数理解析研究所講究録
915巻 1995年 146-156

無理数回転による擬似乱数生成

九州大学大学院数理学研究科　　杉田　洋 (Hiroshi SUGITA)

Abstract. $\{0,1\}$-値数列

$$X_n^{(m)}(\omega_0) := \sum_{i=1}^{m} d_i(\{\omega_0 + n\alpha\}) \pmod 2, \qquad \omega_0 \in [0,1), \qquad n = 0,1,\ldots$$

(α は無理数、$\{x\}$ は実数 x の小数部分、また $d_i(x)$ は x の 2 進小数展開における第 i 桁を表す) は任意の初期値 $\omega_0 \in [0,1)$ に対して、$m \in \mathbf{N}$ が十分大きいとき非常にランダムにふるまい、擬似乱数として用いることができる。さらに各 m について任意次元の周辺分布 (相対度数分布) が具体的に計算できるため、統計的検定を待たずに擬似乱数の評価が可能である。そうした分布の事前評価によれば、実用には $\alpha = (\sqrt{5}-1)/2$ のとき $m = 90$ 程度で十分であることがわかる。

0. 序

統計物理学やニューロネットワークにおける問題のように非常に自由度の大きい系の数値解析には通常の決定論的アルゴリズムは能率が悪いばかりか、事実上不可能なことが多い。そのような場合に広く利用されるのが「モンテカルロ法」と総称されるランダムサンプリングによる手法である。そしてモンテカルロ法の実践にあたって重要な意味を持つのが「乱数」である。

乱数の厳密な定義は Kolmogoroff、Solomonoff、von Mises や Martin-Löf らの仕事によって理論的には確立している ([5] およびその参考文献を見よ)。大雑把に述べると乱数とは「非常に計算量の多いアルゴリズムでなければ生成できない数列」であって「あらゆる統計的検定に合格する」と言う性質を持つ ([13])。換言すれば、現実に計算機によって実行可能な程度のアルゴリズムからは真の乱数を得ることは原理的に不可能なわけである[1]。従って我々は近似的なもの、すなわち**擬似乱数**で我慢しなければならない。

「近似」と言うとき、一般に次の二点は数値解析において是非とも要請したい条件である。

1. 誤差について (何らかの意味で) 定量的に述べることができること。
2. 誤差をいくらでも小さくするアルゴリズムが (原理的に) 存在すること。

本稿で紹介する新しい擬似乱数生成法は上の二つの要請に対する一つの解答を与えるものである。我々の方法では擬似乱数の誤差を確率論の言葉でもって表わし、それが 0 に収束するような実用的な擬似乱数生成アルゴリズムの列を構成する[2]。

本稿の構成を明らかにしておく。第 1 節では我々の擬似乱数の定義と主定理を述べる。第 2 節では力学系による擬似乱数生成の一般論を述べ、我々の方法の位置付けを行う。この節は他の部分と独立しており、急ぐ読者は飛ばして読まれてもよい。第 3 節では我々の擬似乱数の多次元分

[1] そのため、ランダムに起こる物理現象を観測してそのデータを乱数として利用することも行われている。

[2] 理論的には [6] や [7] において構成されているが、それらは実用的ではない。一方、実際に用いられている既存の擬似乱数生成法 ([8][12][20] などを見よ) は主として代数的手法に基づいており、確率的手法を取り入れていない。

布を求めるためのアルゴリズムを紹介する。第4節ではその前の節で与えたアルゴリズムによって多次元分布を求め、その一様性に関して評価する。第5節は我々の擬似乱数を生成するためのCによるプログラム例を掲載した。第6節は参考文献表である。

1. 擬似乱数の定義と極限定理

乱数のモデルとしては原理的にも実際的にも2点集合 $\{0,1\}$ に値をとる平均 $1/2$ の独立同分布確率変数列、すなわち**硬貨投げの確率過程**、を考えれば十分である。以下ではもっぱら、このような乱数のみを扱う。

はじめに、標題の擬似乱数の生成アルゴリズムを述べよう。(Ω, P) を Lebesgue 確率空間、すなわち $\Omega = [0,1)$、$P =$ Lebesgue 測度とする。$\omega \in \Omega$ に対して $d_i(\omega)$ は ω の2進小数展開における小数点以下第 i 桁を表すこととし、各 $m \in \mathbf{N}$ に対して (Ω, P) 上で定義された $\{0,1\}$-値確率過程[3] $\{X_n^{(m)}\}_{n=0}^{\infty}$ を次のように定める。

