Riv. Mat. Univ. Parma (4) 3 (1977), 119-130

C. SUTTI and R. GARDANI (*)

An implementation of a finite algorithm for finding polynomial minima. (**)

1. - Introduction.

In this paper the authors consider the problem of computing all minima of a real polynomial $P(x) = \sum_{i=0}^{n} a_i x^i$. In order to solve this problem they present a code in Fortran 4 of the Sutti's algorithm [**8**]$_1$, with its programming specification.

The main purpose of implementing the method in [**8**]$_1$ is its use in the optimization theme. In fact this method is able to solve line-searches for global minimization of nonconvex functions [**2**]. Indeed the best known algorithms [**7**], [**3**] which were devised to this end yield some practical application difficulties. For example, they assume that the Lipschitz constant L of the objective function is known, whereas L is seldom available. However, even when L can be deduced, as for a polynomial objective function in a fixed interval, its rough estimation reduces the efficiency of the algorithms (see section 4). Moreover all these methods need an assumption of the initial step length and a choice of the stopping criterion. Further, the technique [**8**]$_1$ seems to overcome a major drawback connected with the use of deflation; in fact the former technique is numerically stable, whereas in the latter the large level of cumulative error very often conceals the true solution [**10**].

Then the subroutines provided make possible to implement some random search algorithms for global optimization of n-dimensional polynomial functions. It had been theoretically proved, in fact, that an algorithm, which performs exact global minimizations along random search vectors, converges in probability: but it had been claimed that a such algorithm was generally not implementable [**5**].

(*) Indirizzo degli Autori: Istituto di Matematica, Università, 43100 Parma, Italia.
(**) C. Sutti ha eseguito il lavoro nell'ambito del G.N.I.M. (C.N.R.). — Ricevuto: 20-VI-1976.

In section **2** we present the subroutines with the references to the algorithms used and in section **3** we give their listings. In section **4** we give the results of the numerical experiences which we have had on a C.D.C. 7600.

2. - Subroutines description.

The program consists of five subroutines: MINIMI (the main one), MONO1, GOLMIN, DEV, CAL.

Subroutine MINIMI implements the global polynomial minimization technique which is described in [**8**]$_1$. This technique computes the derivatives of $P(x)$, $P^{(n-i)}(x)$, $2 \leqslant i \leqslant (n-1)$, and, by the algorithm [**9**] for minimizing bimodal functions, determines the unimodality ranges for $P^{(n-4)}(x)$. In fact $P^{(n-4)}(x)$ (or $-P^{(n-4)}(x)$) is bimodal with $P^{(n-2)}(x)$ (or $-P^{(n-2)}(x)$) unimodal on each interval. Then, by proceeding in the same way on the subintervals obtained for $P^{(n-i)}(x)$, $i \geqslant 6$, all the unimodality ranges and all the minima of $P(x)$ are identified by a known number of iterations.

In the presented numerical implementation of technique [**8**]$_1$, some minor variations are introduced in order to decrease the function calls. Namely, in MINIMI, at the first iteration, the Cardano's formula [**1**] is used to determine the roots of the equation $P^{(n-3)}(x) = 0$. Further, a cyclic test is introduced in order to provide all and only the minima and the largest unimodality ranges of $P(x)$. Subroutine MINIMI calls MONO1, DEV and CAL.

Subroutine DEV computes the coefficients of the successive derivatives of $P(x)$: the formula used is the analytical one.

Subroutine CAL computes the value $Q = \sum_{i=0}^{j} A_i x_0^i$, $j \leqslant n$. The iterative formula which we have applied is the following

$$Q = [((A_j x_0 + A_{j-1}) x_0 + A_{j-2}) x_0 + \ldots + A_1] x_0 + A_0 \,.$$

Subroutine MONO1 implements the algorithm as in [**9**], after the suitable adaptation which is described in [**8**]$_1$. Subroutine MONO1 calls the subroutines GOLMIN and CAL.

The former minimizes one-dimensional unimodal functions. To this end we have implemented the golden section method [**8**]$_2$. GOLMIN also calls CAL.

To call the subroutine MINIMI from some application program, only the input parameters of MINIMI, as specified at the beginning of the subroutine, need be defined.

In order to describe all these subroutines, suitable comments are included in their listings.

