数理解析研究所講究録
1011 巻 1997 年 77-88

無理数回転を利用した擬似乱数生成法

— 確率論による数値解析的アプローチ

九州大学大学院数理学研究科　　杉　田　　洋 (Hiroshi SUGITA)

Abstract. Lebesgue確率空間 (Ω, P), $\Omega = [0,1)$, P =Lebesgue測度, の上で定義された $\{0,1\}$-値定常過程

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

(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 $m \to \infty$ のとき硬貨投げの確率過程 (平均 1/2 の $\{0,1\}$-値独立同分布確率変数列) に分布収束する。我々は m を十分大きく取ったとき, この定常過程のサンプルパスを擬似乱数として用いることを提唱する。実用には $\alpha = (\sqrt{5}-1)/2$ のとき $m = 90$ 程度で十分である。

1　序

「擬似乱数生成法」というと, 数学的に純粋でない「まがい物」, という印象を持つ人が少なくない。「乱数」をどのように定義しようとも (cf.[14, 5, 13]),「計算機プログラムによる乱数生成は実現不可能である」ことには間違いなく, 従って実際に計算機プログラムによって生成される擬似乱数は確かに「乱数の近似物」にすぎない。

一方で「近似物」が立派に通用している例もある。たとえば, 無理数や無限級数なども, 有限の記憶領域しか持たない計算機では実現不可能であって, 実際は無理数を有理数で近似して, あるいは無限級数を有限和で近似して計算する。こうした近似を「まがい物」と思う人はほとんどいない。数値解析では「近似」と言うとき, 次の二つの条件を要請する:

(A.1) 誤差について定量的に述べることができること。

(A.2) 誤差をいくらでも小さくするアルゴリズムが存在すること。

これらの要請が満たされれば, 実際的な計算において「真の値」と「近似値」の間に機能的な差はなくなる。無理数の近似有理数あるいは無限級数の近似有限和はまさにこれらの条件を満たしている。一方, 現在主流の各種の擬似乱数は残念ながら数値解析における上の二つの要請を満たしているとは言い難い。

本稿では, 擬似乱数を数値解析における近似のレベルまで引き上げる一つの実用的な手法について述べる。すなわち, 我々は擬似乱数を力学系によって定まる定常過程として位置付けし, その誤差を確率論の言葉でもって表し, さらにその誤差が 0 に収束するような擬似乱数生成アルゴリズムの「列」を構成する。

本稿の構成を明らかにしておく。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。第3節では我々の擬似乱数の定義と主定理を述べる。第3節では我々の擬似乱数の多次元分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。

2 力学系による擬似乱数生成

2.1 硬貨投げの確率過程

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 $\{X_n\}_{n=0}^{\infty}$，すなわち硬貨投げの確率過程，を考えれば十分である。硬貨投げの確率過程はいろいろな性質を持つが，ここでは次の三つの性質に注目する。

(B.1) $\{0,1\}$-値強定過程である。すなわち，任意の $k,l \in \mathbf{N}$ に対して $\mathbf{R}^k$-値確率変数 $(X_0,\ldots,X_{k-1})$ と $(X_l,\ldots,X_{l+k-1})$ が同分布である。

(B.2) エルゴード的である。とくに，確率 1 で，任意の $k \in \mathbf{N}$ と任意の $\varepsilon \in \{0,1\}^k$ に対して[1]，

$$\text{Probab.}\,((X_0,\ldots,X_{k-1})=\varepsilon) = \lim_{N\to\infty}\frac{1}{N}\#\{0\le n\le N-1 \,|\, (X_{n(k-1)},\ldots,X_{nk-1})=\varepsilon\}$$

(B.3) 確率 1 で見本は 2-進正規列である。すなわち，確率 1 で，任意の $k \in \mathbf{N}$ と任意の $\varepsilon \in \{0,1\}^k$ に対して

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

となる。

また，以上三つの性質を持つ確率過程は硬貨投げの確率過程に限られる。以下，本稿で扱う擬似乱数は硬貨投げの確率過程をモデルとするものとする。

2.2 定常過程としての擬似乱数

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(擬似乱数の「種」と言うことも多い) $\omega\in\Omega$ を適当に設定して，

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

と定義される数列 $\{X_n(\omega)\}_{n=0}^{\infty}$ として得られる。

そこで問題は「力学系 (Ω,T)，写像 f および初期値 $\omega\in\Omega$ をどのように設定すれば，(1)で定義される数列 $\{X_n(\omega)\}_{n=0}^{\infty}$ が良い擬似乱数になるか」ということである。