1.Definition

$$X_n^{(m)}(\omega) := \sum_{i=1}^{m} d_i(\{\omega + n\alpha\}) \pmod 2, \quad \omega \in \Omega, \qquad n = 0,1,\ldots \tag{1}$$

ここに α は無理数であり、また記号 $\{x\}$ は実数 x の小数部分を表す。

このとき次の主定理が成り立つ。

2.Theorem ([16]) ほとんどすべての無理数 α に対して[4]、確率過程 $\{X_n^{(m)}\}_{n=0}^{\infty}$ は $m \to \infty$ のとき硬貨投げの確率過程に有限次元分布の意味で収束する。すなわち、任意の $k \in \mathbf{N}$ と任意の $\varepsilon_0, \ldots, \varepsilon_{k-1} \in \{0,1\}$ に対して次式が成り立つ。

$$\lim_{m\to\infty} P\left(X_0^{(m)}(\omega) = \varepsilon_0, \ldots, X_{k-1}^{(m)}(\omega) = \varepsilon_{k-1}\right) = 2^{-k}$$

標題の「無理数回転による擬似乱数」は m が十分大きいときの確率過程 $\{X_n^{(m)}\}_{n=0}^{\infty}$ のサンプルパス (特定の $\omega_0 \in \Omega$ (初期値) を選んだときに実現される数列 $\{X_n^{(m)}(\omega_0)\}_{n=0}^{\infty}$) のことである。これに関しては次の定理が成り立つ。

3.Theorem ほとんどすべての無理数 α、任意の $\omega_0 \in \Omega$、任意の $k \in \mathbf{N}$ および 任意の $\varepsilon_0, \ldots, \varepsilon_{k-1} \in \{0,1\}$ に対して次式が成り立つ[5]。

$$\lim_{m\to\infty} \lim_{N\to\infty} \frac{1}{N} \#\{\, 0 \le n \le N-1 \,|\, X_{nk}^{(m)}(\omega_0) = \varepsilon_0, \ldots, X_{nk+k-1}^{(m)}(\omega_0) = \varepsilon_{k-1} \,\} = 2^{-k}$$

---

[3] 詳しく言うと第2節で述べるとおり強定常性を持つ確率過程である。

[4] Lebesgue 測度に関して。事実としてはすべての無理数で成り立つと思われるが、厳密な証明のためには現在のところ技術的な条件が α に必要。なお、3次元以下の周辺分布はすべての無理数に対して硬貨投げの確率過程のそれに収束する。

[5] $\#\{\cdots\}$ は集合 $\{\cdots\}$ の要素の個数を表す。

3.Theorem の意味することは、m が大きいとき擬似乱数 $\{X_n^{(m)}(\omega_0)\}$ は多次元にわたってほぼ均等に分布すると言うことである。3.Theorem は 2.Theorem と無理数回転 (変換 $\omega \mapsto \{\omega+\alpha\}$) の**一意エルゴード性**と呼ばれる次の性質によって導かれる。

4.Lemma　任意の無理数 α、任意の初期値 $\omega_0 \in [0,1)$ および任意の Riemann 可積分関数 $g:[0,1) \to \mathbf{R}$ に対して次式が成り立つ。

$$\lim_{N\to\infty} \frac{1}{N} \sum_{n=0}^{N-1} g(\{\omega_0 + n\alpha\}) = \int_0^1 g(\omega) d\omega$$

最後に、2.Theorem における収束は距離の概念によって述べることができることに注意しておく。このことは我々の擬似乱数の近似の程度が定量的に述べられることを示す。

まず硬貨投げの確率過程は $\{0,1\}^\infty$ に値を取る確率変数に外ならない。その分布は

$$\mu := \prod_{n=0}^{\infty} \frac{\delta_0 + \delta_1}{2}, \qquad \delta_i \text{ は } i=0,1 \text{ における Dirac 測度}$$

である。同様に 1.Definition で定義された確率過程も $\{0,1\}^\infty$ に値を取る確率変数でありその分布を $\mu^{(m)}$ と書く。すなわち、

$$\mu^{(m)}(B) := P\left(\{X_n^{(m)}(\omega)\}_{n=0}^{\infty} \in B\right), \qquad B \subset \{0,1\}^\infty$$

