プログラムのページ

78-05 連分数近似の最良化プログラム

浜 田 穂 積*

1. はじめに

関数の近似式の形として有理式を用いる場合の，最良近似式の定数を求めるには，Padè 展開の係数の補正を行うより，連分数の係数の補正を行う方が，計算時間，得られる定数の精度のいずれの点でも有利であることを既に示した[1]．ここでは一例として，正接関数 (tan) の定数を計算するプログラムを示す．

2. 計算手順

最良化しようとする関数は次の形の連分数で表わせるものとする．

$$F(x)=\frac{x}{a_1+}\,\frac{b_1x^2}{a_2+}\,\frac{b_2x^2}{a_3+}\cdots\frac{b_{i-1}x^2}{a_i+}\cdots \tag{1}$$

そして，次の形の近似式を求める．

$$G(x)=\frac{x}{c_1+}\,\frac{b_1x^2}{c_2+}\,\frac{b_2x^2}{c_3+}\cdots\frac{b_{n-1}x^2}{c_n} \tag{2}$$

これは $c_i=a_i+d_i$ とするとき，a_i に対する補正量 d_i を定めて最良化する．$\tan x$ の場合，次の通りである．

$$a_i=2_i-1,\ \ b_i=-1 \tag{3}$$

さて，ここで $f(x)$，$g(x)$ を次により定義する．

$$\left.\begin{aligned}f(x)&=a_1+\frac{b_1x^2}{a_2+}\,\frac{b_2x^2}{a_3+}\cdots\frac{b_{m-1}x^2}{a_m+}\cdots\\ g(x)&=c_1+\frac{b_1x^2}{c_2+}\,\frac{b_2x^2}{c_3+}\cdots\frac{b_{n-1}x^2}{c_n}\end{aligned}\right\} \tag{4}$$

$f(x)$，$g(x)$ は共に偶関数である．そこで，ここでは相対誤差に関する最良化を例としてとりあげる．相対誤差関数を $E(x)$ とすると，

$$E(x)=\frac{G(x)-F(x)}{F(x)}\doteqdot\frac{x/g(x)-x/f(x)}{x/f(x)}$$

$$=\frac{f(x)-g(x)}{g(x)} \tag{5}$$

であり，これも偶関数である．f_i，g_i を以下で定義する．

$$f_i=a_i+\frac{b_ix^2}{f_{i+1}} \tag{6}$$

$$g_i=a_i+d_i+\frac{b_ix^2}{g_{i+1}},\ \ g_n=a_n+d_n \tag{7}$$

このとき

$$f(x)=f_1,\ \ g(x)=g_1 \tag{8}$$

である．これらを (5) に代入して次を得る．

$$E(x)=-\frac{1}{g_1}\left(d_1-\frac{b_1x^2}{f_2g_2}\left(d_2-\cdots-\frac{b_{n-1}x^2}{f_ng_n}\left(d_n-\frac{b_nx^2}{f_{n+1}}\right)\cdots\right)\right) \tag{9}$$

これは次の漸化式で計算する．

$$\left.\begin{aligned}&u_{n+1}=-1\\ &u_i=-\frac{1}{g_i}\left(d_i+\frac{b_ix^2u_{i+1}}{f_{i+1}}\right)\ \ (i=n,n-1,\cdots,1)\\ &E(x)=u_1\end{aligned}\right\} \tag{10}$$

また，d_i を得るため，d_i に対する補正量 dd_i を計算してそれで d_i を修正する反復法を用いるという通常行われる方法を用いる．このため，偏差点の x 座標に関して，次が成立するための dd_i を求める．

$$\left.\begin{aligned}&E(x_{j-1},\cdots,d_i+dd_i)+E(x_j,\cdots,d_i+dd_i,\cdots)=0\\ &(j=1,2,\cdots,n)\end{aligned}\right\} \tag{11}$$

