乱数生成に関する最近の話題

手塚　集

1. はじめに

シミュレーションによる推定値の精度は，原理的には使った乱数の個数が多いほどあがるので，乱数1個あたりの生成速度は速ければ速いほどよい．そのため通常は計算機の基本オペレーションを数回使って1つの乱数が発生されるようになっている．その反面，生成される系列はランダムな（一様な）性質を持たなければならず，簡単な算法によって真にランダムな系列を作ることは困難なので，この矛盾した要求を満たすためのさまざまな工夫が必要になってくる．生成速度をすこしでもあげるためには，使う計算機のアーキテクチャやプログラム言語に依存したノウハウも使いこなさなければならないし，その一方で，生成される系列のランダム性（一様性）を理論的に解析し，保証するために整数論等の数学を駆使しなければならない．たとえば，1960年代によく用いられた乱数サブルーチンに次のようなものがある．

$$X_i = 65539\, X_{i-1} \pmod{2^{32}}$$

この方法を例にとると，(mod 2^{32}) の部分は，（1）計算機の語長が32ビットであることと（2）乗算結果の (mod 2^{32}) の計算が，FORTRAN ではオーバーフローによって自然に行なわれることからきている．また（3）$65539 = 2^{16}+2+1$ なので乗算の計算が2回のシフトと加算で置き換えられる，といった細かな計算機のアーキテクチャやプログラム言語に依存した工夫もこらされている．線形合同法と呼ばれるこの種の方法は理論的にも非常に深く調べられており現在でも愛好者は多い．しかし，この方法では周期が計算機の語長で制限されるため長い周期の系列を高速に生成するのがむずかしい．

一方，この10年で計算機の処理速度は飛躍的に向上し，また広くワークステーションも普及したため，大規模シミュレーションが盛んに行なわれるようになり，それに適した長い周期の乱数系列を高速に生成する方法が必要になってきた．その1つが GFSR 乱数と呼ばれる方法で，最近では多くの人（特に計算物理の研究者）がこの方法を用いるようになっている．この方法はかなり以前に提案されていたにもかかわらず，理論が伴なわなかったので賛否両論があったが，ここ10年でいちじるしく理論が発展し，少なくとも線形合同法に関する理論体系に匹敵するものが作り上げられた．その主な背景としては，符号理論等との関連から有限体に関する知識が純粋数学者のみならず技術者への間にも広まったことや，80年代前半に Lidl と Niederreiter によって有限体に関する大著［5］がまとめられたことなどが考えられる．

1980年代のもう1つ大きな出来事として，計算機アーキテクチャに従来とは異なるものが現われるようになったことがある．通常は計算機の基本オペレーションを数回使って1つの乱数が生成されるようになっているが，計算機の基本オペレーションといっても計算機のアーキテクチャが変われば当然変わっていくわけで，それぞれのアーキテクチャに適した乱数生成アルゴリズムが考えられなければならない．その意味で，最近のアーキテクチャとしてとりわけ重要なものである VLSI と並列処理計算機に対しても，それにふさわしい乱数生成アルゴリズムが最近いくつか考えられている．なかでもセル・オートマトンによる方法は VLSI でのインプリメンテーションに向いているし，また Matrix Generator は並列処理に適しているといわれている．

そこで本稿では，GFSR 乱数に関する理論的な成果について初めに解説し，それとの関連としてセル・オートマトンによる方法と Matrix Generator について紹介したい．