ただし、$\{0,1\}^\infty$ には通常の直積位相を入れて B はその位相でもって Borel 集合とする。このとき、2.Theorem から $\mu^{(m)}$ が μ に弱収束すること、すなわち、任意の連続関数 $F:\{0,1\}^\infty \to \mathbf{R}$ に対して

$$\lim_{m\to\infty} \int_{\{0,1\}^\infty} F\, d\mu^{(m)} = \int_{\{0,1\}^\infty} F\, d\mu \tag{2}$$

となることがわかる。さて、$\{0,1\}^\infty$ は可分コンパクト距離空間であるから、その上の連続関数全体の集合は一様位相に対して可分、すなわち可算個の稠密な部分集合 $\{F_n\}_{n=0}^\infty$ を持つ。それで $\{0,1\}^\infty$ 上の確率測度全体の集合 $\mathcal{M}_1(\{0,1\}^\infty)$ に Prohorov の距離 ([1]) と呼ばれるものを

$$d(\xi,\eta) := \sum_{n=0}^{\infty} 2^{-n} \min\left(1, \left|\int_{\{0,1\}^\infty} F_n\, d\xi - \int_{\{0,1\}^\infty} F_n\, d\eta\right|\right), \qquad \xi,\eta \in \mathcal{M}_1(\{0,1\}^\infty)$$

と定める。そこで 1.Definition の確率過程と硬貨投げの確率過程の距離 (あるいは対応する擬似乱数 $\{X_n^{(m)}(\omega_0)\}_{n=0}^\infty$ の誤差) を $d(\mu^{(m)},\mu)$ で与えればよい。(2) から、すぐわかるように

$$\lim_{m\to\infty} d(\mu^{(m)},\mu) = 0$$

が成り立つ。

2. 力学系による擬似乱数生成の一般論

この節では、一般に擬似乱数を計算機によって生成しようとするときに用いられる数学的枠組みについて考察し、前節で定義した擬似乱数生成法がどのような新しい意図を持って考え出されたものかを明かにする。

擬似乱数を計算機で生成しようとするとき、数学的には**力学系**と呼ばれる枠組み (集合 Ω とその上の変換 $T:\Omega\to\Omega$ の組 (Ω,T) のこと) を利用するのが一般的である。$\{0,1\}$-値擬似乱数の場合は、力学系 (Ω,T)、写像 $f:\Omega\to\{0,1\}$ および初期値 $\omega_0\in\Omega$ を適当に設定して、

$$X_n(\omega) := f(T^n\omega_0), \qquad n=0,1,\dots \tag{3}$$

と定義される数列 $\{X_n(\omega_0)\}_{n=0}^{\infty}$ として得られる。たとえば、線形合同法 ([15] に詳しい) と呼ばれる擬似乱数生成法では、$\Omega=\{0,1,\dots,M-1\}$ 上の写像 T を

$$T\omega = a\omega+b \pmod M$$

と定義する。これから、$\{0,1\}$-値乱数を得るには関数 f を次式で定めて (3) を使う。

$$f(\omega) := \begin{cases} 0, & \text{if } \omega < M/2 \\ 1, & \text{if } \omega \geq M/2 \end{cases}$$

もちろん、乱数生成のためには必ず力学系を利用しなければならないと言うことはない。しかし、実用的なプログラムを書くためには力学系を利用するのが最も容易であり、力学系を利用した乱数生成法の可能性と限界を明かにしておくことは重要と思われる。そこで問題を、「力学系 (Ω,T)、写像 f および初期値 $\omega_0\in\Omega$ をどのように設定すれば、(3) で定義される数列 $\{X_n(\omega_0)\}_{n=0}^{\infty}$ が良い擬似乱数になるか」と言うふうに設定してみよう。

我々は確率論で扱い易くするために力学系 (Ω,T) および写像 f に以下のようないくつかの付加的な条件を要請する。それぞれの条件の意味については後で解説する。

1. Ω は無限集合である。
2. 力学系 (Ω,T) に対して $P\circ T^{-1}=P$ を満たす Ω 上の確率測度 P が存在する。
3. 力学系 (Ω,T) は**エルゴード的**である: 任意の $g\in L^1(\Omega,P)$ に対して

$$\lim_{N\to\infty}\frac{1}{N}\sum_{n=0}^{N-1} g(T^n\omega_0) = \int_\Omega g(\omega)P(d\omega), \qquad P\text{-a.e.}\omega_0\in\Omega \tag{4}$$