3. - Subroutins listings.

```
      SUBROUTINE MINIMI(N,A,COPOL,P1,P2,MIN,PMIN,INDVAL,XMIN,MINAX,L,CO,
     *CO1,COA,COA1,COEF)
C TO MINIMIZE POLYNOMIAL ONEDIMENSIONAL FUNCTIONS BY SUTTI'S ALGORITHM
C
C    INPUT AND OUTPUT PARAMETERS
C    N= DEGREE OF THE POLYNOMIAL
C    A= VECTOR OF THE EXTREMES  OF THE UNIMODALITY RANGES. IN INPUT FIRST
C       TWO COMPONENTS CONTAIN THE EXTREMES  OF THE INITIAL INTERVAL,
C       OTHER COMPONENTS ARE UNDEFINED. IN OUTPUT (L+1)  COMPONENTS ARE
C       DEFINED
C    COPOL= VECTOR OF THE N+1 COEFFICIENTS OF THE POLYNOMIAL (DECREASIN-
C           GLY ORDERED)
C    P1= PRECISION IN MINIMI AND MONO1
C    P2= PRECISION IN GOLMIN
C
C    OUTPUT PARAMETERS
C    MIN= VECTOR OF THE MINIMA'S ABSCISSAS
C    PMIN= VECTOR OF THE MINIMA'S VALUES
C    INDVAL= NUMBER OF FUNCTION EVALUATIONS
C    XMIN= ABSCISSA OF THE ABSOLUTE MINIMUM
C    MINAX= VALUE OF THE ABSOLUTE MINIMUM
C    L= NUMBER OF UNIMODALITY RANGES
C    COEF= VECTOR OF THE COEFFICIENTS OF THE POLYNOMIAL AND ITS FIRST
C          (N-2) DERIVATIVES
C
C    CONSTRAINTS
C    N^3
C
C    DIMENSION OF AUXILIARY VECTORS: CO,CO1,COA,CO1A =N+1 TO BE ASSIGNED
C    IN THE CALLING PROGRAM
C    DIMENSION OF THE VECTORS MIN, PMIN MUST BE ASSIGNED =N IN THE CALLING
C    PROGRAM
C    DIMENSION OF THE VECTOR COEF MUST BE ASSIGNED =(N+4)*N/2 IN THE
C    CALLING PROGRAM
C
C    INTERNAL VARIABLES
C    MI= N, FOR N EVEN
C    MI= ((N+1)/2)*2, FOR N ODD
C    I= ITERATION INDEX
C    M= INTERVAL INDEX
C    Q= 0, FOR N EVEN
C    Q=1, FOR N ODD
C    K=0 IF THE POLYNOMIAL IS NOT INCREASING IN THE FIRST EXTREMAL OF
C    THE INITIAL INTERVAL
C    K=1, IF THE POLYNOMIAL IS INCREASING IN THE FIRST EXTREMAL OF THE
C    INITIAL INTERVAL
C
      DIMENSION COPOL(1),MIN(1),A(1),CO(1),CO1(1),COA(1),COA1(1),PMIN(1)
     *,COEF(1)
      INTEGER Q,R,S,T,S1,S2,T1,T2,W,U,PI1,PI2
      REAL MIN,MIN1,MIN2,MINAX
      IDD(N,ID)=(N+1)*ID-((ID-1)*ID)/2+1
      INDVAL=0
      MI=(N+1)/2*2
      CALL DEV(N,COPOL,COEF)
      I=1
      M=1
      NI=N+1
      Q=0
      K=0
C
C    COMPUTATION OF THE REAL ROOTS BY CARDANO'S FORMULA FOR THIRT DEGREE
C    EQUATIONS
C
      IA=IDD(N,N-3)
      AI=COEF(IA)
      BI=COEF(IA+1)
      CI=COEF(IA+2)
      DI=COEF(IA+3)
      DO 398 J=1,3
      IJ=IDD(N,N-2)+J-1
  398 COA(J)=COEF(IJ)
C
C    ONE ROOT
C
      PI=CI/AI-BI**2/(3.*AI**2)
      QI=DI/AI-BI*CI/(3.*AI**2)+2.*BI**3/(27.*AI**3)
      DELTA=4.*PI**3+27.*QI**2
      IF(DELTA.LE.0)GO TO 400
      ROOT=SQRT(DELTA/108.)
      AA=-QI/2.+ROOT
      BA=-QI/2.-ROOT
      Y1=SIGN(ABS(AA)**(1./3.),AA)+SIGN(ABS(BA)**(1./3.),BA)-BI/(3.*AI)