2. GFSR 乱数の理論

GFSR 列は次のように定義される．

$$\alpha, T\alpha, \cdots, T^i\alpha, \cdots$$

ここで，α は2値 (0, 1) の p 次元ベクトル，T は2

てづか　しゅう　日本アイ・ビー・エム㈱　東京基礎研究所
〒102 千代田区三番町5—19

値 (0, 1) の $p\times p$ 行列で，演算は modulo 2 で行なわれるとする．ベクトル $T^i\alpha$ を $(b_1,\cdots,b_p)$ と書くと一様乱数 u_i は次のように定義される．

$$u_i=\sum_{i=1}^{p} b_i\,2^{-i}.$$

Tは乱数の周期を最大にするために，その特性多項式が原始的になるものが使われる．また，p はふつう 500 ぐらいなので u_i は通常上位32ビット程度で切り捨てることになる．ソフトウェアでこの方法を高速に実現するには，次のようにする．Tの特性多項式が原始3項式のとき，$p>q$ としてTは，

$$T^p+T^q+I=O \pmod 2$$

を満たす．ここで，Iは単位行列，Oは零行列である．このことから u_i は漸化式

$$u_i=u_{i-p+q}\ \mathrm{XOR}\ u_{i-p}$$

を満たし，乱数1個が XOR (bitwise exclusive-or) 1回で生成できることになる．u_i の各ビットは同じ漸化式

$$a_i=a_{i-p+q}+a_{i-p} \pmod 2$$

にしたがっており，$u_i, i=1,2,\cdots$，は，

$$u_i=\alpha_{j_1+i}\,2^{-1}+\cdots+\alpha_{jq+i}\,2^{-p} \qquad (1)$$

とも表わせる．ここで，$j_i, i=1,\cdots p$，は $1\le j_i\le 2^p-1$ なる整数である．Lewis と Payne が定義したオリジナルな GFSR 列は，上において $j_i=id$ としたものである．ここで，dは定数である．彼らは性質のよい GFSR 列を得るためのノウハウとしてdを $100p$ 以上にとることを勧めた．しかし，これにはなんらの理論的根拠もなく，後に筆者ら［2］によって GFSR 列が多次元一様分布 (k-distribution) するための必要十分条件が得られた．

(1) でみるように，GFSR 列では異なった重みが a_i にかかって一様乱数が作られているので，その理論的解析は線形合同法より難しくなる．Marsaglia の言葉を借りると，その難しさは，GFSR 列の構造が，“長年かかって幾重にも折り畳まれ歪められていった地表の堆積層”のようになっていることにあると考えられていた．しかし，最近になって［8］，オリジナルな GFSR 列は GF (2) 上の Laurent 級数 $S(x)=\sum_{j=1}^{\infty} s_j x^{-j}$ に S (2) の先頭の p ビットからなる2進小数を対応させる写像 σ_p を用いて次のように書けることがわかった．

$$u_i=\sigma_p(f_i(x)/M(x))$$

ここで多項式列 $f_i(x), i=1,2,\cdots$，は

$$f_i(x)=g(x)f_{i-1}(x) \pmod{M(x)}$$

により生成する．また，$M(x)$ は既約多項式で，$g(x)$ は

$$g^p(x)+g^q(x)+1=0 \pmod{M(x)}$$

を満たすとする．つまり，GFSR 列は GF (2) 上の多項式演算に関する“線形合同法”と見なせるのである．この表現は GFSR 列の理論を作るうえで非常に役立つ．たとえば，$u_i, i=1,2,\cdots$，の引き続く k 個からなるベクトル $(u_i,u_{i+1},\cdots,u_{i+k-1})$ について考えると，このベクトルは GF (2) 上の Laurent 級数による体に関するラティス上の点として考えることができる．つまり，ある基底 $e_1,e_2,\cdots,e_k$ の多項式による線形結合に対応づけられる．最も簡単に見つかる基底 $e_1,e_2,\cdots,e_k$ を1つあげると次のものがある．

$$e_1=\frac{1}{M(x)}(1,g(x),\cdots,g^{k-1}(x)),$$

$$e_2=(0,1,0,\cdots,0),$$

$$\cdots\cdots$$

$$e_k=(0,0,\cdots,0,1).$$

さて，図1でラティス上の点として考えることのメリットを説明しよう．図1-(a) は周期63の GFSR 列のひき続く2個 (u_i,u_{i+1}) のプロットである．図1-(b) はこの点列がどのような関係になっているかを示したものである．じつは，どの四角形の頂点 (P_1,P_2,P_3,P_4) も次の関係を満たしている．