4. 関数 $f:[0,1)\to\{0,1\}$ は $P(f=0)=P(f=1)=1/2$ を満たす。

条件 1. については現実的でないと思われるかもしれない。確かに、現実には計算機のメモリーには限りがあるから、Ω は有限集合とならざるを得ない。従って生成される擬似乱数は周期的になる。既存の擬似乱数生成法は、たとえば Tausworthe 法 [19] のように代数的理論によって非常に長い周期を持つ数列を作りだし、それを擬似乱数として用いることを提案している。しかし、このように有限の状態空間の上の力学系は代数的に少しは解析できても、確率論的にはあまり興味ある対象ではない。それで、非周期的な擬似乱数を生み出せるような力学系を設定するために少なくとも理論上は条件 1. を要請する。たとえ、プログラムで実現するときには、Ω を有限集合で近似するとしても十分実用的効果がある。

条件 2. の確率測度 P は**不変確率測度**と呼ばれ、力学系を確率論によって解析しようとするときいつも存在を仮定する。ここでも擬似乱数 $\{X_n(\omega_0)\}_{n=0}^{\infty}$ の統計的性質を調べるために必要である。いま、初期値 ω_0 を Ω の中から確率 P に従ってランダムに選ぶと言う状況を考える。ω_0 のままだと特殊な初期値を連想するので、これをランダムに選ぶときは ω と書こう。このとき、数列 $\{X_n(\omega)\}_{n=0}^{\infty}$ は確率空間 (Ω, P) 上の確率過程と見なされる。さらに、この確率過程は条件 2. により、**強定常性**と呼ばれる次の性質を持つことがわかる: 任意の $k, n \in \mathbf{N}$ に対して、二つの k 次元確率変数 $(X_0(\omega), \ldots, X_{k-1}(\omega))$ と $(X_n(\omega), \ldots, X_{k-1+n}(\omega))$ は同じ分布に従う。

条件 3. はその強定常確率過程の多次元周辺分布の性質がサンプルパスの相対度数分布に遺伝するために必要な仮定である。

条件 4. は生成される擬似乱数は少なくとも 1 次元分布が均等であることを要請している。

以上の仮定の下で硬貨投げの確率過程を数学的に完全に実現できることが次の例によって明らかにされている (たとえば [2] の最初の十数ページを見よ)。

5.**Example** (Borel の例) $\Omega = [0,1)$、$P =$ Lebesgue 測度、さらに変換 T を $T\omega := \{2\omega\}, \omega \in \Omega$ とする。このとき、関数 f を

$$f(\omega) := d_1(\omega) = \begin{cases} 0, & \omega < 1/2 \\ 1, & \omega \geq 1/2 \end{cases}$$

とすれば、確率空間 (Ω, P) 上の確率過程 $\{X_n(\omega)\}_{n=0}^{\infty}$

$$X_n(\omega) = f(T^n\omega), \quad \omega \in \Omega, \qquad n = 0, 1, 2, \ldots$$

は硬貨投げの確率過程の一つの実現である。

一般に、初期値の選択だけがランダムに行われるような力学系による確率過程で硬貨投げの確率過程を完全に実現するものは本質的に Borel の例と同等である。

しかし、このような確率過程の実現を乱数生成に利用することはできない。確かに P-a.e. の初期値 $\omega_0 \in \Omega$ に対して数列 $\{X_n(\omega_0)\}_{n=0}^{\infty}$ は完全な乱数であるはずだが、残念ながら具体的にどの初期値に対してそうなるかを原理的に知ることができないのである。

このような初期値の選択問題を避けるためには、力学系にエルゴード性よりも強い条件である一意エルゴード性 (無理数回転の場合は前節の 4.Lemma の性質。一般の場合は [21] を見よ。) を仮定すればよい。しかし、その場合は決して真の乱数は得られない。ここに、力学系による乱数生成の限界がある。

前節の 2.Theorem や [7] では一意エルゴード的な力学系でも硬貨投げの確率過程に任意に近い強定常確率過程が得られることを示している。これらは力学系による乱数生成の最大の理論的可能性を示していると言ってもよいかもしれない。

さらに実用面を考えれば、理論を実現する計算機プログラムが十分高速に擬似乱数を生成できる必要があるが、前節の 2.Theorem に基づく擬似乱数生成プログラムは後の節で示すように実用的レベルに達している。

