プログラムのページ

担当　一　松　　信

6607. 不等間隔な点についてのΣにかんして直交する多項式

伊藤猛宏（九州大学工学部）

1. 目　　的

独立変数xの有限個の値 $x=x_\mu$, $\mu=1, 2, \cdots, M$における関数値 $f_\mu\equiv f(x_\mu)$ が既知であるとき，$f(x)$を K 次の多項式 $y_K(x)$ によってつぎのように近似することを考える.

$$y_K(x)=\sum_{\nu=0}^{K}S_\nu^{(K)}P_\nu(x),\ P_\nu(x)\text{: 確かに }\nu\text{ 次の多項式} \qquad (1)$$

このとき残差 e_μ, $e_\mu\equiv f_\mu-y_K(x_\mu)$, $\mu=1, 2, \cdots, M$の平方和 $\|e\|^{(K)}\equiv\sum_{\mu=1}^{M}e_\mu^2$ が最小になるように，すなわち最小2乗近似的に $\{S_\nu^{(K)}\}$, $\nu=0, 1, 2, \cdots, K$を定めることは，正規方程式

$$\left.\begin{aligned}&\sum_{\nu=0}^{K}W_{i\nu}S_\nu^{(K)}=\omega_i,\ i=0, 1, 2, \cdots, K\\&W_{i\nu}\equiv\sum_{\mu=1}^{M}P_i(x_\mu)P_\nu(x_\mu),\ \omega_i\equiv\sum_{\mu=1}^{M}f_\mu P_i(x_\mu)\end{aligned}\right\}\quad(2)$$

を解くことと同値である.

さて，(2) 式の係数 $W_{i\nu}$ に関してさらに

$$W_{i\nu}=\delta_{i\nu}A_\nu,\ A_\nu\equiv W_{\nu\nu},i,\ \nu=0,1,2,\cdots,K \quad (3)$$

が成り立てば，すなわち $\{P_\nu(x)\}$ が勝手な間隔の$\{x_\mu\}$ についての Σ に関して直交すれば，正規方程式(2) の左辺の係数行列における要素は主対角線上のそれを残してすべて消える. したがって

$$S_\nu\equiv S_\nu^{(K)}=\omega_\nu/W_{\nu\nu},\ \nu=0, 1, 2, \cdots, K \qquad (4)$$

となるから，係数 $\{S_\nu^{(K)}\}$ は展開式 (1) の次数Kに関係しない $\{S_\nu\}$ となり，一般の正規方程式 (2) を解く煩雑さも相当に緩和される.

$\{(x_\mu, f_\mu)\}$ を与えて，かような $\{P_\nu(x)\}$ と $\{S_\nu\}$を決定することを目的とする.

2. 多項式の構成

G.E. Forsythe (J. Soc. Indust. Appl. Math., **5** -74, 1957) による.

$$\left.\begin{aligned}&P_0(x)=1;\\&P_1(x)=(x-\alpha_1)P_0(x);\\&P_2(x)=(x-\alpha_2)P_1(x)-\beta_1P_0(x);\\&\cdots\cdots\cdots\cdots\\&P_{\nu+1}(x)=(x-\alpha_{\nu+1})P_\nu(x)-\beta_\nu P_{\nu-1}(x).\\&\alpha_\nu=\frac{\sum_{\mu=1}^{M}x_\mu\{P_{\nu-1}(x_\mu)\}^2}{W_{\nu-1\nu-1}},\quad\beta_\nu=\frac{W_{\nu\nu}}{W_{\nu-1\nu-1}}\end{aligned}\right\}\quad(5)$$

$$\|e\|^{(K)}=\sum_{\mu=1}^{M}f_\mu^2-\sum_{\nu=0}^{K}W_{\nu\nu}S_\nu^2=\|e\|^{(K-1)}-W_{KK}S_K^2 \qquad (6)$$

$$\sigma_K^2\equiv\frac{\|e\|^{(K)}}{M-1-K}=\frac{M-1}{M-1-K}\|e\|^{(K-1)}-\frac{W_{KK}S_K^2}{M-1-K} \qquad (7)$$

(6) 式の残差の平方和は次数 K が大きくなるとき非減少で，$K=M-1$ で計算の誤差を別にすれば0になるから，近似の適切さの判断には使えない. したがって，この効果を打ち消した最も簡単な判定関数 σ_K^2を (7) 式で定義し，その大きさと変動を見て多項式の最適次数Kを選定する.

3. プログラム

上記の算法に基づいて，ALGOLプログラムを procedure の形で二つ作成した. 使用上の制限・注意などを主にして説明する.

