乱数発生の一方法について

植竹恒男

1 はじめに

できるだけ不規則に並べられた数の列は**乱数列**とよばれいろいろな場面で利用されている。たとえば，自然界や社会における統計的現象のモデルを作ったり，標本抽出を行なったりするために乱数列は不可欠なものである（文献〔6〕，〔7〕）。

乱数列は最初「表」の形で与えられた。JIS Z 9031 の「乱数表」はその一例である。しかしながら，応用される範囲がひろがるにつれて，大量の乱数が要求されるようになり，計算機によって乱数を生成する方法が考え出された。その場合，一定の手順によるほかないから，規則性をもたない数列を作ることは本来不可能である。

たとえば，一定の間隔で同じ形をくりかえす数列を避けることができないけれども，その周期をできるだけ大きなものにし，しかも，できるだけ不規則に見えるような数列を生成するのである。このような数列を**擬似乱数列**とよぶ。実は，乱数列の厳密な定義はまだ確立されていないので，現実に存在する乱数列はすべて擬似乱数列と考えるべきであろう。その意味で擬似乱数のことを単に乱数とよぶことが多い。

本稿では，まず，計算機による乱数生成の方法を概観する。その多くは実際に大量の乱数を必要とする場合は役に立たなくなる。そこで Knuth は，「最もよい」乱数を作るための条件をいくつか挙げた。第3節ではその条件を紹介し，この条件を満たしていてもよい乱数が得られない例を示

す。

第4節以下は乱数を生成する新しい方法とその基礎となる理論について述べる。現時点では，「最もよい」乱数列であることを確定することは困難であるけれども，「かなりよい」乱数が得られているように見える。なお，最後に，この数列の興味ある応用を付記した。

2 乱数生成の諸方法について

計算機による乱数生成法として最初のものは，1946年に，J.von Neumann が提案した**平方採中法** (middle-square method) である。これは，ある数を平方して，その中央の部分をとる——という操作をくりかえすことで乱数を生成する方法であるが，小さな周期をもった数列を生じやすいために今では用いられていない。

続いて，1949年に，D. H. Lehner が次のような方法を提案した。これは**乗算型合同式法** (multiplicative congruence method) とよばれる。

適当な定数 a とmを定めておき，ある数 X_0 から出発して，

$$X \leftarrow aX \mod m$$

という規則で順次に乱数Xを定めていく方法である。いいかえれば「aXをmでわった余りをXとする」ということをくりかえしていくのである。

しかし，こんにち最も一般的に用いられているのは，適当な定数a, c, mを定めておき，ある数 X_0 から出発して，

$$X \leftarrow (aX+c) \mod m$$

という規則で順次に乱数Xを作っていく方法で**混合型合同式法** (mixed congruence method) または**線型合同式法**とよばれる。

この方法において重要なのは，いうまでもなく定数 a, c, m の定めかたである。これについては今まで多くの研究がなされてきた。たとえば，$m=10^s$ とし，

$a=10^\alpha+1$ (aは2以上の整数)

cは2の倍数でも5の倍数でもない

とすれば，Xの周期は 10^s であることは理論的に知られている。もちろん，これだけの条件ではよい乱数が得られるとは限らない。実際に試みてみると，多くの書物に書いてある定数 a, c の値は，それほどよい乱数を生成しないのである。特に，大量の乱数列が欲しいときには適当な定数 a, c をみつけることは極めて困難である。

なお，乱数の検定法についても多くの方法が提案されている。まず，乱数が「一様に」分布していることを検定する方法としてはカイ自乗検定法が用いられるが，乱数を一つずつの数の列とみる場合，二つずつの数の組の列とみる場合，一般にn個の数の組の列の場合についてそれぞれ検定することは大変な計算量であり，しかも，すべての場合について良好な結果を得ることは極めて困難である。また，大量の乱数列を必要とする場合は，さらに Spectral test という検定が提案されている(文献〔1〕pp. 82参照)。