3. 多次元周辺分布の計算アルゴリズム

1.Definition で与えた強定常確率過程 $\{X_n^{(m)}(\omega)\}_{n=0}^{\infty}$ の多次元周辺分布

$$P\left(X_0^{(m)}=\varepsilon_0,\ldots,X_{k-1}^{(m)}=\varepsilon_{k-1}\right),\qquad k\in\mathbf{N},\qquad \varepsilon_0,\ldots,\varepsilon_{k-1}\in\{0,1\} \tag{5}$$

について考えよう。無理数回転の一意エルゴード性によって、これは擬似乱数 $\{X_n^{(m)}(\omega_0)\}$ における連 $(\varepsilon_0,\ldots,\varepsilon_{k-1})$ の出現する漸近的相対度数に外ならないことがわかる。

一般に擬似乱数の善し悪しは統計的検定によって行われる。しかし、現実に行ない得る検定はきわめて限られていて十分な評価を得ることは困難である。従って事前に分布について何らかの知識があると都合がよい。我々の擬似乱数の場合は (5) をすべて算出するためのアルゴリズムが存在する。

6.**Lemma** ([16]) (i) 多次元周辺分布 (5) は次の量から算出することができる。

$$E^{(m)}_{0,k_1,\ldots,k_{l-1};\mathrm{odd}}:=P\left(X_0^{(m)}+X_{k_1}^{(m)}+\ldots+X_{k_{l-1}}^{(m)}=\text{奇数}\right),$$

$$0<k_1<\ldots<k_{l-1},\qquad l\in\mathbf{N}$$

(ii) $l\in\mathbf{N}$ が奇数ならば $E^{(m)}_{0,k_1,\ldots,k_{l-1};\mathrm{odd}}=1/2$ である。

6.Lemma から l が偶数のときに $E^{(m)}_{0,k_1,\ldots,k_{l-1};\mathrm{odd}}$ を求めるアルゴリズムがあればよい。そのために、いくつかの記号を導入しなければならない。無理数回転で用いられる無理数 α に対して、

$$\alpha_j:=\{k_j\alpha\},\quad j=1,\ldots,l-1,\qquad (l\text{は偶数})$$

とおき、

$$\begin{cases}\alpha_j^{(m)L} &:= [2^m\alpha_j]/2^m\\ \alpha_j^{(m)U} &:= \{[2^m\alpha_j+1]/2^m\}\\ \beta_j^{(m)} &:= 2^m(\alpha_j-\alpha_j^{(m)L})\end{cases}$$

とする。ここに、記号 $[\cdot]$ は整数部分を表す。次に $\{1,\ldots,l-1\}$ 上の 置換 $\sigma(m,\cdot)$ を次のように定める。

$$1>\beta^{(m)}_{\sigma(m,1)}\geq\beta^{(m)}_{\sigma(m,2)}\geq\cdots\geq\beta^{(m)}_{\sigma(m,l-1)}\geq 0$$

ただし、便宜上 $\beta^{(m)}_{\sigma(m,0)}:=1,\beta^{(m)}_{\sigma(m,l)}:=0$ と約束しておく。そして

$$\alpha_j^{(m),s}:=\begin{cases}\alpha_j^{(m)U}, & \text{if } \sigma(m,j)\leq s\\ \alpha_j^{(m)L}, & \text{if } \sigma(m,j)>s\end{cases}$$

とした上で

$$\boldsymbol{\alpha}^{(m),s}:=(\alpha_1^{(m),s},\ldots,\alpha_{l-1}^{(m),s}),\quad s=0,1,\ldots,l-1$$

とおく。最後に、2 進有限小数の集合を以下のように定義する。

$$D:=\bigcup_{m\in\mathbf{N}}\left\{\frac{n}{2^m}\in[0,1)\,\middle|\,n=0,\ldots,2^m-1\right\}$$

7.Theorem ([16])

$$E^{(m)}_{0,k_1,\ldots,k_{l-1};\ \mathrm{odd}} = \sum_{s=0}^{l-1} \left(\beta^{(m)}_{\sigma(m,s)} - \beta^{(m)}_{\sigma(m,s+1)}\right) B(\boldsymbol{\alpha}^{(m),s})$$