このとき d_i+dd_i を新しい d_i と置いて (11) を反復し，d_i が収束すれば，最良近似式が得られる．このため (11) を，dd_i を微小量と考えて線型化し，dd_i を解く．このとき必要な $\partial E/\partial d_i$ は次で与えられる．

$$\frac{\partial E}{\partial d_i}=(-1)^if_1\frac{b_1b_2\cdots b_{i-1}}{(g_1g_2\cdots g_i)^2}x^{2i-2} \tag{12}$$

以上の関係を用いて次の手順で計算する．

(A) d_i の初期値を求める．それには (9) の $E(x)$ が

$$x_j=\rho\cos\left(\frac{2j-1}{2n}\cdot\frac{\pi}{2}\right)\ \ (j=1,2,\cdots,n)$$

において0になるとして，n 元連立方程式を解く．ただし，g_i の計算においては d_i がまだ計算されていないので，$d_i=0$ すなわち $g_i=f_i$ とする．また ρ は収束半径，すなわち近似区間を $-\rho\leqq x\leqq\rho$ とする．

(B) $E(x)$ が極値をとる x の近似値 y_i として，

* (株)日立製作所システム開発研究所

次の値を初期値に設定する．

$$y_j=\rho\cos\left(\frac{2j}{2n}\cdot\frac{\pi}{2}\right)\quad(j=0,1,2,\cdots,n)$$

（C） d_i の補正量 dd_i を計算して，d_i+dd_i で d_i を置き換える．この手順を数回繰返す．

（D） 極値を与える x（y_i で表わす）を，その近似値の近くにある正しい値に修正する．ここでは近似値と，それをはさむ2点との3点を通る放物線の頂点の x 座標を新たな x 座標とする，という過程を反復する方法を用いている．

（E） 極値の絶対値のバラツキを調べ，収束したと見なせるようになるまで（C），（D）を繰返す．

3. プログラム

上に述べた手順を行うプログラムの概略を**図-1** に，詳細を**図-2** に示す．これは HITAC 8000 シリーズの EDOS/EDOS-MSO ALGOL で書かれたものである．この ALGOL は JIS 7070 相当であるが，このプログラムは JIS 4060 の機能で書かれている．

このプログラムは（1）の形の連分数の係数に対する補正量 d_i を計算して出力する．主プログラムで a_i, b_i と ρ を与える．ここでは

$$a_i=2i-1,\ \ b_i=-1,\ \ \rho=\pi/4$$

と与えて $\tan x$ の相対誤差に関する最良化を，$n=2$ ～8 について行う．

手続き INITDY は手順（A）と（B）を行う．

手順き REFINE は手順（C）と（D）を行う．手順（C）の連立方程式の係数と右辺の符号をすべて逆にしてある．また関数 ER は（9）の $E(x)$ の値を計算するものであるが，$m=20$ とした．これらは単に便宜上の理由からだけである．（C）の繰返しは固定的に8回としたが，これで十分であった．次に手順(D)については Muller 法の変形を用いた．すなわち，3点を通る2次式を想定し，その頂点の x 座標を求め，新しい3点を得る，このとき得られた点の両側に1点ずつ，前の3点の中から選ぶ．もしこの条件が満たされなくなったら収束したと見なし，その前に得られた

