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$5\times10^{10}$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≠O **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

はならず，同前刷の第2図のようにかなり急峻な山ができるが，Tは必ずT_Cに収束し，同図で示されたような値に収束することはないことがわかった．同論文に発表されたものは，L=0 での導函数計算ですでに誤った値が出ており，プログラムミスのある結果が発表されたもののようである．

なおこの方法で用いた修正子は Henrici のいう意味では安定性が十分でないようであるが，$y'=Ay$ ($A<0$) などに対しては梯形則と同じ性質を示す．これらの点については若干の吟味を必要とすると思われたので，

$$y'=1-y \quad y(0)=0,\ (y=1-e-x)$$

について $x=16\,y$ まで収束判定定数 10^{-5} でテストしたが，誤差 0.8×10^{-3} の程度であって振動は起さず，1より大きい値にもならなかった．さらに，固定したきざみ幅で ($x=0(0.001)10,\ \varepsilon=10^{-8}$) テストしたが，これも同様振動はなく，誤差は 0.4×10^{-3} 程度であった．また

$$\left.\begin{array}{l} y''=-y,\ y(0)=0,\ y'(1)=1,\ y=\sin x, \\ y'=e^{-(x+y)}\ y(0)=0,\quad y=\log(2-e^{-x}) \\ y'=1-y^2\ y(0)=0,\ y=\tan h-x \end{array}\right\}(*)$$

についてもこのプログラムによってテストしてみたが，異常は認められなかった．ただ，この修正子

$$y_{n+1}=2\,y_n-y_{n-1}+(h/2)(y_{n+1}'-y_{n-1}')$$

の第3項（y' の項）からの影響がなくなると（$y''\cong0$ のような場合，つまりある一定値に収束するような場合もこれにふくまれる），前の2項のみから値が定まってしまう．したがって，丸めを上位（たとえば10^{-3}～10^{-5}程度）で行なうと，第3項以下の影響がなくなるので，$y_{n+1}=2\,y_n-y_{n-1}$ となり，誤差が増大し，$y=1-e^{-x}$ のような例では1を越す場合が生ずる．

以上の点から，修正子公式を

$$y_{n+1}=y_n+(h/12)(5\,y_{n+1}'+8\,y_n'-y_{n-1}')$$
$$T_C=-h^4y^{\mathrm{IV}}/24$$

に改めた方がよりよい精度を維持できるし，上記の危険もないと考えられたので，このように改めることを提案する．その時は，前述の訂正に次のものをつけ加える．

P281 左欄↓30 (CORRECT: より3行下)

```
Y 2:=(F[I]×5.0+F 1[I]×8.0−F[0])
    ×H/12.0+Y 1[I];
```

このように改めてテストした結果によると，前図の場合より精度もよくなった．一例を下に示す．

取上げた例は前記(*)を同時に解いたもので，プリント間隔 HP=0.1，収束判定定数 $\varepsilon=10^{-6}$ および 10^{-7} で $x=0$～40 まで解いた．$x=40$ における最終値のみを下表に示す．

$\varepsilon=10^{-6}$　　計算時間　10分30秒

$\sin x$	y(計算)=0.7452718875	誤差$\cong1.6\times10^{-4}$
$\log(2-e^{-x})$	〃　0.6931392730	〃　$\cong8\times10^{-6}$
$\tan h$	〃　0.9999999990	〃　$\cong10^{-9}$

$\varepsilon=10^{-7}$　　計算時間　24分

$\sin x$	y(計算)=0.7451336443	誤差$\cong2.0\times10^{-5}$
$\log(2-e^{-x})$	〃　0.6931458157	〃　$\cong1.3\times10^{-6}$
$\tan h$	〃　0.9999999980	〃　$\cong2.0\times10^{-9}$

なお使用計算機は OKITAC-5090 A であり，このテストには ALGOLIP そのものによるものでなく，SIP によるサブルーチンを用いた．また，誤差の計算をするためには函数サブルーチンを用いている．ALGOLIP によるプログラムでは

$$y''+xy'+y=0,\ y(0)=0,\ y'(0)=1$$

のみをテストして HP=0.1，$\varepsilon=10^{-4}$ および 10^{-5} で，$x=0$～1.0 を計算するのに3分であった．SIP によるサブルーチンは，命令，中間結果をふくめて約 $200+n\times5$ 語必要であるが，ALGOLIP によるときは命令だけで約460語になるが，読込みや，印刷形式のための命令がかなり入るので，実際の計算には約300語にもならないようである．したがって，ALGOLIPによっても大した時間の違いはないであろうと思われる．それだけコンパイラーがよくできているようである．

以上，先に発表したプログラムについての訂正を記するとともに，使用公式に対して誤った結果を発表された誤解をうけた向きもあるように思われるので，テスト結果を発表し，さらにこのプログラムに用いる公式に対し新たな提案を試みた次第である．