$$P_2=P_1\ \mathrm{XOR}\ \sigma_p(e_1)$$

$$P_3=P_1\ \mathrm{XOR}\ \sigma_p(e_2)$$

$$P_4=P_1\ \mathrm{XOR}\ \sigma_p(e_1)\ \mathrm{XOR}\ \sigma_p(e_2).$$

ここで，XOR は各座標ごとにとるものとする．つまり，GFSR 列によるベクトル $(u_i,u_{i+1},\cdots,u_{i+k-1})$ の振舞いは，わずか k 個の基底ベクトル $e_1,e_2,\cdots,e_k$ で特徴づけられるのである．

GFSR 乱数の理論は，この後さらに (1) k-distribution, (2) spectral test, (3) discrepancy, (4) autocorrelation 等の性質解明へと展開していくのであるが，ここでは割愛する．(1), (2) については，従来の線形合同法の理論に見事に対応のつくものが得られる．また，計算の手間の点からみても有効なアルゴリズムが導ける．(3) に関連した話題としては，本誌伏見氏の解説を参照されたい．

3. セル・オートマトンによる方法

1次元セル・オートマトンによる乱数生成が Wolfram［9］によって精力的に研究されてから，最近方々でこの方式が用いられるようになってきている．特に，この方式のメリットは VLSI でのインプリメンテ

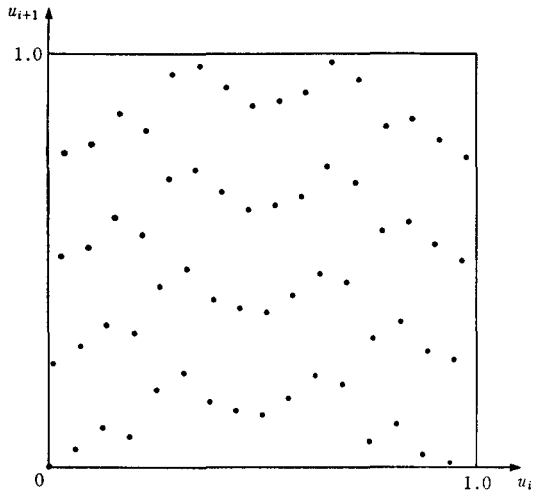

図 1（a） GFSR 列の例（周期63）

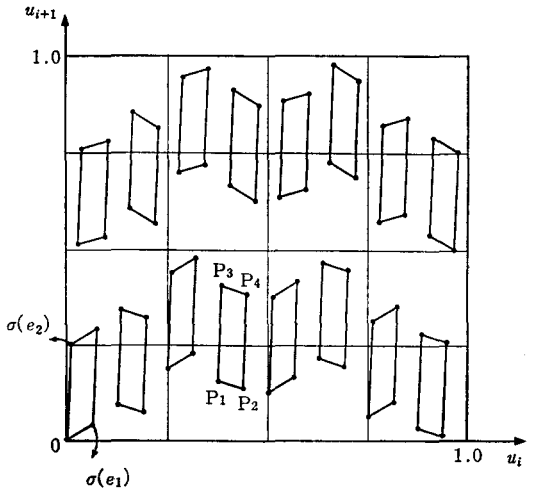

図 1（b） GFSR 列のラティス構造

ーションに向いているところにあり，専用目的のモンテカルロ・シミュレータにチップとして組み込んでしまう場合とか，VLSI サーキットテストのためのランダムパターン生成ハードウェアなどで需要がある．

