プログラムのページ

担当　伊理正夫

6406. Volterra 型積分方程式の解法

三浦　大亮（東洋レーヨン株式会社）

1. 第2種の場合

$g(t)$ および $K(t, \tau)$ が既知で，$f(t)$ の函数値を区間 $[t_0, t_{\text{end}}]$ で求めるものとする．方程式は，

$$f(t)=g(t)+\int_{t_0}^{t}K(t, \tau)\cdot f(\tau)d\tau$$

ここで積分区間 $[t_0, t_{\text{end}}]$ を N 等分して，その間隔を h とすれば，

$$\left.\begin{aligned}&K_{i,j}=K(a+ih, a+jh)=K(t_i, \tau_j),\\&g_i=g(a+ih)=g(t_i),\\&f_i=f(a+ih)=f(t_i)\end{aligned}\right\}\quad(2)$$

とおくことができ，積分の項を

$$I_i=\int_{t_0}^{t_i}K(t, \tau)\cdot f(\tau)d\tau=h\sum_{j=0}^{i}a_{i,j}\cdot K_{i,j}\cdot f_j\quad(3)$$

とおくことができる．

(2)，(3) 式を (1) 式に代入すれば，直ちに次の解法によって逐次 f_i の系列を求めてゆくことができる．

$$\left.\begin{aligned}&f_0=g_0,\\&f_1=g_1+I_1,\\&\cdots\cdots\cdots\\&f_i=g_i+I_i,\\&\cdots\cdots\cdots.\end{aligned}\right\}\quad(4)$$

積分 I_i を求めるには，いくつかの方法があるが，最も簡単な方法として，梯形則を用いるものと，これを両端の差分で修正した Gregory の公式を用いるものがある．ここではこの計算法によるプログラムを記してある．

Gregory の公式は，$F(\tau)$ の積分

$$\begin{aligned}I_i\equiv\int_{t_0}^{t_i}F(\tau)\,d\tau=h\Big(&\frac{1}{2}F_0+F_1+F_2+\cdots\cdots\\&+\frac{1}{2}F_i\Big)-\frac{h^2}{12}(F_i'-F_0')\\&+\frac{h^4}{720}(F_i'''-F_0''')-\cdots\cdots\end{aligned}\quad(5)$$

に，

$$\left.\begin{aligned}&F_0'=\frac{1}{2h}(-3F_0+4F_1-F_2),\\&F_i'=\frac{1}{2h}(F_{i-2}-4F_{i-1}+3F_i)\end{aligned}\right\}\quad(6)$$

を代入して F''' 以上の項を切り捨てたものである．

以上から計算法は次のようになる．

$$\left.\begin{aligned}&\tilde{f}_1=f_{\frac{1}{2}}=\frac{1}{1-\frac{h}{4}K_{\frac{1}{2},\frac{1}{2}}}\Big\{g_{\frac{1}{2}}+\frac{h}{4}K_{\frac{1}{2},0}\cdot f_0\Big\},\\&f_1=\frac{1}{1-\frac{h}{6}K_{1,1}}\Big\{g_1+\frac{h}{6}(K_{1,0}\cdot f_0\\&\qquad+4K_{1,\frac{1}{2}}\cdot f_{\frac{1}{2}})\Big\},\\&\cdots\cdots\\&f_i=\frac{1}{1-\frac{h}{6}K_{i,i}}\{g_i+I_i-D_{i,i}\}.\\&f_0=g_0\quad,\end{aligned}\right\}\quad(7)$$

ただし，$\tilde{f}_1$ は最初のステップのみ $t=a+\frac{1}{2}h$ での値を求めて，f_1 の値の精度を上げようとしたものであり，また

$$\left.\begin{aligned}I_i&=h\Big(\sum_{j=0}^{i}a_{i,j}\cdot K_{i,j}\cdot f_j+\sum_{j=0}^{2}b_j\cdot K_{i,j}\cdot f_j\\&\qquad+\sum_{j=i-2}^{i}c_i\cdot K_{i,j}\cdot f_j\Big)\\&=h\sum_{j=0}^{i}(a_{i,j}+b_j+c_j)K_{i,j}\cdot f_j\\&=\sum_{j=0}^{i}D_{i,j},\\a_{i,0}&=a_{i,j}=\frac{1}{2},\ a_{i,j}=1\ (j\neq 0,\ j\neq i),\\b_0&=-3,\ b_1=4,\ b_2=-1,\ b_i=0\ (i\geqq 3),\\c_j&=b_{i-j},\end{aligned}\right\}\quad(8)$$