      GO TO 405
C
C    THREE ROOTS
C
  400 PHI=ASIN((SQRT(27.)*QI)/(2.*PI*SQRT(-PI)))/3.
      UI=SQRT(-PI/3.)*2.
```

```
      VI=-BI/(3.*AI)
      SI=SIN(PHI)
      CS=COS(PHI)*SQRT(3.)/2.
      Y1=UI*(CS+0.5*SI)+VI
      Y2=-UI*SI+VI
      Y3=-UI*(CS-0.5*SI)+VI
      IF(ABS(Y1-Y2).LE.P1)GO TO 410
      IF(ABS(Y1-Y3).LE.P1)GO TO 415
      IF(ABS(Y2-Y3).GT.P1)GO TO 450
  405 IF(A(M).LT.Y1.AND.Y1.LT.A(M+1))GO TO 430
      GO TO 435
  410 Y1=Y3
      GO TO 405
  415 Y1=Y2
      GO TO 405
C
C   ONE ROOT IN THE INITIAL INTERVAL
C
  430 CALL CAL(COA,2,Y1,INDVAL,PCAL)
      IF(PCAL.GT.0)GO TO 432
      A(M+2)=A(M+1)
      A(M+1)=Y1
      MIN(M)=A(M)
      MIN(M+1)=A(M+2)
      L=L+1
      GO TO 237
  432 MIN(M)=Y1
      GO TO 237
C
C   NO ROOT IN THE INITIAL INTERVAL
C
  435 DO 437 J=1,5
      IJ=IDD(N,N-4)+J-1
  437 COA1(J)=COEF(IJ)
      CALL CAL(COA1,4,A(M),INDVAL,PCAL)
      PM=PCAL
      CALL CAL(COA1,4,A(M+1),INDVAL,PCAL)
      PM1=PCAL
      IF(PM.LT.PM1)GO TO 440
      MIN(M)=A(M+1)
      GO TO 237
  440 MIN(M)=A(M)
      GO TO 237
C
C   ORDERING OF THREE REAL ROOTS
C
  450 IF(Y1.LT.Y2)GO TO 451
      C=Y1
      Y1=Y2
      Y2=C
  451 IF(Y1.LT.Y3)GO TO 452
      C=Y1
      Y1=Y3
      Y3=C
  452 IF(Y2.LT.Y3)GO TO 453
      C=Y2
      Y2=Y3
      Y3=C
  453 IF(A(M).LT.Y1.AND.Y1.LT.A(M+1))GO TO 470
      IF(A(M).LT.Y3.AND.Y3.LT.A(M+1))GO TO 460
      IF(A(M).LT.Y2.AND.Y2.LT.A(M+1))GO TO 465
      GO TO 435
  460 IF(A(M).LT.Y2.AND.Y2.LT.A(M+1))GO TO 463
      Y1=Y3
      GO TO 430
  463 R1=Y2
      R2=Y3
      GO TO 500
  465 Y1=Y2
      GO TO 430
  470 IF(A(M).LT.Y3.AND.Y3.LT.A(M+1))GO TO 480
      IF(A(M).LT.Y2.AND.Y2.LT.A(M+1))GO TO 475
      GO TO 430
  475 R1=Y1
      R2=Y2
      GO TO 500
C
C   THREE ROOTS IN THE INITIAL INTERVAL
C
  480 CALL CAL(COA,2,Y2,INDVAL,PCAL)
      IF(PCAL.GT.0)GO TO 485
      MIN(M)=Y1
      MIN(M+1)=Y3
      A(M+2)=A(M+1)
      A(M+1)=Y2
      L=L+1
      GO TO 237
  485 MIN(M)=A(M)
      MIN(M+1)=Y2
      MIN(M+2)=A(M+1)
      A(M+3)=A(M+1)
      A(M+1)=Y1
      A(M+2)=Y3
```

```
      L=L+2
      GO TO 237
C
C   TWO ROOTS IN THE INITIAL INTERVAL
C
  500 CALL CAL(COA,2,R1,INDVAL,PCAL)
      IF(PCAL.GT.0)GO TO 505
      MIN(M)=A(M)
      MIN(M+1)=R2
      A(M+2)=A(M+1)
      A(M+1)=R1
      L=L+1
      GO TO 237
  505 MIN(M)=R1
      MIN(M+1)=A(M+1)
      A(M+2)=A(M+1)
      A(M+1)=R2
      L=L+1
      GO TO 237
    3 K1=N-2*I+Q
      K2=N-K1+2
      DO 5 J=1,K2
      IJ=IDD(N,K1-1)+J-1
    5 CO(J)=COEF(IJ)
      EPS=2*P2
      CALL CAL(CO,2*I+1-Q,A(M)+EPS,INDVAL,PCAL)
      IF(PCAL.LT.0)GO TO 15
      GO TO 25