Wolfram が調べたものは次のようなものである．n 個のセル $c_1, \cdots, c_n$ が1次元上にならんでいるものとし，各セルは2つの状態 (0, 1) をとるとする．セルの次の時刻における状態は下のルールで決められる．

$$c_i = \phi(c_{i-1}, c_i, c_{i+1}), \quad i=2, \cdots, n-1$$

ここでは，次の時刻における n 個のセルの状態は現在の状態から同時に上の式により変わるものとする．また，c_1 および c_n については次の境界条件が適用される．

$$c_1 = \phi(c_n, c_1, c_2), \quad c_n = \phi(c_{n-1}, c_n, c_1)$$

図 2 はこの境界条件を模式的に示したものである．

さてルール ϕ についてであるが，Wolfram はすべての場合をつくして実験的にその性質を調べた．簡単にわかるように ϕ は3ビットから1ビットへのマッピングなので，その総数は $2^{2^3}=256$ 通りである．また ϕ は GF（2）上の代数式として AND（乗算）と XOR（加算）を使って一意に書けることから“線形”，“非線形”に分けられる．たとえば，

$$\phi(x, y, z) = x \text{ XOR } y \text{ XOR } z \qquad (2)$$

は線形だし，

$$\phi(x, y, z) = x \text{ AND } y \text{ XOR } z$$

のように AND が含まれると非線形ということになる．一般に，非線形なルールのほうがセルの状態を乱数として使う場合ランダムらしく見えるが，周期がどうなるかという理論さえ確立しておらず，不明な点が多い．そのため周期を理論的に決められる線形の場合がよく使われる．**図 3** は2次元セル・オートマトンの例である．この場合にも1次元と似たようなルールが考えられている．

さて，GFSR 列との関係を考えると，最もよく使われる線形セル・オートマトンの場合，その状態を2値ベクトルと見なすと，行列表現できて

$$\alpha, T\alpha, \cdots, T^i\alpha, \cdots$$

のように書ける，たとえば先の（2）は

$$T = \begin{bmatrix} 1 & 1 & & & 1 \\ 1 & 1 & 1 & & \\ & \ddots & \ddots & & \\ & & \ddots & \ddots & \\ 1 & & & 1 & 1 \end{bmatrix}$$

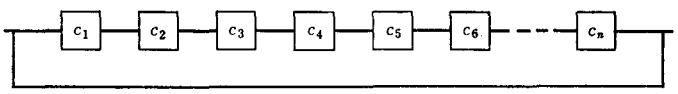

図 2 1次元・セルオートマトン

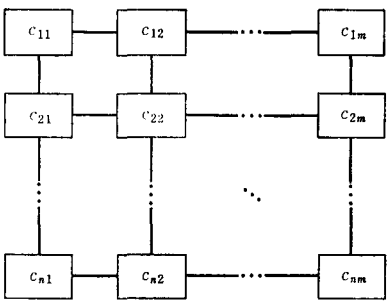

図 3 2次元・セルオートマトン

となっている．これはすなわち GFSR 列である．したがって，線形セル・オートマトンによる乱数の一様性の解析には GFSR 乱数の理論が適用できる．

4. Matrix Generator

Matrix Generator は線形合同法の“行列”版で，一般的には次のように書ける．

$$X_i=AX_{i-1}+B \pmod m$$

ここで A は $k\times k$ の行列，B は k 次元ベクトル，m は整数である．生成される $X_i, i=1,2,\cdots,$ は k 次元ベクトルの列である．m は通常 2^{32} 程度の大きさの整数なので X_i の各成分は 1 つの乱数として扱える．並列計算機やベクトル計算機では X_i を（すなわち，k 個の乱数を）一度に生産することができる．以下では $B=0$ として話しを進めることにする．

まず，広く用いられるのが m が素数の場合である．この時，A の特性多項式が原始的であれば ベクトルの列 X_i の周期は m^k-1 となる．つまり，$(0,\cdots,0)$ を除くすべてのベクトルが 1 周期上で現われることになる．したがって k 次元以下では一様分布しているといえる．しかし，k より高い次元になると従来の線形合同法の場合と同様なラティス構造があることが知られている．このことから，従来の線形合同法の場合に用いられていたスペクトルテストがパラメータAの選択に応用できることになる．m が素数でない場合，特に 2 の冪乗の場合も，使われてはいるが，周期およびラティス構造の点で劣るためあまり勧められない．

