プログラムのページ

担当　森口繁一

6216. 3次方程式の解法

戸田英雄（電試応用数学研）

1. 問題

3次方程式 $X^3+AX^2+BX+C=0$ の実係数 A, B, C を N 組テープにパンチして与える．（その先頭に N をパンチしておく．）これを読みこんで根を計算し，A, B, C を印刷した後に，実根のときはそのまま，複素根のときは実部と虚部に分けて間に＋－の符号をいれ，虚部のあとに I をつけて印刷する．

2. 解法の手順

（1） 与えられた3次方程式の実根の第1近似を山内の式で求める．

（2） Newton の反復法で精度を高める．

（3） 残りの2根は，2次方程式の根の公式から求める．

3. 山内の式

（山内二郎，未発表であるが，この紹介は森口繁一編 ALGOL 入門 (1962) pp. 53~54 にある．）

$$X^3+aX^2+bX-c=0 \ (c>0) \qquad (1)$$

の実根は，$c^{1/3}=V$ とおき，

i） $aV+b>0$ のときは

$$\left(a+\frac{3}{2}V\right)X^2+\left(b-\frac{9}{16}V^2\right)X-\frac{31}{32}c=0$$

を解いて $0<X<V$ のものがよい近似を与える．

なお $a+\frac{3}{2}V=0$ のときは1次方程式をとくことになる．

ii） $aV+b<0$ のときは

$\frac{31}{32}X^2+\left(a+\frac{9}{16}V\right)X+\left(b-\frac{3}{2}V^2\right)=0$ を解いて $V<X$ のものがよい近似を与える．

iii） $aV+b=0$ のときは

$X=V$ とする．

この方法の原理は Lanczos (1956) pp. 6~8 によるものである．3次方程式の一つの実根 Y を0と1の間にただ一つあるように押えこみ，この区間で Chebyshev の多項式 $T_3^*(Y)=32\,Y^3-48\,Y^2+18\,Y-1$ の絶対値が1を超えない性質を利用して，2次方程式の根で実根 Y を近似する方法である．

4. プログラム

```
begin real A, B, C, A1, A2, C1, V, T, F, R,
   G1, G2, X1, X2, X3, P, Q, D, W, EPS;
   integer I, N;
READINTEGER(N); EPS:=₁₀-8;
for I:=1 step 1 until N do
begin READREAL(A); READREAL(B);
      READREAL(C); CRLF;
   PRINTREAL(A); PRINTREAL(B);
      PRINTREAL(C); CRLF;
   if C=0 then begin X1:=0; P:=A/2.0;
                     Q:=B; go to X1TYPE
                     end;
   if C>0 then begin A1:=-A; C1:=-C
                     end
      else begin A1:=A; C1:=C end;
YAMAUTI: V:=EXP(LN(-C1)/3.0);
   T:=A1×V+B;
   if T>0 then
   begin F:=B-0.5625×V×V;
     R:=F×F+3.875×C1×(A1+1.5×V);
     if A1+1.5×V=0 then
     X1:=0.96875×C1/F else
     X1:=(-F+SQRT(R))/(A1+1.5×V)/2.0
   end
   else if T<0 then
   begin F:=A1+0.5625×V;
       R:=F×F-(B-1.5×V×V)×3.875;
       X1:=(-F+SQRT(R))/1.9375
   end
   else X1:=V;
   if C>0 then X1:=-X1;
NEWTON: G1:=((A+X1)×X1+B)×X1+C;
   G2:=(2.0×A+3.0×X1)×X1+B;
   X2:=X1-G1/G2;
   X3:=ABS((X2-X1)/X1);
   CRLF; PRINTREAL(X1);
     PRINTREAL(G1); PRINTREAL(G2);
```

```
    CRLF; PRITREAL(X 2); PRINTREAL(X 3);
    X 1: = X 2;
    if X 3>EPS then go to NEWTON;
    P: = A+X 1; Q: = B+P×X 1; P: = P/2.0;
X1TYPE: CRLF; PRINTREAL(X 1);
QUADRATIC: D: = P×P−Q;
    if D≧0 then
    begin W: = SQRT(D); X 2: = −P+W; X3: =
        −P−W;
      CRLF; PRINTREAL(X 2); CRLF;
         PRINTREAL(X 3)
    end
    else
    begin X 2: = −P; X 3: = SQRT(−D);
      CRLF; PRINTREAL(X 2);
      PRINTSTRING('+−');
      PRINTREAL(X 3); PRINTSTRING('I')
    end
  end
end
```