である．

2. 第1種の場合

$$g(t)=-\int_{t_0}^{t}K(t, \tau)\cdot f(\tau)d\tau\quad(9)$$

から $f(t)$ を求めるには，(7) 式を次のように変形すればよい．

$$\left.\begin{aligned}&f_0=g_0,\\&\tilde{f}_1=\frac{1}{-\frac{h}{4}K_{\frac{1}{2},\frac{1}{2}}}\Big\{g_{\frac{1}{2}}+\frac{h}{4}K_{\frac{1}{2},0}\cdot f_0\Big\},\\&f_1=\frac{1}{-\frac{h}{6}K_{1,1}}\Big\{g_1+\frac{h}{6}(K_{1,0}\cdot f_0\\&\qquad+4K_{1,\frac{1}{2}}\cdot f_{\frac{1}{2}})\Big\},\\&\cdots\cdots\end{aligned}\right\}\quad(10)$$

$$f_i=\frac{1}{-\frac{h}{6}K_{i,i}}\{g_i+I_i-D_{i,i}\}.$$

以下は USSC のために東洋レーヨン株式会社で開発された TORAY ALGOL によるプログラムである．一応第2種に対するもののみを示すが，第1種の場合は(10)式が(7)式と異なる部分のみ変更してやればよい．

プ　ロ　グ　ラ　ム

```
begin
comment  Solving the first & second kind of
         Volterra type integral equation;
comment  Give the problem by real procedure
         K(T, TAU) as the kernel and real
         procedure G(T) as the known
         function, and input the range of
         integration (TO, TEND) and the
         number of steps N;
  real  TO, T, TEND, AIJ, BJ, CJ, DIJ, DII,
        D, H, TAU;
  integer  I, J, N;
  array  F[0:100], B[0:3];
  real  procedure K(T, TAU);
        ここに核函数 K(t, τ) の計算式を書く ;
  real  procedure G(T);
        ここに与函数 g(t) の計算式を書く  ;
  ADVANCE PRINTER (4);
INIT:  READ CARD (TO, TEND, N);
       H:=(TEND-TO)/N;
         B[0]:=−0.125; B[1]:=0.16666667;
         B[2]:=−0.04166667; B[3]:=0;
       WRITE PRINTER ('INTEG-EQ',,,,
                          'T',,,,'F[T]');
       F[0]:=G[TO]; T:=TO+0.5*H;
       F[1]:=(G[T]+0.25*H*K(T, T)
             *F[0])/(1.0−0.25*H*K(T, T));
       T:=TO+H;
       F[1]:=(G[T]+H/6.0*(K(T, TO)*F[0]
            +4.0*K(T, TO+0.5*H)*F[1]))/
            (1.0−H/6.0*K(T, T));
       WRITE PRINTER (,TO,F[0]);
       WRITE PRINTER (,T,F[1]);
       for  I:=2 step 1 unti 1 N do
       begin  T:=T+H; D:=G[T];
              TAU:=TO;
         for  J:=0 step 1 unti 1 I do
           begin  AIJ:=if J=0 then 0.5 else if
             J=I then 0.5 else 1.0;
             BJ:=B[if J<3 then J else 3];
             CJ:=B[if I-J<3 then I-J else 3];
             DIJ:=(AIJ+BJ+CJ)*H;
             if I=J then DII:=1.0-DIJ*K(T, T)
               else D:=D+DIJ*K(T, TAU)
               *F[J];
             TAU:=TAU+H;
         end;
         F[I]:=D/DII;
         WRITE PRINTER (,T,F[I]);
  end
end
```

(注 1)　Nによる割算が mixed expression になっているが，TORAY ALGOL では実行のときには real に変換して行なわれる．

(注 2)　identifier の意味は T=t, TO=t_0, TEND=t_{end}, TAU=τ, F[I]=$f(t_i)$, G(T)=$g(t)$, K[T, TAU]=$K(t,\tau)$, H=h, AIJ=a_{ij}, BJ=b_j, CJ=c_j, DII=$1-h/6\ K_{i,i}$, である．

(注 3)　インプット・データは1枚のカードに当該の区間 [t_0, t_{end}] およびその分割数 N をパンチする．アウトプットは，t とそれに対する $f(t)$ の値が並べてプリントされる．

(注 4)　梯形則をそのまま積分に利用する場合は，BJ と CJ に 0 を assign すればよい．

(注 5)　READ CARD (…) はカードリーダから…への読み込みの procedure. ADVANCE PRINTER (……)は…だけプリンタの行送りをする procedure. WRITE PRINTER (……) は……をプリントする procedure (プリント内容は…の部分に ' ' で囲まれる部分があれば string となる).

