プログラムのページ

担当　森口繁一

6307. 爆発法によるあるゲームのシミュレーション

高瀬啓元（東大大学院数物系研究科）

庶民のギャンブル“パチンコ”を，次のようなゲームとして定式化する．玉を1個投入すると，一定の確率で“アタリ”で玉15個を得，入らねばなにももらえない．はじめに25個もって始め，玉が100個以上たまれば“勝ち”として，0個になれば破産としてゲームは打切られる．勝ちおよび破産の確率は，それぞれどのくらいかを求む．

この問題を，東大計算センターのOKITAC 5090のためのALGOLである“ALGOLIP”を使用してプログラムをした．50 step ごとに，勝ち，破産の確率を印刷し，所望の精度が得られるか，または500 step までで打切るものとしてある．

```
begin integer I, J, K;
  real R, RQ, DELTA, PA, PAI;
  array P, PP [0:100];
START: R:=READREAL;
  DELTA:=READREAL;
  if R=0.0 then go to EXIT;
INITIAL: for I:=0 step 1 until 100 do
  PP[I]:=0.0; PP[25]:=1.0;
  PA:=0.0; RQ:=1.0-R;
  J:=K:=0;
  CRLF(5);
  PRINTSTRING("␣␣R=␣");
  PRINTREAL(R);
  CRLF(2);
EXPLOS:J:=J+1; K:=K+1;
  for I:=0 step 1 until 14 do
  P[I]:=RQ*PP[I+1];
  for I:=15 step 1 until 99 do
  P[I]:=PP[I+1]+(PP[I-14]
    -PP[I+1])*R;
  P[0]:=P[0]+PP[0];
  P[100]:=R*PP[86];
  PA1:=0;
  for I:=87 step 1 until 100 do
  PA1:=PA1+P[I];
  PA:=PA+PA1*R;
  if K=50 then
   begin PRINTINTEGER(J);
     PRINTREAL(P[0]);
     PRINTREAL(PA);
     K:=0
   end;
  if J=500 then go to START;
  if P[0]+PA+DELTA>1.0 then
   begin PRINTINTEGER(J);
     PRINTREAL(P[0]);
     PRINTREAL(PA);
     go to START
   end;
  for I:=0 step 1 until 100 do
   PP[I]:=P[I];
  go to EXPLOS;
EXIT: end
```

ここに用いている解法は，いわゆる“爆発法”である．モンテカルロ法による解法の拡張であり，阪大教授杉山博氏の命名による．確率原理を用いる意図は同じだが，乱数を用いない，exact な近似法である．

n 発目の玉をはじいたとき，玉を I 個持っている確率を $P_n[I]$ とすると，次式が成り立つ．

$P_0[I]=0$ $(I\neq 25)$, $P_0[25]=1.0$,

$P_n[I]=P_{n-1}[I+1]\cdot(1-P)+P_{n-1}[I-14]\cdot P$

ただし，それまでに0か100をこした確率は，累積されるし，I が1～14および100では，少し変ってくる．このようにして両端にたまった量で，そのときまでにケリのつく確率がわかるし，その質量の和を1と比較すれば収束の様子，精度がわかる．プログラムでは，所望精度 Δ とともに確率 R をつぎつぎと読み込んでは求めている．コンパイルの時間は10数秒程度 R を0.06, 0.08, 0.10とかえて計算し，30分ぐらいかかった．ほぼ常識的な結果が得られた．現実と比較して，マニアの方から始めの仮定について異議がいくつかあるだろう．

たとえば，25個と，100個の大きさおよび常に独立に一定な確率とした仮定についてなどが問題となろ

う．前者については，数年前までは50円で25個程度とした初めの投資額が，なんでも倍増のおりから，50または100から初める人も多いだろうし，マニアともなれば，300～800 個ぐらいかせぐまで止めないことだろう．しかし上例で 30 分かかったように，かなり時間的制約をうけているので，上の程度に止めた．一方，“一定の確率”についても問題だが，吉村功氏（東大工）によると，彼および彼の友人の実験によってかなりこの仮定が満されているのが確められたという．

次に爆発法を用いない普通のランダムウォークによるシミュレーションのプログラムを記す．

（未テスト）

```
 begin real PO, PA, U;
    integer I, J, K, RN, NP, LAMDA;
   procedure RANDOM;
    begin
      RN:=RN*LAMDA;
      U:=FLOAT(RN)*1.0₁₀-12
    end RANDOM;
   LAMDA:=3.0↑21;
   RN:=READREAL;
START:R:=READREAL;
    if R=0.0 then go to EXIT;
    IMAX:=READINTEGER;
    B:=W:=0;
   for I:=1 step 1 until IMAX do
    begin NP:=25;
      for J:=1 step 1 until 500 do
       begin RANDOM;
         if U≧R then NP:=NP-1
         else  NP:=NP+14;
         if NP=0 then go to L1;
         if NP≧100 then go to L2;
       end;
      go to L3;
L1: B:=B+1;
      go to L3;
L2: W:=W+1;
L3: end;
   PO:=FLOAT(B)/FLOAT(IMAX);
   PA:=FLOAT(W)/FLOAT(IMAX);
   PRINTREAL(PO);
   PRINTREAL(PA);
 EXIT: end
```

プログラムはかなり簡単になるが，これから得られる結果はあくまで推定値にすぎず，バラツキを少なくするためには，かなりの回数をくり返さねばならない．

なお，OKITAC の ALGOLIP では，乱数の procedure が備えつけてないが，上の方法は比較的簡単だと思う．これで，整数は12ケタなので，$RN_i=\lambda \cdot RN_{i-1}(\mathrm{mode}\ 10^{12})$ に従って擬似乱数を発生していることになる．周期は 5×10^{10} である．

6215. 常微分方程式の数値積分法（自動的にきざみ幅を変化させる PC 法）のたしかめおよび修正

高田 勝（九州大学工学部）

本誌 **3**, 5, pp. 278～282 でのべた表題の ALGOL プログラムをテストするため，OKITAC-5090 A 用 ALGOLIP（東大計算センターで開発されたもの）で switch などを書直してテストした結果，プログラムに若干の誤を発見したので訂正する．

頁 行	訂正 事項
281↓14, 15 左欄	D:=1; …… to ENDSTART; までをとる．
同左↓17	X:=X+H/20 を X:=X+H/2.0
同左↓18	FKT; のあとに次のものを附け加える． COUNT:=ENTIER(HP/H)−1; go to WRITE; STEP: COUNT:=ENTIER(HP/H);
同左↓20	COUNT:=……をとる．
同左↓23	……−Y1[I]×4.0; とする
同左↑1	COUNT:=COUNT−1; をとる．
同左↑2	go to WRITE; を go to STEP; とする．
同左↑11	if D≠0 then begin S1:=2; go to CORRECT end;COUNT:=COUNT−1; とする．
↑17	S1:=2 をとる．
同右↓7	INTERPOL:X:=X−H×1.5; H:=H/2.0; FKT; とする．

以上のように訂正し，ALGOLIP 向きに書きかえて二元連立方程式の例についてテストした結果所望の結果を得た．

なおこの方法で用いた公式でテストした結果として，情報処理学会 37 年度第3回全国大会で横河電機の方より論文が発表され（同前刷 p. 3，ある一つの Predictor-Corrector 法について），この公式による時は誤った答が出ると指摘されたが，当方においてこの ALGOL プログラムにもとずき SIP でプログラムしたサブルーチンでテストした結果，そのようなことに