   15 IF(Q.EQ.0)GO TO 85
   17 PI1=N-2*I+1+Q
      PI2=N-PI1+2
      DO 20 J=1,PI2
      IJ=IDD(N,PI1-1)+J-1
   20 COA(J)=-1*COEF(IJ)
      CALL GOLMIN(COA,2*I-Q,A(M),A(M+1),INDVAL,PGOL,P2)
      PMAX=PGOL
      GO TO 40
   25 K=1
C
C   CHANGE OF THE SIGN OF THE COEFFICIENTS
C
      INDIK=1
   30 DO 35 S=1,3
      W=N-2*I-S+2+Q
      U=N-W+2
      DO 35 J=1,U
      IJ=IDD(N,W-1)+J-1
   35 COEF(IJ)=-1*COEF(IJ)
      IF(INDIK.EQ.2)GO TO 200
      IF(Q.EQ.0)GO TO 37
      GO TO 17
   37 PMAX=MIN(M)
   40 IF(ABS(PMAX-A(M)).LE.P1)GO TO 45
      IF(ABS(PMAX-A(M+1)).LE.P1)GO TO 50
      GO TO 55
   45 MIN(M)=A(M+1)
      GO TO 85
   50 MIN(M)=A(M)
      GO TO 85
   55 IL=1
C
C   REORDERING OF MINIMA AND EXTREMALS
C
   56 L=L+1
      S=L-M
      DO 60 R=1,S
      S1=L+2-R
      S2=L+1-R
   60 A(S1)=A(S2)
      T=L-M-1
      IF(T.EQ.0)GO TO 70
      DO 65 R=1,T
      T1=L+1-P
      T2=L-R
   65 MIN(T1)=MIN(T2)
   70 A(M+1)=PMAX
      IF(IL.EQ.2)GO TO 210
      MIN(M)=A(M)
      IF(K.EQ.1)GO TO 75
      MIN(M+1)=A(M+2)
      GO TO 85
   75 MIN(M+1)=A(M+1)
   85 D2MIN=MIN(M)
      CALL MONO1(A(M),A(M+1),D2MIN,PMAX,I,MIN1,MIN2,N,COEF,INDVAL,Q,P1,P
     *2,CO,CO1,COA,COA1)
      IF(K.EQ.1)GO TO 95
      GO TO 200
   95 MIN2=PMAX
      PMAX=MIN1
      MIN1=A(M)
      K=0
      INDIK=2
      GO TO 30
  200 IF(ABS(PMAX-A(M+1)).LE.P1)GO TO 220
```

```
      IL=2
      GO TO 56
  210 MIN(M)=MIN1
      MIN(M+1)=MIN2
      M=M+1
      GO TO 225
  220 MIN(M)=MIN1
  225 IF(M.LT.L)GO TO 230
      GO TO 235
  230 M=M+1
      GO TO 3
  237 IF(I.LT.MI/2-2)GO TO 240
      IF(I.LT.MI/2-1)GO TO 245
      GO TO 250
  240 I=I+1
      M=1
      GO TO 3
C
C   FINAL COMPARISON, FOR M ODD
C
  245 IF(N.EQ.MI)GO TO 240
      Q=1
      GO TO 240
C
C   CHOICE OF THE TRUE MINIMA
C
  235 IF(L.EQ.1)GO TO 237
      J=1
  265 M=J+1
  270 IF(ABS(MIN(J)-A(M)).LE.P1)GO TO 295
      IF(J.EQ.1)GO TO 275
      GO TO 280
  275 J=J+1
      IF(J.LT.L)GO TO 265
      GO TO 335
  280 IF(ABS(MIN(J)-A(J)).LE.P1)GO TO 285
      GO TO 275
  285 L=L-1
      DO 290 R=J,L
      MIN(R)=MIN(R+1)
  290 A(R)=A(R+1)
      A(L+1)=A(L+2)
      IF(J.LT.L)GO TO 270
      GO TO 335
  295 IF(ABS(MIN(M)-A(M)).LE.P1)GO TO 305
      L=L-1
      DO 300 IN=J,L
      MIN(IN)=MIN(IN+1)
  300 A(IN+1)=A(IN+2)
      IF(J.LT.L)GO TO 270
      GO TO 335
  305 L=L-1
      MIN(J)=A(M)
      IF(M.GT.L)GO TO 320
      DO 310 IN=M,L
  310 MIN(IN)=MIN(IN+1)
      LU=L+1
      DO 315 R=M,LU
  315 A(R)=A(R+1)
      GO TO 275
  320 A(L+1)=A(L+2)
      GO TO 275
  335 IF(ABS(MIN(J)-A(J)).GT.P1)GO TO 237
      L=L-1
      GO TO 237