ここに $B(\cdot)$ は $D^{l-1} = \overbrace{D \times \cdots \times D}^{l-1}$ の上で定義されたある実数値関数で、$B(\boldsymbol{\alpha}^{(m),s})$ の値は

$$B(\boldsymbol{\alpha}^{(0),s}) = 0, \qquad s = 0, 1, \ldots, l-1$$

および次の漸化式で計算される。

$$B(\boldsymbol{\alpha}^{(m),s}) = \begin{cases} \dfrac{1}{2}B(\boldsymbol{\alpha}^{(m-1),s_2}) + \dfrac{1}{2}B(\boldsymbol{\alpha}^{(m-1),s_1+s_2}), & \text{if } s_1 \text{ is even,} \\ \dfrac{1}{2}\left(1 - B(\boldsymbol{\alpha}^{(m-1),s_2})\right) + \dfrac{1}{2}\left(1 - B(\boldsymbol{\alpha}^{(m-1),s_1+s_2})\right), & \text{if } s_1 \text{ is odd.} \end{cases}$$

ただし、s_1, s_2 は次で与えられる。

$$s_1 := \sum_{j=1}^{l-1} d_m(\alpha_j^{(m),s}), \qquad s_2 := \sum_{j=1}^{s} d_m(\alpha_{\sigma(m,j)})$$

4. 多次元周辺分布の事前評価の例

我々の擬似乱数の場合は 7.Theorem を用いると統計的検定を待つまでもなく、多次元周辺分布に関する統計的性質を調べることができる。以下に挙げる例では、無理数回転に用いる無理数として黄金分割の比として知られる次の数を採用した[6]。

$$\alpha = \frac{\sqrt{5}-1}{2}$$

はじめに、我々の乱数の 2 項間の相関について

$$a^{(m)}(K) := \max_{1 \le k \le K} \left| E^{(m)}_{0,k;\ \mathrm{odd}} - \frac{1}{2} \right| \qquad N_c^{(m)}(K) := \frac{1}{16\left(a^{(m)}(K)\right)^2} \tag{6}$$

と定義する。$N_c^{(m)}(K)$ を臨界サンプル数[7]と呼ぶ。次の各々の帰無仮説

$$E^{(m)}_{0,k;\ \mathrm{odd}} = \frac{1}{2}, \qquad k = 1, 2, \ldots, K,$$

に関して検定 (危険率 5%) を行うとき、サンプル数が $N_c^{(m)}(K)$ 以下ならば、上の各々の仮説はそれぞれ 93% 以上の確率で採択されることが期待できる ([16])。

(6) を $K = 10000$ のときに計算し、表にしたのが 8.Table[8] である。$a^{(m)}(K)$ の値のすぐ右側の ( ) の中は最大値がどのような k によって達成されたかを表わす。

---

[6] この数が 2.Theorem および 3.Theorem の主張を成り立たせる無理数かどうか、筆者は厳密な回答ができない。しかし、実用上は 2 進小数展開で 100 桁程度あればよく、厳密な議論は大して問題にならない。

[7] [16] で述べられている critical sample number はここでの $N_c^{(m)}(K)$ の 4 倍である。

[8] 8.Table と 9.Table において $a^{(m)}(10000)$ と $b^{(m)}(16)$ が $m \to \infty$ のときほぼ指数的に減少することが読み取れる。実際、理論的にもほとんどすべての無理数 α についてそのことが示せる ([16])。

8.Table

m	$a^{(m)}(10000)$	(k)	$N_c^{(m)}(10000)$
10	0.4860680	(5473)	2.6×10^{-1}
20	0.1084934	(1449)	5.3×10^{0}
30	0.0435756	(305)	3.3×10^{1}
40	0.0029834	(305)	7.0×10^{3}
50	0.0001943	(610)	1.7×10^{6}
60	0.0000136	(8484)	3.4×10^{8}
70	1.2×10^{-6}	(7264)	4.1×10^{10}
80	2.0×10^{-7}	(7697)	1.6×10^{12}
90	8.5×10^{-9}	(165)	8.7×10^{14}
100	2.9×10^{-9}	(5201)	7.7×10^{15}

9.Table

m	$b^{(m)}(16)$	$k_1,\ldots$
10	0.1099945	1,9,10
20	0.0053298	9
30	0.0008288	9
40	0.0000769	9
50	8.0×10^{-6}	9
60	6.1×10^{-7}	9
70	5.9×10^{-8}	1
80	6.8×10^{-9}	16
90	2.1×10^{-9}	16
100	3.0×10^{-10}	1