さらに，乱数の値が互いに独立であることの検定も行なわれるがこの場合も前述と同様な困難がつきまとう。

実際には，乱数列をどのような場面で用いるかによってそれに合った検定法をえらび，それで良好な結果が得られればよしとするほかないであろう。

3 Knuth の条件について

D. E. Knuth は過去における乱数発生法を検討した結果，次の方法を "nicest" で "simplest" な発生法として推賞している(文献〔1〕)。

任意の X_0 から出発して，次の帰納的な関係によって乱数Xを発生する。

$$X \longleftarrow (aX+c) \mod m$$

ここで，m, a, c は次のようにえらぶ。

i) mは計算機のレジスタの最大けた数がeであれば二進法の場合は，

$$m=2^e$$

十進法の場合は，

$$m=10^e$$

とする。

ii） aは $m=2^e$ であれば，mod 8 で5に等しいように，$m=10^e$ であれば，mod 200 で21に等しいようにえらぶ。

iii） aは m/100 より大きく，$m-\sqrt{m}$ より小さいようにえらぶ。また，aの二進表示または十進表示は単純な形あるいは規則正しい形であってはならない。たとえば，

$$a=3141592621$$

は適する。この数は（ii）の条件も満たしている。ただし，大量の乱数を発生する場合は Spectral test にかけてみる必要がある。このテストによれば上のaの値はかならずしもよくない。むしろ，

$$a=3141592221 \qquad m=2^{35}$$

のほうがよい。

iv） cは$m=2^e$ のときは奇数を，$m=10^e$ のときは5の倍数でないものをえらぶべきである。また，

$$\frac{c}{m} \fallingdotseq \frac{1}{2}-\frac{1}{6}\sqrt{3}$$

を満たすようにえらぶ。右辺は，方程式 $1-6x+6x^2=0$ の根である。

v） 乱数Xを用いるときは，その有効数字のうちで左端にあるものが主に貢献するように配慮すべきである。

Knuth の五つの条件は，混合型合同式法に関するこれまでの研究の集大成であるが，iii）でいっているように大量の乱数を得たい場合にはなお問題がある。しかも筆者が試みてみた結果では，たとえば，

$$a=3141592621 \qquad m=10^{10}$$

$$c=2113248651$$

X_0	1	2	3	4	5	6	……
周 期	67	18	18	18	57	18	……

はこれらの条件を満たしているにもかかわらず，次の表のように，ごく小さい周期でくりかえし，乱数列とは程遠いものになってしまう。

すなわち Knuth の五つの条件だけでは“よい”乱数を得るためにはなお不十分である。

4　新しい乱数生成法とその原理

筆者の乱数発生法は，次のようなものである。X＝1から始めて，

$$X \longleftarrow aX \qquad \mathrm{mod}\quad m$$

のような規則で乱数Xを順次に作っていく。ただし，aの値は，mによって定まる適当な数の集合のなかから任意にえらぶものとする。ただし，mは素数とする。この方法は一見，乗法型合同式法に似ているけれども，その場合は，Xは任意の X_0 から始め，aは一定であったことに注意しよう。

ところで，上のような操作を（m－1）回くりかえせばちょうどXが1になることが証明されるのである。したがって，その次からXは同じ形をくりかえす。いいかえれば，上のようにして得られた数列の周期はつねに（m－1）に等しい。ただし，これは最小の周期とは限らないが，もし，（m－1）より小さい周期があるとすれば，それはmの約数でなければならないことは明らかであろう。なお，mは素数であるから（m－1）はつねに偶数である。

さらに，任意の素数mに対して，最小の周期がちょうど（m－1）になるようなaが少なくとも一つ存在する。たとえばm＝37のときa＝2とすれば，次のような数列が得られる。

1，2，4，8，16，32，27，17，34，31，25，13，26，15，30，23，
9，18，36，35，33，29，21，5，10，20，3，6，12，24，11，22，
7，14，28，19，1