C
C   COMPUTATION OF PMIN
C
  250 DO 260 M=1,L
      CALL CAL(COPOL,N,MIN(M),INDVAL,PCAL)
  260 PMIN(M)=PCAL
C
C   CHOICE OF MINAX
C
      MINAX=PMIN(1)
      XMIN=MIN(1)
      IF(L.EQ.1)GO TO 346
      DO 345 J=2,L
      IF(MINAX.GT.PMIN(J))GO TO 340
      GO TO 345
  340 MINAX=PMIN(J)
      XMIN=MIN(J)
  345 CONTINUE
  346 RETURN
      END
      SUBROUTINE MONO1(A,B,D2MIN,PMAX,I,MIN1,MIN2,N,COEF,INDVAL,Q,P,P2,C
     *01,C01A,C02,C02A)
C TO MINIMIZE BIMODAL FUNCTIONS BY SUTTI'S-TRABATTONI'S- BRUGHIERA'S
C   METHOD
C
C   INPUT AND OUTPUT PARAMETERS
C   A= FIRST EXTREMAL OF THE INTERVAL
C   B= SECOND EXTREMAL OF THE INTERVAL
```

```
C     D2MIN= ABSCISSA OF THE MINIMUM OF THE SECOND DERIVATIVE
C     P=P1, PRECISION FOR THE COMPARISONS
C
C     OUTPUT PARAMETERS
C     MIN1= THE FIRST MINIMUM ON (A(M),A(M+1))
C     MIN2= THE SECOND  MINIMUM ON (A(M),A7M+1))
C     PMAX= SEPARATING POINT THE TWO UNIMODALITY RANGES
C
      DIMENSION CO1(1),CO1A(1),CO2(1),CO2A(1),COEF(1)
      REAL MIN1,MIN2
      INTEGER Q
      IDD(N,ID)=(N+1)*ID-((ID-1)*ID)/2+1
      M=N-2*I-1+Q
      K=N-M+2
      KI=K-1
      DO 1 J=1,K
      MI=IDD(N,M-1)+J-1
      CO1(J)=COEF(MI)
    1 CO1A(J)=-1*CO1(J)
      DO 2 J=1,KI
      MI=IDD(N,M)+J-1
      CO2(J)=COEF(MI)
    2 CO2A(J)=-1*CO2(J)
      KK=2*I+2-Q
      KKK=2*I+1-Q
      CALL GOLMIN(CO1,KK,A,B,INDVAL,PGOL,P2)
      AM1=PGOL
      IF(ABS(AM1-B).GT.P)GO TO 6
      IF(ABS(D2MIN-A).LE.P)GO TO 9
      CALL GOLMIN(CO2A,KKK,A,D2MIN,INDVAL,PGOL,P2)
      DPMAX=PGOL
      IF(ABS(DPMAX-A).LE.P)GO TO 9
      CALL GOLMIN(CO1,KK,A,DPMAX,INDVAL,PGOL,P2)
      AM3=PGOL
      IF(ABS(AM3-DPMAX).LE.P)GO TO 9
      CALL GOLMIN(CO1A,KK,AM3,AM1,INDVAL,PGOL,P2)
      AM4=PGOL
      GO TO 15
    6 CALL GOLMIN(CO1A,KK,A,AM1,INDVAL,PGOL,P2)
      AM2=PGOL
      IF(ABS(AM2-A).LE.P)GO TO 7
      CALL GOLMIN(CO1,KK,A,AM2,INDVAL,PGOL,P2)
      AM3=PGOL
      PMAX=AM2
      GO TO 16
    7 CALL GOLMIN(CO1A,KK,AM1,B,INDVAL,PGOL,P2)
      AM4=PGOL
      IF(ABS(AM4-B).GT.P)GO TO 8
      IF(ABS(D2MIN-A).LE.P.OR.ABS(D2MIN-B).LE.P.OR.ABS(D2MIN-AM1).LE.P)G
     1O TO 9
      IF(D2MIN.LT.AM1)GO TO 12
      CALL GOLMIN(CO1,KK,D2MIN,B,INDVAL,PGOL,P2)
      AM3=PGOL
      IF(ABS(AM3-D2MIN).GT.P)GO TO 10
      CALL GOLMIN(CO2,KKK,D2MIN,B,INDVAL,PGOL,P2)
      DPMIN=PGOL
      IF(ABS(DPMIN-B).LE.P)GO TO 9
      CALL GOLMIN(CO1,KK,DPMIN,B,INDVAL,PGOL,P2)
      AM3=PGOL
      IF(ABS(AM3-DPMIN).LE.P)GO TO 9
      MIN1=AM1
      MIN2=AM3
      CALL GOLMIN(CO1A,KK,AM1,AM3,INDVAL,PGOL,P2)
      AM4=PGOL
      PMAX=AM4
      GO TO 20
    8 CALL GOLMIN(CO1,KK,AM4,B,INDVAL,PGOL,P2)