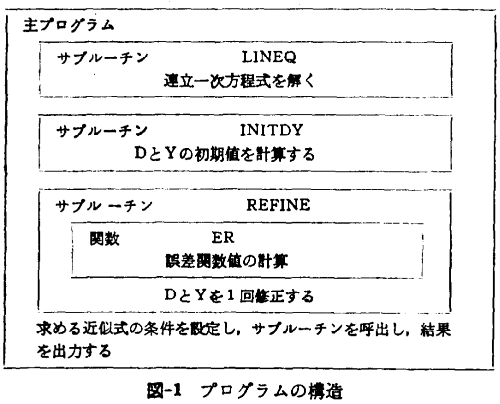

図-1 プログラムの構造

```
BEGIN
  INTEGER I,N;
  REAL RADIUS,MAXERROR,CONVERGENCY;
  REAL ARRAY A,B(1:20),C,D(1:13),E,Y(0:13);
  PROCEDURE LINEQ(N,Z);
    VALUE N;
    INTEGER N;
    REAL ARRAY Z;
  BEGIN
    INTEGER I,J,K;
    FOR J:=1 STEP 1 UNTIL N DO
      FOR I:=J+1 STEP 1 UNTIL N+1 DO
      BEGIN
        Z(J,I):=Z(J,I)/Z(J,J);
        FOR K:=J+1 STEP 1 UNTIL N DO
          Z(K,I):=Z(K,I)-Z(K,J)*Z(J,I)
      END ELIMINATION;
    FOR J:=N-1 STEP -1 UNTIL 1 DO
      FOR I:=J+1 STEP 1 UNTIL N DO
        Z(J,N+1):=Z(J,N+1)-Z(J,I)*Z(I,N+1)
  END LINEQ;
  PROCEDURE INITDY(R,N,A,B,D,Y);
    VALUE R,N;
    INTEGER N;
    REAL R;
    REAL ARRAY A,B,D,Y;
  BEGIN
    INTEGER I,J;
    REAL HPI,S,X;
    REAL ARRAY F(2:N+1),Z(1:N,1:N+1);
    HPI:=1.5707963267948966;
    FOR J:=1 STEP 1 UNTIL N DO
    BEGIN
      X:=R*COS((2*J-1)/(2*N)*HPI);
      S:=A(20);
      FOR I:=19 STEP -1 UNTIL N+2 DO
        S:=B(I)*X*X/S+A(I);
      FOR I:=N+1 STEP -1 UNTIL 2 DO
        F(I):=S:=B(I)*X*X/S+A(I);
      Z(J,1):=1;
      FOR I:=1 STEP 1 UNTIL N DO
        Z(J,I+1):=-Z(J,I)*B(I)*X*X/F(I+1)**2;
      Z(J,N+1):=-Z(J,N+1)*F(N+1)
    END;
    LINEQ(N,Z);
    FOR J:=1 STEP 1 UNTIL N DO
      D(J):=Z(J,N+1);
    FOR J:=0 STEP 1 UNTIL N-1 DO
      Y(J):=R*COS((2*J)/(2*N)*HPI);
    Y(N):=0
  END INITDY;
  PROCEDURE REFINE(N,A,B,C,D,E,Y,ME,CONV);
    VALUE N;
    INTEGER N;
    REAL ME,CONV;
    REAL ARRAY A,B,C,D,E,Y;
  BEGIN
    INTEGER I,J,K;
    REAL S,T,U,V,W,X,E1,E2,E3;
    REAL ARRAY G(1:N),Z(0:N,1:N+1);
    REAL PROCEDURE ER(X);
      VALUE X;
      REAL X;
    BEGIN
      INTEGER I;
      REAL S,T,U;
      FOR I:=20 STEP -1 UNTIL N+1 DO
        S:=(IF I=20 THEN 0
                    ELSE B(I)*X*X/S)+A(I);
      U:=-1;
      FOR I:=N STEP -1 UNTIL 1 DO
      BEGIN
        T:=(IF I=N THEN 0
                  ELSE B(I)*X*X/T)+A(I)+D(I);
        U:=-(U*B(I)*X*X/S+D(I))/T;
        S:=B(I)*X*X/S+A(I)
      END;
      ER:=U
    END ER;
    COMMENT CORRECT D;
```