この数列の周期は36である。しかも，ふたたび1になるまでの36個の数はすべて互いに異なる。

上のようなaはただ一つとは限らない。たとえば，m＝37とすれば，

a＝2，5，13，15，17，18，19，20，22，24，32，35

の12個の値が条件に適する。

なお，m＝37のとき，その他の各aの値に対して生成される数列の最小周期は次のようになる。

a＝3，4，7，21，25，28，30　のとき　18

a＝8，14，23，29　のとき　12

a＝9，12，16，33，34　のとき　9

a＝11のとき　6，a＝6，31　のとき　4

a＝10，26のとき　3，a＝36　のとき　2

a＝1のときはもちろん1である。

次に，mが5から67までの素数値をとるとき，それぞれの場合について，最大周期（m－1）をもつようなaの値の個数kをしらべてみると次のようになる。

m	5	7	11	13	17	19	23	29	31	37	41	43	47	53	59	61	67
k	2	2	4	4	8	6	10	12	8	12	16	12	22	24	38	16	20

条件に適するaの個数kは上表ではつねに偶数であるがこのことは偶然ではない。上のようにしてできた数列の一つを，

$a_0(=1)$，a_1，a_2，……，a_{m-1}，$a_m(=1)$……(i)

とすれば，これを逆に並べた数列，

$a_m(=1)$，a_{m-1}，……，a_2，a_1，$a_0(=1)$……(i)′

と同じもの（に対応する条件に適したa）が，かならず存在するからである。そして，はじめの数列を定めるaの値と，あとの数列を定めるaの値との和はつねにm＋1に等しい。

たとえば，m＝37の場合，条件に適する12個のaの値のうち，2と35，5と32，13と24，15と22，17と20，18と19の和はいずれも37に等しい（互

いに乗法逆元)。

どの素数mについても，条件に適するaの値の集合はこのような一対のものの集合に類別されるから，条件に適するaの個数はつねに偶数になる。

また，数列(i)において，その第(m－1)/2項はつねにm－1になることもみちびかれる。

以上のような理論を基礎にして，いろいろなm，aに対する数列(i)のなかから乱数列に適するものをみつけてみようというのである。

5　条件に適するmとaをみつけること。

まず，十分大きな素数mに対して，前述の条件に適するaをいくつかみつけることができれば，それらによって生成された数列（i）は，2から(m－1)までの整数がかならず一回ずつ，しかも一回ずつしか表れないようなものであることが保証される。この範囲では，同じ数が二回表れることはあり得ないのである。したがって，あとは統計的な検定だけを行なえばよい。

たとえば，mとして素数1,000,003をえらべば，適当なaをえらぶことによって2から1,000,002までの約百万個の整数を重複なく生成することができるであろう。

次に，mを与えて任意の a_0 から始めて，各aに対応する最小周期を計算するプログラムを示す。

最初xを1とし，変換 x←ax(mod m) をくりかえして，ふたたびxが1になったところで，変換した回数kを印刷する。このようなことをaが(m－2)に達するまで続ける（フローチャート参照)。