さて，初期値 $\omega\in\Omega$ はユーザが自由に指定できるようにしておく。これは一つの擬似乱数生成アルゴリズムによって様々な擬似乱数系列を生成できるようにするためである。確率論的には初期値 ω をある Ω 上の確率測度 P に従って選択するという状況を想定するのが自然だろう。そうすれば数列 $\{X_n(\omega)\}_{n=0}^{\infty}$ は確率空間 (Ω,P) 上の確率過程と見なされ，乱数のモデル「硬貨投げの確率過程」と同じ土俵に乗ることができる。

確率測度 P は原理的にはどのようなものを仮定してもよい。しかし，様々な計算をうまく行うために T-不変であると仮定するのがよい。このとき，$\{X_n(\omega)\}_{n=0}^{\infty}$ は強定常過程となって条件(B.1)を満たす。

また，擬似乱数のサンプルからその分布を推定できるという実際的な仮定は是非とも必要だから，エルゴード性(B.2)も仮定しよう。

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

2.3 一意エルゴード性

以上の仮定の下で(B.3)を満たすようにできる，すなわち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実現できる。

1.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[2]。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(たとえば[2]の最初の十数ページを見よ)。

もっとも，Borelの例は乱数生成には使えない。計算機では初期値 ω を 2-進有限小数に設定せざるを得ないが，それだとすべての軌道がたちどころに 0 に収束してしまう。この難点を乗り越えるために，香田氏らは2次以上の多項式を用いたカオス的力学系による擬似乱数生成法を考案している([9, 11][3])。しかしながら，カオス的力学系では長い軌道の追跡が数値計算上不可能であるため，性質(B.2)が数値的に成り立つことを保証するのは小さい k の場合を除いて絶望的である。たとえ長い軌道の追跡が可能になったとしても，カオス的力学系では性質(B.2)を満たさない初期値 ω が必ず存在することにも注意しなければならない。

そのため，ここではカオス的力学系を擬似乱数生成には採用しない。そのかわり，任意の初期値 ω に対して性質(B.2)を要請する。そのためには一意エルゴード性:

(B.4) 任意の有界連続関数 $g : \Omega \to \mathbf{R}$ に対して

$$\lim_{N\to\infty} \frac{1}{N} \sum_{n=0}^{N-1} g(T^n\omega) = \int_{\Omega} g(\omega')P(d\omega'), \qquad \forall \omega \in \Omega \tag{2}$$

を仮定しよう。このとき，初期値 ω の選び方によって擬似乱数の統計的性質は左右されないので大変都合がよい。しかし，この場合は理論的にも決して硬貨投げの確率過程は得られないことに注意しよう。

2.4 擬似乱数の誤差と統計的検定

我々の定式化では擬似乱数は定常過程であるから，硬貨投げの確率過程からの隔たり，すなわち擬似乱数の誤差，を確率論の概念によって述べることができる。それは「擬似乱数の分布と硬貨投げの確率過程の分布の距離」と定義すればよい。一般に位相空間 S 上の確率測度全体 $\mathcal{M}_1(S)$ には標準的な位相として「弱収束の位相」と呼ばれる位相が知られている。それは，すべての連続関数 $f : S \to \mathbf{R}$ に対して，写像

$$\mathcal{M}_1(S) \ni \mu \mapsto \int_S f(s)\mu(s) \in \mathbf{R}$$

[2]実数 x に対して $\{x\}$ は x の小数部分を表す。
[3]本講究録の香田氏の記事も参照せよ。

が連続になるような最弱の位相である。とくに，可分完備距離空間上の「弱収束の位相」は距離付け可能であり，その位相を与える距離を一般にProhorovの距離(詳しくは[1]を見よ)という。そこで擬似乱数の誤差は $\mathcal{M}_1(\{0,1\}^\infty)$ 上のProhorovの距離で計る。

Prohorovの距離をもって擬似乱数の誤差と定義する理由は，擬似乱数の統計的検定と関係がある。実際，擬似乱数の善し悪しを見極める各種の統計的検定は，擬似乱数の分布が硬貨投げの確率過程とどれくらい離れているかを確かめる作業であると言ってよい。

一般に統計的検定とはどういうものかを述べよう。まず，関数[4] $F:\{0,1\}^\infty \to \mathbf{R}$ に硬貨投げの確率過程 $\mathbf{X}=\{X_n\}_{n=0}^\infty$ を代入して得られる確率変数 $F(\mathbf{X})$ については，その分布が良く分かっているとする。そこで $\{0,1\}$-値擬似乱数の一つのサンプル $\mathbf{x}=\{x_n\}_{n=0}^\infty$ を同様に代入して得られた値 $F(\mathbf{x})$ が，$\mathbf{x}$ を $\mathbf{X}$ のサンプルと仮定したときにおよそ起こり得ない値ならば，$\mathbf{x}$ は擬似乱数としてふさわしくないと判断し，また，十分起こり得る値ならば $\mathbf{x}$ を擬似乱数として採択する。このような作業が擬似乱数の検定である。