（つづく）

```
      FOR K:=1 STEP 1 UNTIL 8 DO
      BEGIN
        FOR J:=0 STEP 1 UNTIL N DO
        BEGIN
          X:=Y(J);
          G(N):=T:=A(N)+D(N);
          FOR I:=N-1 STEP -1 UNTIL 1 DO
            G(I):=T:=B(I)*X*X/T+A(I)+D(I);
          S:=A(20);
          FOR I:=19 STEP -1 UNTIL 1 DO
            S:=B(I)*X*X/S+A(I);
          Z(J,1):=S/T/T;
          FOR I:=2 STEP 1 UNTIL N DO
            Z(J,I):=-Z(J,I-1)*B(I-1)*X*X/G(I)**2;
          Z(J,N+1):=ER(X)
        END;
        FOR J:=N STEP -1 UNTIL 1 DO
          FOR I:=1 STEP 1 UNTIL N+1 DO
            Z(J,I):=Z(J,I)+Z(J-1,I);
        LINEQ(N,Z);
        FOR I:=1 STEP 1 UNTIL N DO
          D(I):=D(I)+Z(I,N+1)
      END CORRECT D;
      COMMENT CORRECT Y;
      E(0):=ER(Y(0));
      FOR J:=1 STEP 1 UNTIL N-1 DO
      BEGIN
        U:=(Y(J-1)+Y(J))/2; E1:=ER(U);
        V:=(Y(J)+Y(J+1))/2; E2:=ER(V);
        W:=Y(J);            E3:=ER(W);
        FOR X:=(U*U*(E2-E3)+V*V*(E3-E1)+W*W*(E1-E2))
          /(U*(E2-E3)+V*(E3-E1)+W*(E1-E2))/2
          WHILE (X-U)*(X-V)<0 & X¬=W DO
        BEGIN
          S:=ER(X);
          IF ABS(S)>ABS(E3) THEN
          BEGIN
            IF X>W THEN BEGIN V:=W; E2:=E3 END
                   ELSE BEGIN U:=W; E1:=E3 END;
            W:=X;
            E3:=S
          END ELSE
            IF X>W THEN BEGIN U:=X; E1:=S END
                   ELSE BEGIN V:=X; E2:=S END
        END;
        Y(J):=W;
        E(J):=E3
      END;
      E(N):=ER(Y(N));
      S:=SIGN(E(0));
      ME:=T:=E(0)*S;
      FOR J:=1 STEP 1 UNTIL N DO
      BEGIN
        S:=-S;
        ME:=MAX2(ME,E(J)*S);
        T:=MIN2(T,E(J)*S)
      END;
      CONV:=T/ME;
      FOR I:=1 STEP 1 UNTIL N DO
        C(I):=LOG(ABS(Z(0,I))/ME)
    END REFINE;
    RADIUS:=.78539816339744831;
    FOR I:=1 STEP 1 UNTIL 20 DO
    BEGIN
      A(I):=2*I-1;
      B(I):=-1
    END;
    OUTPUT1(2,'*11B"TAN(X) R="D.7D',RADIUS);
    OUTPUT0(2,'//" N  I"11B"D(I)"13B"MRE"');
    FOR N:=2 STEP 1 UNTIL 13 DO
    BEGIN
      INITDY(RADIUS,N,A,B,D,Y);
      I:=0; CONVERGENCY:=0;
      FOR I:=I+1 WHILE CONVERGENCY<.9999 & I<=15 DO
        REFINE(N,A,B,C,D,E,Y,MAXERROR,CONVERGENCY);
      OUTPUT3(2,'/40B-ZD.DD,-5ZD,4B-D.6D',
        -LOG(1-CONVERGENCY),I-1,E(0)/MAXERROR);
      FOR I:=1 STEP 1 UNTIL N DO
      BEGIN
        IF I=1 THEN OUTPUT1(2,'/ZD',N)
               ELSE OUTPUT0(2,'/8B');
        OUTPUT2(2,'-ZDB,-D.16DE+DD',I,D(I));
        IF I=1 THEN OUTPUT1(2,'-D.3DE+DDB',MAXERROR)
               ELSE OUTPUT0(2,'11B');
        OUTPUT3(2,'-ZD.DD,2(-ZD.6D)',
          C(I),Y(I),E(I)/MAXERROR)
      END
    END
  END MINIMAXIMIZATION OF CONTINUED FRACTION
```

図-2　連分数の最良化プログラム

新しい点の x 座標の値を極値をとる x の値とする.