計算の手順はこのように極めて単純であるが，mを大きくとると，この計算に莫大な時間を要する。たとえば，

$$m=1,000,003$$

とすれば，あるaの最小周期が1,000,002であることを判定するのに六十時間かかる。また，

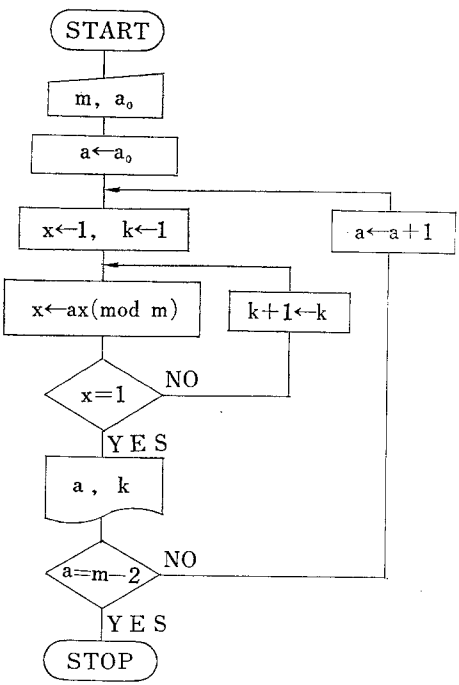
$m=99,991$（素数）

とすれば，同じ計算に約二時間かかる。aの最小周期が$(m-1)$に等しいかどうか判定するだけならば，上のプログラムで「$x=1$」の判定のあとに，

$$k>\frac{m-1}{2} \quad ?$$

という判定条件を入れればよい。これによって計算時間を半分に短縮できる。

なお，計算機としては，プログラムメモリつきの電卓RICOH—MONROE 1766を使用した。このような長時間にわたる計算のためには大型コンピュータより，この種の，いわゆる「パーソナルコンピュータ」の方が気兼ねなく使える。



最初はmとして，10^{10} 程度の素数をねらったのであるが，このくらいの order になると，上のプログラムで一つの a を判定するのに百年以上かかる！

そこで，われわれの計算機の能力からみて，10^5 の order の数のなかからみつけることを試みた。

m＝99991とし，aを10001から 10065 まで試み，条件に適する a として次の15個を得た。

10005，10009，10020，10026，10034，10037，10039，10041，10042，10043，10048，10050，10053，10059，10060

計算に要した時間はのべ約二百時間であった。

6　検定

前項で得たmと a の値に対して乱数としての検定を行なった。すでに述べたようにすべての検定を行なうことは困難なので，多量に，しかも，1個ずつの乱数として用いる場合に限定して行なった。次に示した表は，その一部で，Xを6個の数，

$$\{1,\ 2,\ 3,\ 4,\ 5,\ 6\}$$

が「一様に」分布するように変換したとき，1が出る累積相対度数を計算したものである。これは，1個のさいころを振る実験の simulation とみることもできる。Xは，

$$\left[\frac{6X}{m}+1\right]\quad ([\ ]\text{はガウスの記号})$$

のように変換し，一万回の「試行」を行なった。

参考までに（＊）の行に従来の方法で生成した乱数列の一例をあげた。この場合は，混合型合同式法により，

$$m=10^{10},\quad a=101,\quad c=1$$

とし，Xは2から始めた。

この結果からみる限り従来試みたいろいろな乱数生成法にくらべて特に

劣ったところはみられない。

（なお，一つのaについて，一万回試行するのに約二時間を要した。）

a	1000	2000	3000	4000	5000	6000	7000	8000	9000	10000
10005	.160	.159	.151	.159	.163	.159	.160	.159	.159	.159
10009	.172	.169	.167	.164	.164	.166	.164	.163	.164	.166
10020	.178	.171	.171	.170	.168	.169	.170	.170	.169	.169
10026	.181	.167	.171	.163	.164	.163	.165	.165	.167	.167
10034	.136	.150	.160	.160	.160	.161	.162	.163	.162	.164
10037	.174	.160	.166	.163	.164	.166	.167	.164	.167	.166
10039	.174	.176	.179	.179	.174	.173	.173	.171	.170	.170
10041	.173	.174	.171	.169	.170	.167	.168	.169	.167	.170
10042	.152	.167	.166	.165	.166	.167	.166	.166	.164	.163
10043	.135	.143	.151	.156	.158	.160	.157	.156	.160	.159
10048	.158	.166	.164	.162	.164	.168	.170	.171	.169	.169
10050	.153	.159	.164	.167	.168	.167	.169	.167	.167	.167
10053	.172	.167	.173	.173	.173	.171	.169	.168	.166	.164
10059	.163	.169	.168	.167	.174	.172	.169	.168	.168	.169
(*)	.157	.160	.160	.166	.166	.172	.172	.170	.167	.170