      AM3=PGOL
   11 PMAX=AM4
      MIN2=AM3
      MIN1=AM1
      GO TO 20
    9 PMAX=B
      MIN1=AM1
      MIN2=AM1
      GO TO 20
   10 CALL GOLMIN(CO1A,KK,AM1,AM3,INDVAL,PGOL,P2)
      AM4=PGOL
      GO TO 11
   12 CALL GOLMIN(CO1,KK,A,D2MIN,INDVAL,PGOL,P2)
      AM3=PGOL
      IF(ABS(AM3-D2MIN).GT.P)GO TO 14
      CALL GOLMIN(CO2A,KKK,A,AM3,INDVAL,PGOL,P2)
      DPMAX=PGOL
      IF(ABS(DPMAX-A).LE.P)GO TO 9
      CALL GOLMIN(CO1,KK,A,DPMAX,INDVAL,PGOL,P2)
      AM3=PGOL
      IF(ABS(AM3-DPMAX).LE.P)GO TO 9
      MIN1=AM3
      MIN2=AM1
      GO TO 17
   14 CALL GOLMIN(CO1A,KK,AM3,AM1,INDVAL,PGOL,P2)
      AM4=PGOL
```

```
   15 PMAX=AM4
   16 MIN1=AM3
      MIN2=AM1
   17 CALL GOLMIN(CO1A,KK,AM3,AM1,INDVAL,PGOL,P2)
      PMAX=PGOL
   20 RETURN
      END
      SUBROUTINE CAL(A,N,X,INDVAL,PCAL)
C TO COMPUTE THE VALUE OF THE POLYNOMIAL IN A POINT
C
C   INPUT AND OUTPUT PARAMETERS
C   A= VECTOR OF THE ORDERED COEFFICIENTS OF THE POLYNOMIAL
C   X= POINT IN WHICH THE POLYNOMIAL'S VALUE MUST BE COMPUTED
C
C   OUTPUT PARAMETERS
C   PCAL= VALUE OF THE POLYNOMIAL
C
      DIMENSION A(1)
      PCAL=A(1)
      MU=N+1
      DO 80 I=2,MU
   80 PCAL=PCAL*X+A(I)
      INDVAL=INDVAL+1
      RETURN
      END
      SUBROUTINE DEV(N,COPOL,COEF)
C TO COMPUTE THE SUCCESSIVE COEFFICIENTS OF THE 1-ST, 2-ND,...,(N-2)-TH
C DERIVATIVES OF THE POLYNOMIAL
C
C   INPUT AND OUTPUT PARAMETERS
C   COEF= VECTOR OF THE COEFFICIENTS COMPUTED
C
      DIMENSION COPOL(1),COEF(1)
      IDD(N,ID)=(N+1)*ID-((ID-1)*ID)/2+1
      L=N+1
      DO 1 J=1,L
    1 COEF(J)=COPOL(J)
      KU=L+1
      M=N-2
      DO 2 ID=1,M
      K=N-ID+1
      KI=IDD(N,ID-1)
      DO 2 J=1,K
      KL=KI-1+J
      COEF(KU)=(N-J-ID+2)*COEF(KL)
    2 KU=KU+1
      RETURN
      END
      SUBROUTINE GOLMIN(CO,N,A1,B1,INDVAL,PGOL,P)
C TO COMPUTE THE MINIMUM OF UNIMODAL FUNCTIONS BY GOLDEN SECTION METHOD
C
C   INPUT AND OUTPUT PARAMETERS
C   P=P2
C   INPUT PARAMETERS
C   CO= COPOL
C   OUTPUT PARAMETERS
C   PGOL= ABSCISSA OF THE MINIMUM
C
      DIMENSION CO(1)
      REAL K1,K2
      A=A1
      B=B1
      IN=N+1
      T=1.61803
      I=2
    1 H=(B-A)/T
      GO TO(10,11,12)I
   10 K2=K1
      GO TO 15
   11 K2=A+H
   15 K1=B-H
      GO TO 17
   12 K1=K2
      K2=A+H
   17 CALL CAL(CO,N,K1,INDVAL,PCAL)
      F1=PCAL
      CALL CAL(CO,N,K2,INDVAL,PCAL)
      F2=PCAL
      IF(ABS(K2-K1).LE.P)GO TO 5
      IF(F1-F2.LE.0)GO TO 2
      A=K1
      I=3
      GO TO 1
    2 B=K2
      I=1
      GO TO 1
    5 IF(F2.LT.F1)GO TO 6
      PGOL=K1
      GO TO 16
    6 PGOL=K2
   16 CONTINUE
      RETURN
      END
```