〔注 1〕 ALGOL 60 では，このプログラム中の

V: = EXP(LN(−C 1)/3.0);

の代わりに

V: = (−C 1)↑(1/3);

と書いてよいことになっている．そうかくと，V = (−C 1)$^{1/3}$ の感じがよく出る．しかし，できあがるプログラムは上のものとほとんど同じになる．

〔注 2〕 0.5625 = 9/16, 3.875 = 4×31/32, 1.9375 = 2×31/32, 0.96875 = 31/32.

5. 数 値 例

方程式 $126x^3-26x+3=0$ の3根を計算した際の印刷記録は次のようになった．最後の3行が答である．

```
+.00000 00000 −19.    −.20634 92060 +00.    +.23809 52400 −01.
−.50650 39499 +00.    −.16154 77720 −02.    +.56328 95479 +00.
−.50363 60151 +00.    +.56622 16096 −02.
−.50363 60151 +00.    −.12474 49109 −04.    +.55459 85012 +00.
−.50361 35223 +00.    +.44660 89891 −04.
−.50361 35223 +00.    −.75851 85813 −09.    +.55453 05335 +00.
−.50361 35209 +00.    +.26583 40457 −08.
−.50361 35209 +00.
+.37880 78319 +00.
+.12480 56890 +00.
```

6217. 実対称行列の固有値問題

清水留三郎（東大計算センター）

実対称行列Aの左三角部分の要素の値を行ごとに読み取り，行列Aの固有値および固有ベクトルを計算し，それらを印刷する．固有値問題の解法には threshold Jacobi method を用いている．行列Aの要素は a_{11}, a_{21}, a_{22}, ……, a_{n1}, ……, a_{nn} の順に $A[0]$, $A[1]$, ……, $A\left[\frac{1}{2}n(n+1)-1\right]$ へ読み込まれる．プログラム中の procedure JACOBI の入力パラメターは行列の次数 N, 前記の構成の行列 A, 固有ベクトルの要不要を示すスイッチ VECTOR, 閾値の初期値 MINMAX, 閾値の最終値 EPS である．このうち MINMAX には行列の要素の各行の最大値の中の最小値をとるのが実用的であろう．また EPS はもちろん解の精度を左右するものである．出力パラメター LAMDA は固有値を表わす．固有ベクトルは $A\left[\frac{1}{2}n(n+1)\right]$ 以降に，各ベクトルの第1成分 n 個，第2成分 n 個，……，第 n 成分 n 個の順でそれぞれ得られる．