1) 多項式 $\{P_\nu(x)\}$ と係数 $\{S_\nu\}$ をきめるプログラム　$\{P_\nu(x)\}$ をきめるということは $\{\alpha_\nu, \beta_\nu\}$ を求めるということである. つぎのような形式で宣言される.

```
procedure GENEPO (V 1, V 2, M, KMAX, KP,
EPS, A, B, C, D, SA 1, SA 2, SA 3, SA 4, K);
value M, KMAX, KP, EPS; integer M, KMAX,
K, KP; real EPS, A, B, C, D; array V 1, V 2,
SA 1, SA 2, SA 3, SA 4;
    begin·································end;
```

パラメータの V1 から EPS までが入力としてプログラム本体から受け渡されるもので，AからKまでがこの procedure で定められるものである. 各パラメータの意味・制限などを列挙するとつぎのようになる.

V 1, V 2: データ，V 1[1]$=x_1$, V 2[1]$=f_1$, … V 1[M]$=x_\mu$, V 2[M]$=f_\mu$ ただし，$\max_\mu|x_\mu|\leqq10^{50}$, $\max_\mu|f_\mu|\leqq10^{50}$ であり，$x_1, x_2, \cdots$ の大小の順序は一定する必要はない.

M：データの組織．$M \leqq 300$.

KMAX：作り出す多項式の次数の上限．この上限はM−1（このときはLagrange補間になってしまう）でもおさえられる．つまり次数 K≦min{KMAX, M−1}．次数をp次より高めたくないときはKMAX=P としておけばよい．

KP：係数表$\{(S_\nu, \alpha_\nu, \beta_\nu)\}$をパンチし始める次数．初めてやるときは0としておく．

EPS：σ_K^2の許容値．σ_K^2<EPSかつ$|\sigma_K^2-\sigma_{K-1}^2|$<EPS^{2}になれば，$\sigma_K^2$が収束したと判断して，そのときの$K$を多項式の次数とする．しかし，この判定法は適切であるかどうか，わからない．その意味で係数表$\{(S_\nu, \alpha_\nu, \beta_\nu)\}$は上の条件が満足されなくとも計算したところまではパンチされるようにしてある．その場合 min{KMAX, M−1} までパンチされることになる．

⦿A, B, C, D：$\tilde{x}_\mu=Ax_\mu+B$, $f_\mu=C\tilde{f}_\mu+D$ によって，$\{(x_\mu, f_\mu)\}$は$\{(\tilde{x}_\mu, \tilde{f}_\mu)\}$の範囲$[0, 1]\times[0, 1]$に移して扱われる．係数表$\{(S_\nu, \alpha_\nu, \beta_\nu)\}$はこの$\{(\tilde{x}, \tilde{f})\}$にかんするものである．以下⦿印のものはパンチされる．

SA 1：σ_ν^2.

⦿SA 2：S_ν.

⦿SA 3：α_ν. α_0は定義していないが0としておく．

⦿SA 4：β_ν. β_0は定義していないが0としておく．

⦿K：σ_K^2が収束したときは，そのときのKの値，つまり(1)式の次数，そうでないときは0になる．次数=0が最良近似になることは許されない．

2）引数xの一つの値に対応する多項式$y_K(x)$の値を計算するプログラム　これはつぎのような形式で宣言される．

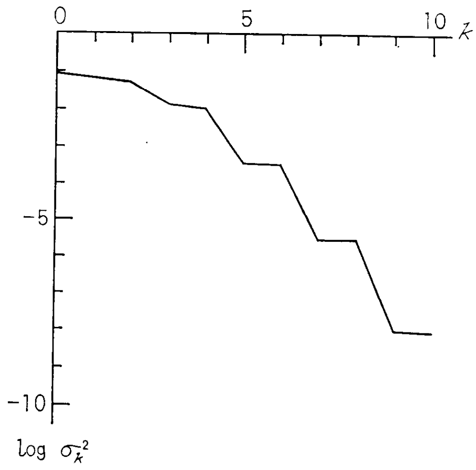
procedure CALPO (K, A, B, C, D, SA 2, SA 3, SA 4, X, Y); **value** K, A, B, C, D; **integer** K; **real** A, B, C, D, X, Y; **array** SA 2, SA 3, SA 4;
　　begin………………………………**end**;