4. - Computational results.

We ran the above program in single precision on a C.D.C. 7600 computer. At first we tested the following polynomials

$$P^5 = 12x^5 - 45x^4 - 20x^3 + 90x^2\,, \qquad P^6 = x^6 - 15x^4 + 27x^2 + 250\,.$$

Successively we considered some functions of several variables concerning the global minimization [**2**]. For these we have executed the line-searches starting from given initial points along the negative normalized gradients. These functions [**4**] are

$$F^1 = 100(x_1^2 - x_2)^2 + (x_1 - 1)^2\,, \qquad F^2 = 1/4 \sum_{k=1}^{5} [(x_k - 1)^4 - 8k^2(x_k - k)^2 + 2k^3 x_k]\,,$$

$$F^3 = 1/3x_1^6 - 2.1x_1^4 + 4x_1^2 + x_1x_2 + 4x_2^4 - 4x_2^2\,.$$

The initial points are respectively

$$x^1 = (-1.2, 1)\,, \qquad x^2 = (1, 1, 1, 1, 1)\,, \qquad x^3 = (2, 1.5)\,.$$

The one-dimensional polynomial functions to be minimized are then

$$PF^1 = (73.478)\alpha^4 - (445.73)\alpha^3 + (752.25)\alpha^2 - (232.87)\alpha + (24.2)\,,$$

$$PF^2 = (0.160\,84)\alpha^4 + (3.0074)\alpha^3 - (24.127)\alpha^2 - (451.64)\alpha + (967)\,,$$

$$PF^3 = (0.270\,24)\alpha^6 - (3.2429)\alpha^5 + (15.16)\alpha^4 - (32.76)\alpha^3 + (35.192)\alpha^2 - (26.197)\alpha + (1.7983)\,.$$

In table 4.1 we summarise some our test results.

TABLE 4.1

FUNCTION	INTERV	$P1$	$P2$	MIN	INDVAL	ITER	TIME
P^5	$-10,10$	$0.5 \cdot 10^{-1}$	10^{-3}	-10.0000*	403	2	14 M.S.
P^6	$-10,10$	$0.5 \cdot 10^{-1}$	10^{-3}	$-\ 3.0030$*	305	2	15 M.S.
PF^1	0,1	$0.5 \cdot 10^{-1}$	10^{-3}	0.18350	2	1	10 M.S.
PF^2	0,1	$0.5 \cdot 10^{-1}$	10^{-3}	1.0000	3	1	9 M.S.
PF^3	0,1	$0.5 \cdot 10^{-1}$	10^{-3}	0.99881	79	2	11 M.S.

FUNCTION is the name of the objective functions as above.

INTERV is the spanned interval. For the function PF^1, PF^2, PF^3 we present the results concerning the interval [0, 1], because this is useful for sake of comparison with different line-searches.

$P1$ is the precision by which the points are considered distinct or not in MINIMI and MONO1.

$P2$ is the precision in GOLMIN ($P1$ must be greater than $P2$). After several checks using different $P1$ and $P2$, it came out that the solution error only depends on $P2$ as in the usual line-searches $[\mathbf{8}]_2$. This agrees with the algorithm feature $[\mathbf{8}]_1$, which prevents a cumulative error, since different subintervals are spanned with different functions, in the successive iterations.

Also the rounding errors on the values assumed by the successive objective functions in the given points have been estimated. In fact, if the error e on $Q = P^{(n-i)}(x_0)$, $2 \leqslant i \leqslant n$, were increasing with $(n-i) = j$, the application of of algorithm $[\mathbf{8}]_1$ to polynomials of high degree would be unsuitable. But, it has been verified that the relative errors are of the same order by using the following formula [**6**]

$$|e| = 2^{-t} g_{i+1}(1 + 2^{-t}) j ,$$

where t is the number of digits of the computer used ($t = 15$ on a C.D.C 7600) and g_{i+1} is computed by the iterative formula

$$g_1 = 0 , \quad g_i = |x_0|\{g_{i-1} + |c_{i-1}|\} + |c_i| , \qquad i = 2, \ldots, (j+1) ,$$

with

$$c_1 = A_j , \quad c_i = c_{i-1} x_0 + A_{j-i+1} , \qquad i = 2, \ldots, (j-1) .$$

MIN is the global minimum. When the objective function presents more minima, the absolute one is indexed by a star. If the objective function presents more global minima, these ones can be individuated by comparing the values which the function assumes in the found minima. In fact, because the computational errors, the program individuates always one global minimum.