GFSR 列との違いは次の点である．GFSR 列では GF（2）上のベクトル空間で考えたのに対して，ここでは GF（m）をあつかう．m は通常 3^{32} 程度の素数である．ただ，一様乱数の定義が異なり，GFSR 列では，ベクトル 1 個が 1 つの乱数に変換されたが，ここではベクトルの 1 つ 1 つの成分がすでに十分な精度を持っているため，m で各成分を正規化するだけでよい．この違いのために一様分布の理論は違ったものになっている．しかし，ソフトウェアによる高速な生成法については，先の GFSR 列に用いた手法が使える，つまり A の特性多項式が原始 3 項式になるようにすると，$k>s$ として $\boldsymbol{A}$ は，

$$A^k+bA^s+cI=O$$

を満足する（b,c は定数）．そのため，ベクトル $X_i=(x_i^{(1)},\cdots,x_i^{(k)})$ の 1 つ 1 つの成分が

$$x_i^{(j)}=bx_{i-k+s}^{(j)}cx_{i-k}^{(j)} \pmod m, j=1,\cdots,k,$$

のような簡単な漸化式で生成される．GF(m) 上の原始 3 項式としてすでに見つかったいくつかについては［4］に表があるが，今後はこのような表をさらに整備することが必要になるだろう．

5. まとめ

本来，GFSR 乱数は線形合同法の代案として考えられたものであり，その定義からは表面上全く異なるタイプの方法と思えたにもかかわらず，じつは代数系を変えてみるとこれも“線形合同法”になっていた，という事実は乱数研究の奥深さを示しているようでじつに面白い．ただし，この両者は理論的には共通しているとはいっても，インプリメンテーション上は大きな違いがある．前にも述べたとおり，従来の線形合同法は長い周期の乱数を高速に生成するには適さず，一方，GFSR 乱数では，任意の長さの周期列を非常に高速に生成できる．そして周期が大きくなると，実際にはその上位ビットしか用いないため，パラメータさえ正しく選べば（“よい”ラティスであれば），ラティス構造の影響からも逃れられることになる．

以上ざっとシミュレーションに使う乱数に関して最近の話題を拾ってみたが，この他，暗号やランダマイズドアルゴリズムに関連して乱数は研究されており，まだまだいろいろな課題が残っているというのが現状である．最後にここで扱ったテーマに関する主な参考文献として，GFSR 乱数については［1, 3, 6, 7］，セル・オートマトン関連では［9］，Matrix Generator では，［3, 6］等をあげておく．

謝　辞

本稿をまとめるにあたり，非常に有益なコメントをくださった東京大学伏見正則教授に感謝します．

参考文献

［1］ 伏見正則：乱数．東大出版会，1989.

［2］ Fushimi, M. and Tezuka, S.: “The k-distribution of generalized feedback shift register pseudorandom numbers”, *Comm. ACM*, Vol, 26 (1983), 516-523.

［3］ L'Ecuyer, P.: “Random numbers for simulation”, *Comm. ACM*, Vol. 33, No. 10 (1990), 85-97.

［4］ L'Ecuyer, P. and Blouim, F.: “Linear congruential generators of order $k>1$”, *Proc. of*

the 1988 Winter Simulation Conference, IEEE Press, (1988), 432-439.

[5] Lidl, R. and Niederreiter, H.: *Finite Fields*, Cambridge Univ. Press, 1983.

[6] Niederreiter, H.: "Recent trends in random number and random vector generation", *Annals of O. R.*, Vol. 31 (1991), 323-346.

[7] Tezuka, S.: "Walsh-spectral test for GFSR pseudorandom number generators", *Comm. ACM.*, Vol. 30 (1987), 731-735.

[8] Tezuka, S.: "Lattice structure of pseudo-random sequences from shift register generators", *Proc. of the 1990 Winter Simulation Conference*, IEEE Press, (1990), 266-269.

[9] Wolfram, S.: "Random sequence generation by cellular automata", *Advances in Appl. Math.* Vol. 7 (1986), 123-169.