3.　計算例

(i)　第2種方程式：

$$f(t)=2.0-e^{-t}+\int_0^t(t-\tau)e^{-t+\tau}\cdot f(\tau)d\tau.$$

(ii)　第1種方程式：

$$t+e^{-t}-1=\int_0^t(1+t-\tau)\cdot f(\tau)d\tau.$$

(i)　の計算結果

t	解析解(t+1)	梯形則による解	Gregory 法による解
0.0	1.0	1.000 000 0	1.000 000 0
0.1	1.1	1.099 999 8	1.099 999 8
0.2	1.2	1.199 409 8	1.199 998 4
0.3	1.3	1.299 158 9	1.299 994 9
0.4	1.4	1.398 928 6	1.399 991 7
0.5	1.5	1.498 714 2	1.499 988 7
0.6	1.6	1.598 512 4	1.599 985 8
0.7	1.7	1.698 320 2	1.699 983 2
0.8	1.8	1.798 134 7	1.799 980 7
0.9	1.9	1.897 954 1	1.899 978 2
1.0	2.0	1.997 777 3	1.999 976 0
1.1	2.1	2.097 601 8	2.099 973 5
1.2	2.2	2.197 427 7	2.199 971 2
1.3	2.3	2.297 253 3	2.299 969 0
1.4	2.4	2.397 078 0	2.399 966 8
1.5	2.5	2.496 901 4	2.499 964 3
1.6	2.6	2.596 722 8	2.599 962 2
1.7	2.7	2.696 542 1	2.699 960 0
1.8	2.8	2.796 358 3	2.799 957 7
1.9	2.9	2.896 172 2	2.899 955 5
2.0	3.0	2.995 982 5	2.999 953 2

(ii) の計算結果

t	解析解 te^{-t}	梯形則による解	Gregory 法による解	(−)Gregory 法による解
0.0	0.000 000 0	0.000 000 0	0.000 000 0	0.000 000 0
0.1	0.090 483 7	0.083 700 9	0.083 700 9	0.083 700 9
0.2	0.163 746 2	0.190 472 9	0.193 638 3	0.163 262 5
0.3	0.222 245 4	0.196 441 7	0.148 156 0	0.231 426 2
0.4	0.268 128 0	0.299 471 6	0.461 648 6	0.269 117 2
0.5	0.303 265 5	0.270 123 4	−0.201 265 5	0.302 757 8
0.6	0.329 287 2	0.366 734 1	1.644 356 1	0.332 507 7
0.7	0.347 549 2	0.307 230 2	−3.080 286 1	0.348 522 2
0.8	0.359 463 2	0.404 468 8	9.294 662 1	0.358 952 3
0.9	0.365 913 0	0.316 233 6	− 22.924 745	0.366 675 0
1.0	0.367 879 0	0.422 822 6	61.077 680	0.368 131 3
1.1	0.366 158 1	0.305 306 3	− 157.881 06	0.365 644 7
1.2	0.361 432 8	0.428 391 2	412.851 28	0.361 281 9
1.3	0.354 291 6	0.279 916 9	−1 074.848 7	0.354 091 2
1.4	0.345 235 8	0.426 960 0	2 802.987 5	0.344 712 9
1.5	0.334 700 0	0.243 885 1	−7 305.080 0	0.334 258 6
1.6	0.323 035 2	0.422 846 3	19 042.741	0.322 639 9
1.7	0.310 562 8	0.199 729 1	− 49 635.979	0.310 063 4
1.8	0.297 538 2	0.419 467 7	129 383.09	0.297 065 5
1.9	0.284 196 1	0.148 921 0	− 337 251.09	0.283 755 9
2.0	0.270 670 0	0.419 624 3	879 085.41	0.270 234 3