7 おわりに

乱数生成とその利用に関する研究はむしろ学際的なテーマであり，理論的な側面（主として整数論）と実験的な側面（コンピュータとその周辺技術）をもつ。本稿は理論的な面については厳密な証明は省略し，その結果だけを述べるにとどめ，実験的な面に重点を置いた。

乱数生成法については本文に述べたほかにもすでにいろいろな方法が提案されている（たとえば，文献〔2〕）。

しかし，実際問題としては，なるべく単純な方法で，目的に適したよい乱数を作ることが要求されるので，本文で述べた方法が採用されてきたのであろう。本稿の方法も単純でよい乱数を生成する方法として，試みる価値があると思う。すでにみたように，十分大きな周期をもつような定数をみつける計算は容易でないけれども，将来，今までよりもはるかに高速度

の計算機が得られるならば，さらに試みてみたいと思う。

〔付記〕

本稿で得た数列には別な用途がある。この数列を円周上に目盛ることによって，自然数の乗除用の計算尺を作ることができるのである。ただし，ここでいう「乗除」は剰余系におけるものである。次にその一例を示す。

m＝37，a＝15とすれば数列，

1，15，3，8，9，24，27，35，7，31，21，19，26，20，4，23，12，32，36，22，34，29，38，13，10，2，30，6，16，18，11，17，33，14，25，5，1，……

を得る。円周を36等分して，この数列を目盛ったものを図のように二つ作り，同心円の形で互いに回転できるようにすれば，目はずれのない乗除の計算尺ができる。たとえば，4×8の計算は次のように行なう。

(1) 大円の目盛1と小円の目盛4を合わせる。

(2) 大円の目盛8に合う，小円の目盛を読む。

このような計算ができる理由は次のように説明できる。

4は1にa＝15を順次に十四回かけて得られたものである。また，8は1に15を三回かけて得られたものであるから，(2)の操作によって，1にaを，

14＋3＝17（回）

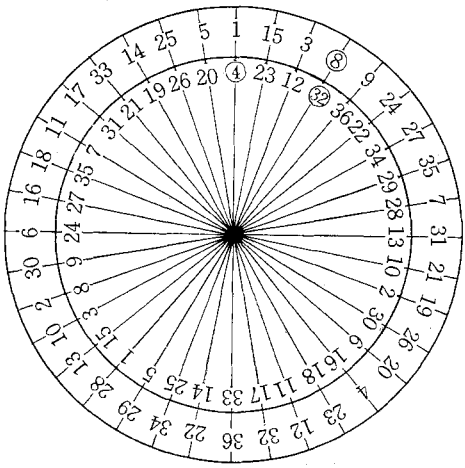
かけて得た数32が求める積になるのである。

上の例のように，積が37より小さい場合は，通常の乗法と同じになるが，図をみてもわかるように，一般には，

4×24→22　　mod　37

のような"乗法"が行なわれるのである。したがって，mを十分大きくとっておけば，通常の乗除法のための計算尺が作れるはずである。ただし，この計算尺では，数の目盛が不規則であることが実用上欠点になるだろう。

本文はそのことを逆に利用したことになる。



参考文献

1. D. E. Knuth, Seminumerircal Algorithms (The Art of Computer Programming Vol. 2), 1971, Addison-Wesley.
2. F. Gebhardt, "Generating Pseudo-Random Numbers by Shuffling a Fibonacci Sequence", Math. comp. 21, 1967, pp. 708-709.
3. T. E. Hull & A. R. Dobell, "Random Number Generators", SIAM Review 4, 1962, pp. 230-254.
4. M. Donald MacLaren & G. Marsaglia, "Uniform Random Generators" JACM 12, 1965, pp. 83-89.
5. 日本規格協会，統計数値表，1972，pp. 418-447.
6. 山内二郎・森口繁一・一松信，電子計算機のための数値計算法Ⅰ（数理科学シリーズ1），1965，培風館.
7. 岸田孝一・藤井良治，シミュレーションの演習，1974，産報.

（筆者は本学助教授・数学）