なお，このプログラムは本来固定小数点演算で納まるように考えられているので，浮動小数点演算では不要な statement が一部含まれている．

```
THRESHOLD JACOBI METHOD:
begin integer N, VECTOR, I;
  array LAMDA[1 : 5], A[0 : 39];
  real MINMAX, EPS;
procedure JACOBI(N, A, VECTOR, MINMAX,
      EPS, LAMDA);
  value N, VECTOR, MINMAX, EPS;
  array A, LAMDA; real MINMAX, EPS;
  integer N, VECTOR;
begin integer M, N 1, N 2, I, ROW, ROUTE,
      UPDOWN, IJ, II, JJ, J, ROUTEI,
      ROUTEJ, K;
  real THRESH, AII, AIJ, AJJ, DIJ, COSPHI,
      SINPHI, HFCS2P, HFSN2P, SQ,
      RADIUS;
  M: = (N×(N+1)) div 2;
```

```
  if VECTOR=1 then
  begin N1:=N+1; N2:=N↑2-1;
      for I:=N2 step -1 until 0 do
        A[M+I]:=0;
      for I:=N2 step-N1 until 0 do
        A[M+I]:=0.99999 99999;
      ROW:=N+N; ROUTE:=M+N↑2
  end else
  begin ROW:=N; ROUTE:=M end;
  THRESH:=MINMAX;
CYCLE: UPDOWN:=0;
  IJ:=M;
  for I:=N-1 step -1 until 1 do
  begin II:=IJ:=JJ:=IJ-1;
      for J:=I-1 step -1 until 0 do
      begin IJ:=IJ-1; JJ:=JJ-(J+2);
            AIJ:=A[IJ];
  if ABS(AIJ)<THRESH then go to CONT;
  UPDOWN:=-1;
  AII:=A[II]×0.5; AJJ:=A[JJ]×0.5;
  DIJ:=AJJ-AII;
  if ABS(AIJ)≧ABS(DIJ) then
  begin HFCS2P:=(DIJ×0.5)/AIJ;
      if HFCS2P<0 then
      begin HFCS2P:=-HFCS2P; HFSN2P
        :=-0.5 end else HFSN2P:=0.5;
      SQ:=HFCS2P↑2;
      if SQ=0 then go to PHI1 else go to PHI2
  end else
  begin HFCS2P:=0.5;
        HFSN2P:=(AIJ×0.5)/DIJ;
      SQ:=HFSN2P↑2;
      if SQ=0 then
      begin COSPHI:=0.99999 99999; go to SN
      end
  end;
PHI2: RADIUS:=SQRT(0.25+SQ);
  HFCS2P:=(HFCS2P×0.5)/RADIUS;
  HFSN2P:=(HFSN2P×0.5)/RADIUS;
PHI1: COSPHI:=SQRT(0.5+HFCS2P);
SN: SINPHI:=HFSN2P/COSPHI;
  ROUTEI:=ROUTE+I;
  ROUTEJ:=ROUTE+J;
  for K:=ROW step -1 until 1 do
  begin if N<K then
      begin ROUTEI:=ROUTEI-N; ROUTEJ
        :=ROUTEJ-N; go to ROT
      end;
      if I<K then
      begin ROUTEI:=ROUTEI-K; go to JK
      end;
      ROUTE I:=ROUTE I-1;
      if J<Kthen JK: ROUTEJ:=ROUTEJ-K
              else ROUTEJ:=ROUTEJ-1;
  ROT: SQ:=A[ROUTEI];
      A[ROUTEI]:=SQ×COSPHI-A[ROUTEJ]
        ×SINPHI;
      A[ROUTEJ]:=A[ROUTEJ]×COSPHI+
        SQ×SINPHI
    end K;
      AIJ:=DIJ×HFCS2P+AIJ×HFSN2P;
      AIJ:=AIJ+AIJ;
      A[II]:=AII-AIJ+AJJ;
      A[IJ]:=0;
      A[JJ]:=AII+AIJ+AJJ;
CONT:
    end J
  end I;
      if EPS<THRESH then
      begin THRESH:=THRESH×0.1; go to
        CYCLE end;
      if UPDOWN<0 then go to CYCLE;
      II:=M-1;
      for I:=N step -1 until 1 do
      begin LAMDA[I]:=A[II]; II:=II-I end
end JACOBI;
START: READINTEGER(N);
  for I:=0 step 1 until (N×(N+1)) div 2-1
      do READREAL(A[I]);
  READREAL(MINMAX);
  VECTOR:=1;
  EPS:=0.0000001;
  JACOBI(N, A, VECTOR, MINMAX, EPS,
      LAMDA);
  for I:=1 step 1 until N do PRINTREAL
      (LAMDA[I]);
  for I:=(N×(N+1)) div 2 step 1
        until (N×(N+1) div 2+N↑2-1 do
        PRINTREAL(A[I])
end
```