パラメータのKからXまでが入力としてプログラム本体から受け渡されるもので，Yがこのprocedureで定められる．K, A, B, C, D, SA 2, SA 3 および SA 4 などは procedure GENEPO におけると同じ意味をもち，Xが引数xの値であり，Yが$y_K(x)$の値である．

procedure GENEPO によってパンチしたカードを直接この procedure で使うことができる．次数Kの値をパンチされた値より小さくしたいときは，そのカードだけ別にパンチすればよい．

第1表　$J_0(x)$, $0\leqq x\leqq 10$ の近似式作成のためのデータとそれによる結果の比較

X	F	FC
$.0000_{10}\ 0$	$.1000_{10}\ 1$	$.1000_{10}\ 1$
$.5000_{10}\ 0$	$.9385_{10}\ 0$	$.9386_{10}\ 0$
$.9000_{10}\ 0$	$.8075_{10}\ 0$	$.8073_{10}\ 0$
$.1200_{10}\ 1$	$.6711_{10}\ 0$	$.6710_{10}\ 0$
$.1400_{10}\ 1$	$.5669_{10}\ 0$	$.5669_{10}\ 0$
$.1500_{10}\ 1$	$.5118_{10}\ 0$	$.5119_{10}\ 0$
$.2000_{10}\ 1$	$.2239_{10}\ 0$	$.2241_{10}\ 0$
$.2400_{10}\ 1$	$.2510_{10}-\ 2$	$.2571_{10}-\ 2$
$.2700_{10}\ 1$	$-.1424_{10}\ 0$	$-.1425_{10}\ 0$
$.2900_{10}\ 1$	$-.2243_{10}\ 0$	$-.2244_{10}\ 0$
$.3000_{10}\ 1$	$-.2600_{10}\ 0$	$-.2602_{10}\ 0$
$.3500_{10}\ 1$	$-.3801_{10}\ 0$	$-.3802_{10}\ 0$
$.3900_{10}\ 1$	$-.4018_{10}\ 0$	$-.4018_{10}\ 0$
$.4200_{10}\ 1$	$-.3766_{10}\ 0$	$-.3765_{10}\ 0$
$.4400_{10}\ 1$	$-.3423_{10}\ 0$	$-.3422_{10}\ 0$
$.4500_{10}\ 1$	$-.3205_{10}\ 0$	$-.3205_{10}\ 0$
$.5000_{10}\ 1$	$-.1776_{10}\ 0$	$-.1776_{10}\ 0$
$.5400_{10}\ 1$	$-.4121_{10}-\ 1$	$-.4130_{10}-\ 1$
$.5700_{10}\ 1$	$.5992_{10}-\ 1$	$.5982_{10}-\ 1$
$.5900_{10}\ 1$	$.1220_{10}\ 0$	$.1220_{10}\ 0$
$.6000_{10}\ 1$	$.1506_{10}\ 0$	$.1506_{10}\ 0$
$.6500_{10}\ 1$	$.2601_{10}\ 0$	$.2602_{10}\ 0$
$.6900_{10}\ 1$	$.2981_{10}\ 0$	$.2983_{10}\ 0$
$.7400_{10}\ 1$	$.2786_{10}\ 0$	$.2787_{10}\ 0$
$.7800_{10}\ 1$	$.2154_{10}\ 0$	$.2153_{10}\ 0$
$.8000_{10}\ 1$	$.1716_{10}\ 0$	$.1715_{10}\ 0$
$.8800_{10}\ 1$	$-.3923_{10}-\ 1$	$-.3918_{10}-\ 1$
$.9400_{10}\ 1$	$-.1768_{10}\ 0$	$-.1767_{10}\ 0$
$.9800_{10}\ 1$	$-.2323_{10}\ 0$	$-.2325_{10}\ 0$
$.1000_{10}\ 2$	$-.2459_{10}\ 0$	$-.2459_{10}\ 0$



