); TAB(70);"X(5)=";X(5)
430   LPRINT : LPRINT V$ :LPRINT
440   STOP

900   REM  SUBROUTINE FOR  FUNCTION EVALUATION
905   M = 0 :F =0
910   FOR I = B TO E
920   AE(I) = X(5)*((RE(I)^X(1))*(WE(I)^X(2))*(FR(I)^X(3))*
      ((1/DST(I))^X(4)))
930   DIFF(I) = ABS((DAR(I) - AE(I))/DAR(I))*100
940   M = M + DIFF(I)
950   NEXT I
952   F = (M/ E)
953   PRINT USING P$ ; X(1),X(2),X(3),X(4),F
960   RETURN

1200  REM SUBROUTINE PRINTS CENTROID F
1220  PRINT CENTROID :  LPRINT "CENTROID CALCULATIONS"
1225  gosub 900
1226  PRINT "IT = "; IT
1250  RETURN


3400  REM MINIMISATION OF A FUNCTION OF SEVERAL VARIABLES
3420  REM          BY  NELDER AND MEAD METHOD
3440  REM  INITIAL SIMPLEX EVALUATION
3442  ER = 0
3444  FOR JN = 1 TO N+N
3446  FOR I = 1 TO N : X(I) = S(JN,I) : NEXT I
3448  GOSUB 900 : IF ER <> 0 THEN ER = 1 :GOTO 3562
3450  R(3,JN) = F
3452  LPRINT USING P$; X(1),X(2),X(3),X(4),F
3453  NEXT JN
```

```
3454   REM ITERATION (BEST:KN,WORST: N1,NEXT WORST: N2)
3456   FOR IT = 1 TO IM
3458   F=R(3,N+N): FK=F: KN=N+N: F1=F: N1=N+N: F2=-1E+20
3460   FOR J = 1 TO (N+N-1)
3462   F = R(3,J)
3464   IF F<FK THEN FK = F: KN=J: GOTO 3470
3466   IF F>F2 AND F<=F1 THEN F2 = F: N2 = J: GOTO 3470
3468   IF F>F2 THEN F2 = F1 :N2 = N1 :F1 = F: N1 = J
3470   NEXT J  : PRINT "N1 = "; N1

3472   REM    CENTROID
3473   PRINT "CENTROID"

3474   FOR I = 1 TO N
3476   R(2,I) = R(1,I) : R(1,I) = 0
3478   FOR J = 1 TO N+N
3480   IF J <> N1 THEN R(1,I) = R(1,I)+(S(J,I)/(N+N-1))
3482   NEXT J
3483   PRINT "R(1,I) = "; R(1,I)
3484   NEXT I
3485   FOR I = 1 TO N : X(I) = R(1,I) : NEXT I : GOSUB 1200

3486   REM   REFLECTION
3487   PRINT "REFLECTION"

3488   FOR I = 1 TO N : X(I) = 2*R(1,I)- S(N1,I) : NEXT I
3490   ER = 0 : GOSUB 900 : IF ER <> 0 THEN 3528
3492   IF F>FK THEN 3508
3494   REM SUCESSFUL STEP
3495   PRINT "SUCESSFUL STEP"
3496   FOR I = 1 TO N : S(N1,I) = X(I) : NEXT I : R(3,N1) = F
3497   FK = F : KN = N1

3498   REM   EXPANSION
3499   PRINT "EXPANSION"

3500   FOR I = 1 TO N : X(I)= 2 * X(I)-R(1,I) : NEXT I
3502   ER = 0 : GOSUB 900 : IF ER <> 0 THEN 3528
3504   IF F <= FK THEN FOR I = 1 TO N: S(N1,I) = X(I)
3505   NEXT I : R(3,N1) = F
3506   GOTO 3548

3508   REM   NEUTRAL
3509   PRINT "NEUTRAL"

3510   IF F>= F2 THEN 3514
3512   FOR I = 1 TO N : S(N1,I) = X(I) : NEXT I
3513   R(3,N1) = F : GOTO 3548

3514   REM   UNSUCCESSFUL STEP
3515   PRINT "UNSUCESSFUL STEP"

3516   IF F < F1 THEN FOR I = 1 TO N : S(N1,I) = X(I)
3517   NEXT I : R(3,N1) = F : F
```

```
3518   REM   CONTRACTION
3519   PRINT "CONTRACTION"

3520   FOR I = 1 TO N : X(I) = (R(1,I)+S(N1,I))/2 : NEXT I
3522   ER = 0 : GOSUB 900 : IF ER <> 0 THEN 3528
3524   IF F < FK THEN KN = N1 : FK = F
3526   IF F < F1 THEN FOR I = 1 TO N : S(N1,I) = X(I) : NEXT I
3527   R(3,N1)=F : GOTO 3548

3528   REM  REDUCING SIMPLEX SIZE  :LPRINT "REDUCTION"
3529   PRINT "REDUCTION"

3530   FOR J = 1 TO N+N
3532   IF J<>KN THEN FOR I = 1 TO N : S(J,I) = (S(J,I) + S(KN,I))/2
3533   NEXT I
3534   NEXT J
3536   FOR J = 1 TO N+N
3538   IF J = KN THEN 3546
3540   FOR I = 1 TO N : X(I) = S(J,I) : NEXT I
3542   GOSUB 900 : IF ER <> 0 THEN ER = 2 : GOTO 3562
3544   R(3,J) = F : IF F < FK THEN FK = F :KN = J
3546   NEXT J
3548   SX = 0 : SK = 0 : F = FK
3550   FOR I = 1 TO N
3552   D = R(1,I)-R(2,I) : SX = SX + D*D : X(I) = S(KN,I)
3553   D = X(I) - R(1,I) : SK = SK + D*D
3554   NEXT I
3556   IF SQR(SX) <= EP AND SQR(SK) < EP THEN 3562
3558   NEXT IT
3560   ER = 2
3562   RETURN
```