(注) (−) Gregory 法とは Gregory 法における修正項の C の符号を逆にしたもの

4. 結果の検討

第1種，第2種ともに $a_{i,j}$ を適当に選べば

$$g_i+h\sum_{j=0}^{i}a_{i,j}\cdot K_{i,j}\cdot f_j=0, \tag{11}$$

$$g_{i+1}+h\sum_{j=0}^{i+1}a_{i+1,j}\cdot K_{i+1,j}\cdot f_j=0 \tag{12}$$

と書ける．両式で f_i の誤差の部分についてのみ考えると，(12)−(11) 式から

$$\sum_{j=0}^{i}(a_{i+1,j}\cdot K_{i+1,j}-a_{i,j}\cdot K_{i,j})e_j + a_{i+1,i+1}\cdot K_{i+1,i+1}\cdot e_{i+1}=0 \tag{13}$$

が求まる．この例では任意の j について

$$\frac{K_{i+1,j}-K_{i,j}}{K_{i+2,j}-K_{i+1,j}}=1 \quad \left(\text{すなわち}\frac{\partial^2 K_j}{\partial t^2}=0\right) \tag{14}$$

であるので，$K_{i+2,i+1}-K_{i+1,i+1}=H$，$K_{i,i}/K_{i+1,i+1}=K$，$H/K_{i+1,i+1}=\eta$ とおき $a_{i,i}=a$ とすると，(13)式より

$$\begin{cases}\sum_{j=0}^{i}(a_{i+1,j}\cdot K_{i+1,j}-a_{i,j}\cdot K_{i,j}) + (1-a)\cdot K_{i,i}e_i+a\cdot K_{i+1,i+1}\cdot e_{i+1}=0,\\ \sum_{j=0}^{i}(a_{i+2,j}\cdot K_{i+2,j}-a_{i+1,j}\cdot K_{i+1,j}) + H\cdot e_{i+1}+(1-a)K_{i+1,i+1}\cdot e_{i+1} + a\cdot K_{i+2,i+2}\cdot e_{i+2}=0.\end{cases}$$

これから差をとって

$$(1-a)K\cdot e_i+(2a-1-\eta)e_{i+1}-\frac{a}{K}e_{i+2}=0 \tag{15}$$

が得られる．

1ステップごとの誤差の拡大係数を ρ とすれば

$$-\frac{a}{K}\rho^2+(2a-1-\eta)\rho+(1-a)K=0. \tag{16}$$

したがって

$$\rho=-\frac{K}{2a}\left\{(1-2a+\eta)\pm\sqrt{1+2(1-2a)\eta+\eta^2}\right\}. \tag{17}$$

(11)式および(12)式が誤差の拡大伝播について安定であるためには $|\rho|<1$ でなければならぬが，この条件を満足するためには次の式が成り立たなければならない．

(i) $a\leqq\dfrac{1+\eta}{2}$ のとき $a\geqq\dfrac{K(K+1+\eta)}{(K+1)^2}$ (18)

(ii) $a>\dfrac{1+\eta}{2}$ または $a<0$ のとき

$$a\geqq\frac{K(K-1-\eta)}{(K-1)^2}. \tag{19}$$

上の計算例では，第1種の場合には，梯形則にしても Gregory の公式を使用しても，$K=1$，$\eta=h$ として(18)も (19) の式も成立させない．また，Gregory の差分修正の一方の符号を逆にした場合は，(19) 式を満足することがわかる．

第2種の方程式では，一般に $a<0$ で，$K<1+\eta$ であれば安定である (convolution 型であれば常に安定)．

(昭和39年4月30日受付)

6407. Factor Analysis (Factor Loadings の最尤法による推定)

吉沢　正（東京大学工学部）

以下は最尤法による factor loadings の推定値を求める一つの計算法のプログラムである．実際に ALGOLIP（東大計算センター OKITAC 5090 の ALGOL）で書いて実験ずみである．factor loadings の推定については最尤法に限らず従来よりいろいろな方法が考えられており，最尤法による推定についても何通りかの計算手順を考えることができるが，どの方法が電子計算機に適しているかなどという点ではほとんど議論されていない．ここに述べる方法は固有値の計算を含む繰り返し法であるが，その収束性は証明されていない．しかし従来の論文などに見られる方法に比べて，簡単な初期条件から出発して解を求めることができるので便利な方法である．

ここでの方法はおもに Lawley[1] によるが，factor model の factors の分類その他の概念については Harman[2] がくわしい．まず（1）式の形のモデルを考える．