第1図　次数Kの増加にともなう判定関数σ_K^2の変動

4. 使用例

区間[0, 10]におけるベッセル関数$J_0(x)$の近似

第2表　$J_0(x)$, $0 \leqq x \leqq 10$ のための係数表 $\{(S_\nu, \alpha_\nu, \beta_\nu)\}$, $\nu=0, 1, \cdots, 10$

```
 10,/ORDER  OF GENERATED POLYNOMIAL
  .999999999766E- 1,    .000000000000E  0,    .140182999987E  1,  -.401829999872E  0,
 /A, B, C AND D
  .356020344887E  0,    .000000000000E  0,    .000000000000E  0, /ORDER  0,
 -.464066465618E  0,    .471999999951E  0,    .820959999109E- 1, /ORDER  1,
  .172074746527E  1,    .532629213994E  0,    .777546080935E- 1, /ORDER  2,
 -.852259517834E  1.    .515363879968E  0,    .667310635618E- 1, /ORDER  3,
 -.102817421146E  2,    .477769242250E  0,    .658418497478E- 1, /ORDER  4,
  .684589020014E  2,    .475807802286E  0,    .643372499325E- 1, /ORDER  5,
  .132343883105E  2,    .486792765907E  0,    .633585193136E- 1, /ORDER  6,
 -.189581996322E  3,    .531027465593E  0,    .627090565685E- 1, /ORDER  7,
 -.101895136610E  2,    .505845694337E  0,    .564626294654E- 1, /ORDER  8,
  .274397969246E  3,    .465459445142E  0,    .689074451511E- 1, /ORDER  9,
 -.266597414315E  2,    .553178057540E  0,    .511022612190E- 1, /ORDER 10,
 /COEFFICIENT(I), ALPHA(I) AND BETA(I)
```

式を EPS$=10^{-4}$ で作成した例を示す．第1表のXおよびF欄をそれぞれ独立変数 x_μ の 30 個の値およびそれに対応する関数値 $f(x_\mu)$ の 30 個の値として与えた．つまり 3. の procedure GENEPO における array V 1 および V 2 とした ($M=30$).