次に一般の多次元周辺分布 (K-次元以下) の評価を考えよう。このとき各偶数 l について

$$E^{(m)}_{0,k_1,\ldots,k_{l-1};\ \mathrm{odd}}, \qquad 1\le k_1<\cdots<k_{l-1}\le K$$

を評価すればよい。これらを全部調べることは比較的小さな K についてさえ計算量が莫大になり大きな K では絶望的に思えるが、それでも少しは望みがある。9.Table は $K=16$ の場合を計算したものである。左の欄は、

$$b^{(m)}(16):=\max_{1\le k_1<\ldots<k_{l-1}\le 16}\left|E^{(m)}_{0,k_1,k_2,\ldots,k_{l-1};\ \mathrm{odd}}-\frac{1}{2}\right|,$$

を表わし、右の欄は最大値がどのような $k_1,\ldots$ によって達成されたかを表わす。9.Table からは次の仮説が成り立つように見受けられる。

10.Hypothesis 各 $K\in\mathbf{N}$ に対して m が十分大きいとき、

$$\max_{1\le k_1<\cdots<k_{l-1}\le K}\left|E^{(m)}_{0,k_1,\ldots,k_{l-1};\ \mathrm{odd}}-\frac{1}{2}\right|=\max_{1\le k\le K}\left|E^{(m)}_{0,k;\ \mathrm{odd}}-\frac{1}{2}\right|$$

10.Hypothesis が正しければ、我々は 2 項間の相関の最大値さえ評価すればよいことになる[9]。

5. プログラム例

ここに、$\alpha=(\sqrt{5}-1)/2$ および $m=90$ の場合に、我々の擬似乱数を生成するための C によるプログラムの例を挙げる。前節の 8.Table で見るように擬似乱数の精度としては実用上十分であると期待される。

[9] 10.Hypothesis が正しければ 2.Theorem および 3.Theorem がすべての無理数に対して成り立つこともわかる。

```
/*============================================================*/
/*    Implementation of Pseudo-random number generator by     */
/*    m90-method with the irrational number (sqrt(5)-1)/2.    */
/*============================================================*/
#include <stdio.h>

#define  LIMIT   0x3fffffff
#define  CARRY   0x40000000

unsigned long  omega[5];                    /* Current seeds */

void  m90SetSeeds(s0, s1, s2, s3, s4)       /* Initialization */
unsigned long  s0,s1,s2,s3,s4;
{
    omega[0] = s0 & LIMIT; omega[1] = s1 & LIMIT; omega[2] = s2 & LIMIT;
    omega[3] = s3 & LIMIT; omega[4] = s4 & LIMIT;
}

char  m90RandomBit()                        /* Returns 0 or 1 at random */
{
    static unsigned long alpha[5] = {    /* Irrational number (sqrt(5)-1)/2 */
        0x278dde6e, 0x17f4a7c1, 0x17ce7301, 0x205cedc8, 0x0d042089
    };
    char data_byte;
    union bitarray {
        unsigned long of_32bits;
        char of_8bits[4];
    } data_bitarray;
    int j;

    for (j=4; j>=1; ){
        omega[j] += alpha[j];
        if ( omega[j] & CARRY ){
            omega[j] &= LIMIT;
            omega[--j]++;
        }
        else --j;
    }
    omega[0] += alpha[0]; omega[0] &= LIMIT;
    data_bitarray.of_32bits = omega[0] ^ omega[1] ^ omega[2];
    data_byte =  data_bitarray.of_8bits[0] ^ data_bitarray.of_8bits[1]
              ^ data_bitarray.of_8bits[2] ^ data_bitarray.of_8bits[3];
    data_byte = ( data_byte >> 4 ) ^ data_byte;
    data_byte = ( data_byte >> 2 ) ^ data_byte;
    return( 1 & (( data_byte >> 1 ) ^ data_byte));
}

void  main()
{
    int j;
    m90SetSeeds(0,0,0,0,0);
    for (j=1; j<=50; j++)
        printf("%d",m90RandomBit());
    printf("\n");
}
```

無理数回転を正確に実行するために、このプログラムでは多倍長加算を行う。すなわち、我々が必要としているのは 90 bit だが ($m = 90$)、ここでの加算は 150 bit で行っている。このため、少なくとも 2^{50} 個以上の擬似乱数を丸め誤差の影響を受けずに正確に生成することができるであろう[10]。