手順（E）は主プログラムで行う. その前にあらかじめ手続き REFINE の中で次の値を計算しておく.

$$s=\mathrm{sgn}(E(y_0))$$

$$e_{\max}=\max_j\{(-1)^j sE(y_i)\}$$

$$e_{\min}=\min_j\{(-1)^j sE(y_i)\}$$

$$k=e_{\min}/e_{\max}$$

そして $k>0.9999$ となったら収束したと見なす. ただし繰返しは多くても 15 回までとする. 図-3 に出力の一部を示す. このうち（f）の最後の値は, $e_{\max}$ だけ関数値を変化させる d_i の値を示すものであり, d_i の桁数を定める目安となる[1].

TAN(X) R=0.7853982

N	I	D(I)	MRE	(a)	(b)	(d)
				15.36	2	-1.000000
2	1	1.3037929752988941E-03	1.302E-03	2.99 (c)	0.575938	1.000000
	2	-1.2934853229009223E-01		1.66	0.000000	-1.000000
				5.45	2	-0.999996
3	1	-5.7731383860934924E-06	5.773E-06	5.34	0.690019	0.999998
	2	1.3694983146858293E-03		4.22	0.409994	-1.000000
	3	-1.3667814820018976E-01		2.63	0.000000	0.999996
				4.95	2	-0.999989
4	1	1.4163813770077187E-08	1.416E-08	7.95	0.730582	0.999997
	2	-6.0756854900322005E-06		6.83	0.568360	-0.999989
	3	1.1660949257125745E-03		5.23	0.312588	1.000000
	4	-1.4065716062610192E-01		3.35	0.000000	-0.999989
				4.33	2	-0.999953
13	1	-4.1194531795307344E-38	4.120E-38	37.49	0.779863	0.999964
	2	1.9695814420733019E-34		36.36	0.763312	-0.999989
	3	-4.3490963080864388E-31		34.77	0.735909	0.999998
	4	7.4204756818177160E-28		32.88	0.697941	-0.999983
	5	-1.0904144276738556E-24		30.77	0.649826	0.999962
	6	1.4276733277037168E-21		28.48	0.592132	-0.999953
	7	-1.6788108774384405E-18		26.04	0.525581	0.999964
	8	1.7637126661480532E-15		23.48	0.451058	-0.999986
	9	-1.6308953813025866E-12		20.61	0.369605	1.000000
	10	1.2945807170060710E-09		18.05	0.282410	-0.999997
	11	-8.4831619766865312E-07		15.19	0.190768	0.999980
	12	4.3063751824868296E-04		12.26	0.096156	-0.999961
	13	-1.4975671554311330E-01		9.26	0.000000	0.999953

MRE: Maximum Relative Error

（a）　$-\log_{10}(1-k)$

（b）　収束計算の繰返し回数

（c）　$\log_{10}\left(\max_j\left|\frac{\partial E(x_j)}{\partial d_i}\right|\Big/e_{\max}\right)$

ただし $e_{\max}$ は MRE の値

（d）　極値をとる x の値と, そのときの誤差 $E(x_i)/e_{\max}$

図-3　出　力　例

4. お わ り に

正接関数（tan）の $|x|\leqq\pi/4$ にお

ける最良近似連分数を，$n=2\sim13$ について計算したプログラムを示した．いずれの場合も，収束計算の外側のループ2回で，バラツキ（k）が 0.9999 に収まった．計算条件は特によすぎるというわけではないので，この結果は計算法のよさを示していると考えられる．

プログラムは ALGOL で記述したので，不自然な制約も少なく，読みやすいものになった．また，いわゆる構造的プログラミングの一つの条件である **go to** 文なしのプログラムができたのも，ALGOL を用いたメリットであろう．

参 考 文 献

1) 浜田穂積：有理式近似および連分数近似の最良化について，情報処理，Vol. 19，No. 11，pp. 1065～1071 (1978).

(昭和52年4月27日受付)
(昭和53年1月17日再受付)