この場合，次数 K が 10 次まで高められたとき所定の収束判定条件が満足された．Kの増加に伴う伴定関数 σ_K^2 の変化の模様を第1図に示す．この例ではKとともに σ_K^2 は単調減少しているが，場合によってはそうでないこともある．第1表のFC欄は作成した多項式の値を procedure CALPO によって計算したものである．また第2表は係数表 $\{(S_\nu, \alpha_\nu, \beta_\nu)\}$ のパンチカードのリストである．なお第1図の σ_K^2 および第2表の $\{(S_\nu, \alpha_\nu, \beta_\nu)\}$ の値が独立変数も関数値も [0, 1] に normalize したときの値であることは注意を要する．

```
procedure GENEPO (V 1, V 2, M, KMAX, KP, EPS, A, B, C, D, SA 1, SA 2, SA 3, SA 4, K);
    value M, KMAX, KP, EPS; integer M, KMAX, K, KP;
    array V 1, V 2, SA 1, SA 2, SA 3, SA 4; real EPS, A, B, C, D;
  begin integer S, I, J; real S 1, S 2, S 3, S 4, W 1, W 2, W 3, W 4, AF,
      BF, CF, DF, EE; array V 1 P, V 2 P, V 3, V 4, V 5 [1: 300];
      EE:=EPS↑2; A:=AF:=₁₀50; B:=BF:=-₁₀50;
      for I:=1 step 1 until M do begin W 1:=V 1 [I]; W 2:=V 2 [I]; W 3:=W 1-A;
      if W 3<0 then A:=W 1; W 3:=W 1-B; if W 3>0 then B:=W 1; W 3:=W 2-AF;
      if W 3<0 then AF:=W 2; W 3:=W 2-BF;
      if W 3>0 then BF:=W 2 end argument in [A, B] function in [AF, BF];
      C:=1.0/(B-A); D:=-C*A; CF:=1.0/(BF-AF); DF:=-CF*AF;
      for J:=1 step 1 until M do
      begin V 1 P [J]:=C*V 1 [J]+D; V 2 P [J]:=CF*V 2 [J]+DF;
      end each datum has been mapped into the range [0, 1]*[0, 1];
      A:=C; B:=D; C:=1.0/CF; D:=-C*DF; S:=M-1;
      S 1:=FLOAT(M): SA 3 [0]:=SA 4[0]:=S 2:=I:=0;
      for J:=1 step 1 until M do
      begin S 2:=S 2+V 2 P[J]↑2; V 3[J]:=0; V 4[J]:=1.0
      end hereafter I represents the degree of polynomials,
      J location of data;
    L 1:S 3:=0; for J:=1 step 1 until M do S 3:=S 3+V 2 P[J]*V 4[J];
      W 1:=SA 2[I]:=S 3/S 1; S 4:=S 2-W 1↑2*S 1; K:=S-I;
      if K=0 then go to EXITDAME; W 1:=SA 1[I]:=S 4/FLOAT(K);
      if I=0 then go to L 2; W 2:=W 1-SA 1[I-1];
```

```
  if ABS(W 1)<EPS∧ABS(W 2)<EE then go to EXITOK;
  if I=KMAX then go to EXITDAME;
L 2:W 1:=0; for J:=1 step 1 until M do
  begin W 3:=V 4[J]; W 2:=V 5[J]:=V 1 P[J]*W 3; W 1:=W 1+W 2*W 3 end;
  W 3:=SA 3[I+1]:=W 1/S 1; W 2:=SA 4[I]; W 1:=0;
  for J:=1 step 1 until M do
  begin V 5[J]:=V 5[J]-W 3*V 4[J]-W 2*V 3[J]; W 1:=W 1+V 5[J]↑2;
  V 3[J]:=V 4[J]; V 4[J]:=V 5[J] end;
  I:=I+1; SA 4[I]:=W 1/S 1; S 1:=W 1; S 2:=S 4; go to L 1;
EXITDAME: K:=I; J:=0; SPACE(50); PRINT('NOTbCONVERGED');
  CRLF(5); go to EXIT;
EXITOK: K:=I; J:=1;
EXIT: SPACE(12); PRINT('DEGREE'); SPACE(23);
  PRINT ('ALPHA(I)'); SPACE(22); PRINT('BETA(I)');
  SPACE(20); PRINT('COEFFICIENT(I)'); CRLF(3);
  for I:=0 step1 until K do
  begin SPACE(10); PRINT('I='); SPACE(8); PRINT(2, I);
  SPACE(10); PRINT(12, SA 3[I]); SPACE(10);
  PRINT(12, SA 4[I]); SPACE(10);PRINT(12, SA 2[I]); CRLF(2)
  end; CRLF(5);
  SPACE(10); PRINT('INbNORMALIZEDbMAGNITUDE'); CRLF(3);
  SPACE(13); PRINT('NORM');SPACE(15);
  PRINT('SUMbOFbSQUARESbOFbRESIDUALS');
  SPACE(7); PRINT('STANDARDbDEVIATION');
  SPACE(9); PRINT(SQUAREbOFbTESTbFUNCTION'); CRLF(3);
  for I:=0 step 1 until K do
  begin W 1:=SA 1[I]; W 1:=ABS(W 1); W 3:=W 1*FLOAT(S-I);
  W 2:=SQRT(W 3); W 4:=W 2/SQRT(FLOAT(M)); SPACE(9);
  PRINT(4, W 2); SPACE(18); PRINT(4, W 3);
  SPACE(18); PRINT(4, W 4); SPACE(18); PRINT(4, W 1); CRLF(2)
  end; CRLF(5);
  SPACE(10); PRINT('XX=A*X+B, bXbANDbXXbAREbARGUMEMENTSbOFb
  REALbANDbNORMALIZEDbMAGNITUDEbRESPECTIVELY'); CRLF(3);
  SPACE(10); PRINT('F=C*FF+D, bFbANDbFFbAREbVALUESbOFbFUNCTIONb
  OFbREALbANDbNORMALIZEDbMAGNITUDEbRESPECTIVELY'); CRLF(5);
  SPACE(30); PRINT('A='); SPACE(8); PRINT(12, A); SPACE(10);
  PRINT('B='); SPACE(8); PRINT(12, B); CRLF(3);
  SPACE(30); PRINT('C='); SPACE(8);PRINT(12, C); SPACE(10);
  PRINT('D='); SPACE(8); PRINT(12, D); CRLF(5);
  PUNCH(2, K); PUNCH('/bORDERbOFbGENERATEDbPOLYNOMIAL');
  FEED; PUNCH(12, A); PUNCH(12, B); PUNCH(12, C); PUNCH(12, D);
  FEED; PUNCH('/bA, B, CbANDbD'); FEED;
  for I:=KP step 1 until K do
  begin PUNCH(12, SA 2[I]); PUNCH(12, SA 3[I]);
```

```
        PUNCH(12, SA 4[I]); PUNCH('/bORDER'); PUNCH(2, I);
        FEED end;
        PUNCH('b/COEFFICIENT(I), bALPHA(I)bANDbBETA(I)'); FEED;
        if J=0 then K:=0
     end; halt
procedure CALPO (K, A, B, C, D, SA 2, SA 3, SA 4, X, Y);
      value K, A, B, C, D, X; integer K; real A, B, C, D, X, Y; array SA 2, SA 3, SA 4;
    begin integer I; real XX, YY, V 3, V 4, V 5;
        XX:=A*X+B; I:=V 3:=YY:=0; V 4:=1.0;
      L 1: YY:=YY+SA 2[I]*V 4; if I=K then go to L 2;
        V 5:=(XX-SA 3[I+1])*V 4-SA 4[I]*V 3; V 3:=V 4; V 4:=V 5;
        I:=I+1; go to L 1;
      L 2: Y:=C*YY+D
    end; halt
```