複雑な関数の数値積分とランダムサンプリング†

杉田　洋

1　はじめに

一般に，数値積分では被積分関数の滑らかさのクラスごとに適した方法が選ばれ，その各々の数値積分法の誤差限界は被積分関数の滑らかさを表す量とサンプル点の数によって評価される．決定論的なサンプリングによる数値積分法として最も適用範囲が広いものは，準モンテカルロ法として知られるもので，有界変動関数に対して適用される (cf. 後述の式 (3))．しかし，もっと複雑な関数，たとえば有界変動でないものや，有界変動であっても全変動が積分値に比べて極端に大きいような場合は，もはや準モンテカルロ法すら適用することは普通できない．そのような場合，本質的にランダムなサンプリングによる数値積分法 — モンテカルロ積分 — が必要になる．

複雑な関数の数値積分は，確率変数の平均 (期待値) を求める場合によく現れる．なぜなら，確率空間上の関数として実現したとき，確率変数はたいてい非常に複雑だからである．しかしもちろん，複雑でない確率変数もたくさんあるわけで，それらは決定論的な方法で能率よく数値積分を行うことができる．もしある確率変数の数値積分にランダムサンプリングが必要だとしたら，それは‘ランダムな現象を記述する確率変数だから’ ではなく‘関数として複雑だから’ である．小論が念頭に置いているのは‘確率変数の数値積分’ だが，このような事情から表題を‘複雑な関数の数値積分 . . . ’ とした．

コンピュータによる数値積分にランダムサンプリングを用いるというアイデアはフォン ノイマン (von Neumann) に始まると言われている．ほぼコンピュータが生を受けた時期である．それは同時に‘ランダムサンプリングをコンピュータでどのように実現するか’ という問題の始まりでもあった．

コルモゴロフ (Kolmogorov) やチャイティン (Chaitin) らは 1960 年代に当時揺籃期の計算理論を援用して‘計算の複雑さ’ という概念を作り出し乱数の理論を築いた ([8, 17])．マルティン-レーフ (Martin-Löf) が‘万能検定’ と呼ばれるすべての検定より優位に立つ普遍的な検定を構成し，ある $\{0,1\}$-列が乱数であることと，それが万能検定に採択されることが同値であることを示すに及んで，乱数の理論は一応の完成を見た ([21])．しかし，乱数は数値計算にはまったく役に立たないものであった．なぜなら，乱数は‘それを書き下すことより簡便な生成アルゴリズムが存在しないような数列’ というのが定義であったから．さらに万能検定は現実的なスケールを無視してあくまで漸近的な理論の整合性を重んじて構成されているので，たとえ乱数が天下り式に与えられたとしても，それは現実的スケールの下でランダムに見えるとは限らない．

とはいえ，現実にランダムサンプリングの需要があるので，明確な理論を欠いたまま乱数モドキという意味の‘疑似乱数’ を生成する方法が次々と発案された．そして少なくとも工学的にはほぼ満足の行く成功を収めた ([13, 19, 22, 23])．

疑似乱数の明確な “定義” は，コルモゴロフからさらに 20 年を経て 1980 年代に暗号理論において出現する ([3, 4, 36])．それはコルモゴロフの複雑さをより実用的なレベルで考えた計算量の理論を基とする．それによれば，疑似乱数生成は標語的には‘小さなランダム性’ を‘大きなランダム性’ に見せ掛ける技術といえる．まるで鍍金 (メッキ) のように，貴重な少量の資源 (小さなランダム性) を表面の見えるところ (検定が実際に実行可能な程度の計算量を持つような有限次元分布) だ

†岩波書店，‘数学’，56-1，(2004)，1-17，より転載．

けに張り付けることが目的である．見えないところ (計算量が大きくて実際は実行不可能な検定) には気を配る必要はない．このことが達成されている疑似乱数は ‘安全’ といわれる (定義 4.1).

先に二重引用符付きで “定義” とした理由は，そのように定義した概念を満足するような数学的実体 — すなわち安全な疑似乱数 — が存在することの証明がまだ得られていないからである．それは ‘$\mathbf{P} \neq \mathbf{NP}$ 予想’ と絡む難しい問題である．それでも，疑似乱数生成の本質的な問題が明らかにされたことは大きな進展であろう．ただし，用途を限れば，‘少ないランダム性でもって大きなランダム性と同じ働きをさせる’ ということは実現の可能性がある．たとえばモンテカルロ積分に関しては，このことは達成されている，といってよい (§ 4.3).

複雑な関数の数値積分にランダムサンプリングが必要である一方で，複雑な関数に規則的なサンプリング — たとえば準モンテカルロ法 — を適用してみると，生成されるサンプル列が極めてランダムに見えることがしばしばある ([10, 11, 26, 27, 31, 32]). このことを利用して [26, 27] で構成された疑似乱数生成器の安全性について § 5.2 で考察する.

コンピュータによるランダムサンプリングは，応用科学に携わる人々が主たるユーザであることも手伝って直感的に議論されることが多く，数学的緻密さを伴わず流布している．残念ながら研究者に限ってさえ，標準的なコンセンサスが確立されていない，というのが実情である．従って現時点では，小論で述べることは一確率論研究者としての筆者の提言に過ぎないことをご理解頂きたい.

2　ルベーグ確率空間と硬貨投げの確率過程

最初に小論を通して用いる記号を準備しておく.