このプログラムでは実行速度を上げるために、次のようなトリックを利用している: 関数 $f^{(m)}(\omega) := \sum_{i=1}^{m} d_i(\omega) \pmod 2$ の値を計算する部分で、排他的論理和の演算を用いている。たとえば、$\omega \in [0,1)$ の最初の 16 bit が

$$0100111001011011$$

であったとしよう。このとき、1 が 9 個あるから、$f^{(16)}(\omega) = 1$ である。次に、この bit の並びを真二つに分けて、それらの排他的論理和 (XOR) をとってみると、

$$01001110 \ \text{XOR} \ 01011011 = 00010101$$

となる。演算結果 ω' は 8 bit になるが、これは 1 を 3 個持っているから、$f^{(8)}(\omega') = 1$ である。一般にこの手続きによって 1 の個数の偶奇は変わらないことに注意せよ。上のプログラムではこうしたトリックを何回も用いて計算速度を上げている (演算 XOR はきわめて早く処理される)。もし、アセンブリ言語を使用できる場合はパリティフラグが有用であろう。

6. 参考文献

[1] P.Billingsley, *Convergence of Probability measures*, John Willey & Sons, (1968)

[2] P.Billingsley, *Probability and measure*, 2nd edition, John Willey & Sons, (1986)

[3] N.Bouleau and D.Lépingle, *Numerical methods for stochastic processes*, John Wiley & Sons, (1994)

[4] R.Burton and M.Denker, On the Central Limit Theorem for Dynamical Systems, *Trans. AMS.* **103-2**(1987), 715-726

[5] G.J.Chaitin, Algorithmic Information Theory, *IBM J. Res. Develop.* **21** (1977), 350-359

[6] J.N.Franklin, Deterministic simulations of random processes, *Math. Comp.* **17** (1965), 28-59

[7] K.Fukuyama, The central limit theorem for Rademacher system, *Proc. Japan Acad.* **70**, Ser. A, No.7 (1994), 243-246

[8] 伏見正則、乱数、(東京大学出版会)、(1989)

[9] 香田徹、"カオスの間接的時系列解析法とその応用"、システム制御情報学会誌、**37**, No.11 (1993), 661-668

[10] 前節の 8.Table によれば $N_\epsilon^{(90)}(10000) = 8.7 \times 10^{14} < 2^{50}$ なので擬似乱数を 2^{50} 個も生成すると統計的には誤差が大きくなる。もっとも、8.7×10^{14} と言う数は実用上十分大きく、1 秒間に 10^8 bit を使っても 8.7×10^{14} bit を使い切るには 3ヵ月以上かかる。

[10] T.Kohda and A.Tsuneda, Pseudonoise Sequences by Chaotic Nonlinear Maps and Their Correlation Properties, IEICE Trans., **E76-B**, No.8 (1993), 855-862

[11] L.Kuipers and H.Niederreiter, *Uniform distribution of sequences*, Interscience, (1974)

[12] Knuth D.E., *The Art of Computer Programming*, 2nd ed., Addison-Wesley, (1981), (邦訳) 準数値算法/乱数 (渋谷政昭訳)、サイエンス社、(1983)

[13] Martin-Löf, The definition of random sequences, *Inform. Control* **7** (1966), 602-619

[14] S.Ogawa, Pseudorandom Functions Whose Asymptotic Distributions Are Asymptotically Gaussian, *Math. Anal. and Appl.*, **158**, No.1, (1991)

[15] S.K.Park and K.W.Miller (訳:西村恕彦)、"乱数生成系で良質なものはほとんどない"、**bit** (共立出版) 4 月号、5 月号、(1993)

[16] H.Sugita, Pseudo-random number generator by means of irrational rotation, preprint, (1994)

[17] 数理解析研究所講究録 498、乱数プログラム・パッケージ、(1983)

[18] 数理解析研究所講究録 850、確率数値解析における諸問題、(1993)

[19] R.C.Tausworthe, Random numbers generated by linear recurrence modulo two, *Math. Comp.* **19**, (1965), 201-209

[20] 津田孝夫、モンテカルロ法とシミュレーション、培風館、改定版 (1977)

[21] P.Walter, *An Introduction to Ergodic Theory*, Springer, (1981)

杉田 洋
九州大学大学院数理学研究科
810 福岡市中央区六本松 4–2–1
E-mail: sugita@math.kyushu-u.ac.jp