INDVAL is the number of evaluations of the polynomial and of its derivatives which have been required in the execution, in fact this is the usual criterion of efficiency in mathematical programming.

ITER is the number of the interactions done; this is what needed to verify the theory about the algorithm $[\mathbf{8}]_1$.

TIME is the execution time on a C.D.C 7600.

As for memory requirements, we mention that the five subroutines employ about $1.5K$ words. We note that the memory requirement is reduced if

GOLMIN, DEV and CAL are substituted by some other available routines. In fact these are often provided by the computer library.

In any case the memory which is required from the program subroutines must be increased by the dimensions of the parameters, as specified on the heading « constraints » in the listing. Thus the memory requirement is slightly more than $n^2/2$ words.

For comparison, we minimized the previous functions by the Shubert's method [7]. Note that, for a polynomial P on the interval $[0, l]$, we can assume as Lipschitz constant $L = \sum_{i=1}^{n} |a_i i| l^{i-1}$. So, for any function test, this value has been a priori computed, then it has been assigned among the input data of the program.

In table 4.2 we present the numerical results.

TABLE 4.2

FUNCTION	INTERV	PF	PX	MIN	INDVAL	TIME
P^5	— 10,10	10^{-2}	10^{-2}	— 10.0000	1401	6375 M.S.
P^6	— 10,10	10^{-2}	10^{-2}	2.9956	1028	2203 M.S.
PF^1	0,1	10^{-3}	10^{-3}	0.18409	215	121 M.S.
PF^2	0,1	10^{0}	10^{-3}	0.99973	6	8 M.S.
PF^3	0,1	10^{0}	10^{-3}	0.99991	93	26 M.S.

PF and PX are the precisions which have been used in the stopping criteria.

From the tables 4.1 and 4.2, we can conclude that the implemented algorithm is in general more efficient than the Shubert's method.

Finally, about the use of MINIMI for globally random minimizing n-dimensional functions, they seem to us very promising our first numerical results. We mention that, for the Rosenbrock's function, we reached the minimum point, from the starting point (50, 50), after 544 function evaluations (257 line-searches). The computed minimum value of the function was $0(10^{-12})$.

References.

[1] H. BREMERMANN, *A method of unconstrained global optimization*, Math. Biosci. **9** (1970), 1-15.

[2] L. C. W. DIXON, J. GOMULKA and G. P. SZEGO, *Towards a global optimization technique. Towards global optimization*, North-Holland 1975, 29-54.

[3] YU. G. EVTUSHENKO, *Numerical methods for finding global extrema (case of non uniform mesh)*, Ž. Vyčisl. Mat. i Mat. Fiz. (11) **6** (1971), 1390-1403.

[4] J. HARDY, *An implemented extension of Branin's method. Towards global optimization*, North-Holland 1975, 117-142.

[5] M. GAVIANO, *Some general results on the convergence of random search algorithms in minimization problems, Towards global optimization.* North-Holland 1975, 149-157.

[6] G. PETERS and J. H. WILKINSON, *Practical problems arising in the solution of polynomial equations*, J. Inst. Math. Appl. **8** (1971), 16-35.

[7] B. O. SHUBERT, *A sequential method seeking global maximum of a function*, SIAM J. Numer. Anal. (3) **9** (1972), 379-388.

[8] C. SUTTI: $[\bullet]_1$ *A technique for the global minimization of a polynomial*, Calcolo **13** (1976), 397-402; $[\bullet]_2$ *Metodi di minimizzazione monodimensionale*, Minimization Algorithms - Mathematical theories and computer results, Academic Press 1972, 147-173.

[9] C. SUTTI, L. TRABATTONI and P. BRUGHIERA, *A method for minimization of a one-dimensional nonunimodal function*, Minimization Algorithms - Mathematical theories and computer results, Academic Press 1972, 181-192.

[10] J. H. WILKINSON, *The algebraic eigenvalue problem*, Clarendon Press 1965, 464-465.

Abstract.

This paper provides subroutines for the global minimization of polynomials. To find the minima, the Sutti's algorithm $[\mathbf{8}]_1$ *is used. Some numerical results are presented.*

* * *