我々の定式化の下では，$\{0,1\}$-値強定常過程である擬似乱数 $\mathbf{X}'=\{X'_n\}_{n=0}^\infty$ を同様に代入し $F(\mathbf{X}')$ の分布が $F(\mathbf{X})$ の分布に近ければ近いほど，擬似乱数のサンプルが上記の意味で採択される確率が高くなる。すなわち，Prohorovの距離で測った誤差が小さい擬似乱数は各種統計的検定に合格する確率が高くなる。

実際には $F(\mathbf{X}')$ の値は何らかのサンプル平均であることが多いので，$\mathbf{X}'$ の分布が各サンプルの漸近的相対度数に遺伝していること，すなわち，一意エルゴード性(B.4)が成り立つと都合がよい。

2.5 問題設定 — 確率論による数値解析的アプローチ

以上の概念を準備すれば擬似乱数生成法の問題設定を述べることができる。序で述べたように擬似乱数を数値解析的レベルまで引き上げるためには(A.1)(A.2)の条件を満たすようにしなければならないが，前節までで(A.1)はクリアした。(A.2)のために，次のように問題を設定しよう。

(Ω,P) を確率空間，T を Ω 上の P-不変な変換で一意エルゴード的であるとする。そこで問題は次の性質を持つ写像列 $f^{(m)}:\Omega\to\{0,1\}$ を構成することである:

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

で定まる強定常過程の列 $\{X_n^{(m)}\}_{n=0}^\infty$，$m\in\mathbf{N}$，が $m\to\infty$ のとき，硬貨投げの確率過程に分布収束する。

このような写像列 $f^{(m)}$ を構成できれば理論的には(A.2)をクリアできる。擬似乱数としては十分大きな m に対して(3)で定まるものを採用すればよい。しかし，実用的な観点からはそのような写像 $f^{(m)}$ が計算機によって高速に計算されることが望ましい[5]。

---

[4]現実的な検定では有限個のサンプルしか扱わないから F は $\{0,1\}^\infty$ の積位相に関して連続であるとしてよい。

[5]実際，[7]では我々と同様の枠組みの下でGauss型独立同分布確率変数列に分布収束するようなものを実現しているが収束の早さが遅い。そのため，擬似乱数生成に利用しようとすると膨大な計算量を必要とするので実用的でない。

3 無理数回転を利用した擬似乱数生成

3.1 定義と極限定理

それでは，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 $\{0,1\}$-値強定常確率過程 $\{X_n^{(m)}\}_{n=0}^{\infty}$ を次のように定める。

2.Definition

$$X_n^{(m)}(\omega) := \sum_{i=1}^{m} d_i(\{\omega + n\alpha\}) \pmod 2, \quad \omega \in \Omega, \qquad n = 0, 1, \ldots \tag{4}$$

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。

3.Theorem ([18]) ほとんどすべての無理数 α に対して[6]，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分布収束する。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。

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

無理数回転の一意エルゴード性(変換 $T : \omega \mapsto \{\omega + \alpha\}$ が(B.4)を満たすこと)によって，各サンプルパスについて次の定理が成り立つ。

4.Theorem ほとんどすべての無理数 α，任意の $\omega \in \Omega$，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。

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

4.Theoremは，m が大きいとき擬似乱数 $\{X_n^{(m)}(\omega)\}$ の各サンプルパスは多次元にわたってほぼ均等に分布するということを意味する。

[6]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。

3.2 多次元周辺分布の計算アルゴリズム

2.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。我々の擬似乱数では (5) をすべて算出するためのアルゴリズムが存在する。

5.**Lemma** ([18]) (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。

5.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。

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

6.**Theorem** ([18])

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

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。

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

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

我々の擬似乱数の場合は 6.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[7]。

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

4.1 2項間相関

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[8]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([18])。

(6)を $K=10000$ のときに計算し，表にしたのが 7.Table[9] 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。

---

[7]この数が 3.Theorem および 4.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。

[8][18]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。

[9]7.Table と 8.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([18])。

7.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}

8.Table

m	$b^{(m)}(16)$	$k_1,\dots$
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

4.2 多項間相関

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

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

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。8.Table は $K=16$ の場合を計算したものである。左の欄は，

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

を表わし，右の欄は最大値がどのような $k_1,\dots$ によって達成されたかを表わす。8.Table からは次の仮説が成り立つように見受けられる。

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

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

実は 9.Hypothesis は 3.Theorem の証明を詳しく見ると成り立つことが十分期待できるのであるが現在のところ厳密な証明はない。もし 9.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[10]。

5 プログラミング

5.1 C によるプログラム例

数値解析の常として，精度を高めるためには計算時間が長くなり，計算時間を短くするためには精度を落とさなければならない。実用に供するためには精度が高ければそれでよい，というわけには行かない。

[10] 9.Hypothesis が正しければ 3.Theorem および 4.Theorem がすべての無理数に対して成り立つこともわかる。

ここでは $\alpha = (\sqrt{5}-1)/2$ 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 7.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。これは後述のように生成速度の観点からも十分実用になる。

```
/*============================================================*/
/*    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[11]。

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。

5.2 補足

5.2.1 擬似乱数の生成速度について

m90randombit は1秒間にパソコン PC-9821Xa10 で約1,000,000個のランダムビットを生成する。この生成速度は従来の擬似乱数生成法より遅いと感じるユーザも少なくないだろう。しかし計算量の理論によって明らかにされたように (邦語の解説記事としてはたとえば [10])，より精度の高い乱数を生成するためにはより複雑なプログラムがどうしても必要になる。そのため，精度の高い乱数生成に時間がかかるのは理論上，止むを得ない。最近，乱数の研究者のあいだでは簡便なプログラムで乱数を生成させることの限界が訴えられるようになってきた (cf. [15])。そのため多少時間がかかっても高精度の乱数が望まれている。実際，乱数生成に時間が多少かかるという欠点はたとえば後述の並列計算によってハードウエア的に克服されるからである。

5.2.2 周期について

よく尋ねられる質問として「その擬似乱数の周期はどのくらいか」というのがある。無理数回転を利用した擬似乱数の場合，理論的には周期は明らかに存在しない (あるいは無限大)。もっとも，無理数回転も実際には近似した「有理数回転」で代用するから，もちろん，周期はあるわけで，たとえば関数 **m90randombit** では周期は 2^{150} である。

しかしながら，前節で述べたように，**m90randombit** は周期の存在しない擬似乱数を実用限界を超えるほど大量に生成するので機能的には「周期を持たない乱数」を生成していると言える。

そもそも，2^{100} を超える周期について議論することはまったく不毛である。なぜなら，それほど多くの乱数を使うことは現実にはありえないから。

---

[11] 前節の 7.Table によれば $N_c^{(90)}(10000) = 8.7 \times 10^{14} < 2^{50}$ 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。

5.2.3 有理数回転による擬似乱数生成

関数 `m90randombit` は厳密には「有理数回転による擬似乱数生成」である。ただ，余分な桁数で計算することにより機能的には無理数回転と同等というに過ぎない。一方で，有理数回転による擬似乱数生成自身も十分良い擬似乱数生成法なのである。具体的には `m90randombit` において加算を90bitで行う。そのとき，擬似乱数は周期的になるが 4.Theorem の関数 B を計算することによって多次元分布を知ることができて，それは 6.Table とほとんど変わらない。加算の負担が減る分だけ擬似乱数の生成速度が少し早くなる。

5.2.4 並列計算について

無理数回転を利用した擬似乱数生成法は並列計算に大変適している。いま，プロセッサが K-個あるとする。最初に擬似乱数の初期値 $\omega \in [0,1)$ を選び，次にどれか一つのプロセッサで

$$\omega_j := \{\omega + j\alpha\}, \qquad j = 0, \ldots, K-1$$

を計算する。そして第 j 番目のプロセッサには初期値 ω_j で無理数 $K\alpha$ の無理数回転を行い擬似乱数生成を生成させる。これで，全体では初期値 ω で無理数 α の無理数回転による擬似乱数を生成していることになる。注目すべきことに，初期値の設定後，計算は各プロセッサごとに完全に独立して行われるので大変能率がよい。

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] 小林孝次郎，コルモゴロフの複雑さとランダムネス，*bit* vol.**27** No.5, 共立出版，May (1995), 88-95.

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

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

[13] 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)

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

[15] H.Niederreiter, New developments in umiform pseudorandom number and vector generations, *Lecture Notes in Statistics* **106**, Springer (1995), 87-120.

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

[17] 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)

[18] H.Sugita, Pseudo-random number generator by means of irrational rotation, *Monte Carlo Methods and Applications*, VSP, **1-1**, 35-57(1995).

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

[20] 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)

[21] 杉田 洋, 無理数回転による擬似乱数生成, 数理解析研究所講究録 915, 数値計算アルゴリズムの現状と展望 II, (1995)

[22] H.Sugita and S.Takanobu, Limit theorem for symmetric statistics with respect to Weyl transformation: Disappearance of dependency, (1996), preprint

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

[24] 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)

[25] 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(工学部分室)
812-81 福岡市東区箱崎6–10–1
E-mail: sugita@math.kyushu-u.ac.jp