定義 2.1　(i) $\mathbb{T}^1$ を 1 次元トーラス (区間 $[0,1)$ 上に加法として $(x+y) \bmod 1$ を考えた群)，$\mathcal{B}$ を $\mathbb{T}^1$ 上のボレル (Borel) 集合族，$\mathbb{P}$ をルベーグ (Lebesgue) 測度とする．確率空間 $(\mathbb{T}^1, \mathcal{B}, \mathbb{P})$ をここではルベーグ確率空間と呼ぶ.
(ii) 各 $m \in \mathbb{N}$ に対して $d_m(x) \in \{0,1\}$ を $x \in \mathbb{T}^1$ の 2 進展開の小数点以下第 m 桁目とする．すなわち

$$d_1(x) = \mathbf{1}_{[1/2,1)}(x), \quad d_m(x) = d_1(2^{m-1}x), \quad x \in \mathbb{T}^1.$$

(iii) 各 $m \in \mathbb{N}$ に対して $\Sigma^m := \{\, i/2^m \mid i = 0, \ldots, 2^m - 1 \,\} \subset \mathbb{T}^1$ とおく．さらに $\mathcal{B}_m := \sigma\{\, [a,b) \mid a, b \in \Sigma^m,\, a < b \,\}$ とする．$x \in \mathbb{T}^1$ に対して

$$\lfloor x \rfloor_m := \lfloor 2^m x \rfloor / 2^m = \sum_{i=1}^{m} 2^{-i} d_i(x) \in \Sigma^m.$$

ただし，$\lfloor x \rfloor_\infty := x$ と約束する.
(iv) 各 $m \in \mathbb{N}$ に対して P_m を可測空間 $(\Sigma^m, 2^{\Sigma^m})$ 上の一様確率測度とする.

関数列 $\{d_m\}_{m=1}^{\infty}$ はルベーグ確率空間上の確率変数列として公平な硬貨投げの確率過程である (ボレルの硬貨投げのモデル [5, 6]). $\mathcal{B}_m$ は m 回までの硬貨投げ $\{d_i\}_{i=1}^{m}$ によって定まる事象の全体である．f が $\mathcal{B}_m$-可測であるための必要十分条件は $f(x) \equiv f(\lfloor x \rfloor_m)$ となることである．確率空間 $(\Sigma^m, 2^{\Sigma^m}, P_m)$ は m 回の硬貨投げを記述する．確率空間 $(\mathbb{T}^1, \mathcal{B}_m, \mathbb{P})$ は $(\Sigma^m, 2^{\Sigma^m}, P_m)$ と写像 $\lfloor \cdot \rfloor_m : \mathbb{T}^1 \to \Sigma^m$ によって同型である.

f が $\mathcal{B}_m$-可測のとき，小論では ‘f は (高々)m-ビットのランダム性を持つ’ ということにする．これは f が m 回の硬貨投げの関数であることを意識した用語である.

ルベーグ確率空間上の確率変数 f の平均 (期待値) を $I[f]$ と書く．すなわち

$$I[f] := \int_{\mathbb{T}^1} f(y)dy.$$

とくに確率空間を指定せずに与えられた確率変数については，平均を $\mathbf{E}$ で表すことにする．

3 複雑な関数の数値積分

3.1 ある数値計算例

やや迂遠であるが，後の説明のために数値計算の実例を挙げておこう．ルベーグ確率空間上に次の 2 つの確率変数を定義する: $x \in \mathbb{T}^1$ に対して

$$S_m(x) := \sum_{i=1}^{m} d_i(x), \tag{1}$$

$$f_m(x) := \mathbf{1}_{\{S_{2m-1}(\cdot)\leq m-1\}}(x) = \begin{cases} 1, & (S_{2m-1}(x) \leq m-1) \\ 0, & (S_{2m-1}(x) \geq m) \end{cases}. \tag{2}$$

前者は硬貨を m 回投げたときに表 ($d_i(x) = 1$) の出る回数を表す確率変数，後者は硬貨を $(2m-1)$ 回投げたとき表の出る回数が $(m-1)$ 回以下であれば 1 をそうでないときは 0 を返す確率変数，とそれぞれ解釈できる．

ここで f_{50} の積分値を求めることを考えてみよう．表裏が同等に出ることから $I[f_{50}] = \mathbb{P}(S_{99} \leq 49) = 1/2$ は容易に分かる．このように，理論的解析だけで積分値が計算できるのが，もちろん，最も望ましい．

次に望ましいのは，決定論的なアルゴリズムによって近似値が誤差評価と共に得られることである．数値解析学では，被積分関数の滑らかさに応じて様々な数値積分法が開発されている．しかし，f_{50} は確率論的対象としてはむしろ簡単な部類に属するにも拘らず，$\mathbb{T}^1$ 上の関数としては非常に多くの不連続点を持ち，解析的性質として分かることはせいぜい有界変動であることくらいであろう．有界変動関数に対する決定論的な数値積分法として**準モンテカルロ法** (quasi-Monte-Carlo method)，すなわち '小さい差異を持つ一様分布列 (sequence of low discrepancy)' によるサンプリング法がある．$[0,1)$-値数列 $\{x_n\}_{n=1}^{\infty}$ が小さい差異を持つとは，すべての $\mathbb{T}^1$ 上の有界変動関数 f に対して

$$\left| \frac{1}{N} \sum_{n=1}^{N} f(x_n) - I[f] \right| \leq c(N)||f||_{BV} \times \frac{\log N}{N}, \quad N \in \mathbb{N} \tag{3}$$

が成り立つことをいう ([18, 23])．ここに $||f||_{BV}$ は f の $\mathbb{T}^1$ 上の全変動，$c(\cdot) > 0$ は f に依存しない有界関数である．たとえばパラメータ $\alpha := (\sqrt{5}-1)/2$ としたときのワイル (Weyl) 変換 (無理数回転) による点列

$$x_n = (n\alpha) \bmod 1, \quad n = 1, 2, \ldots \tag{4}$$

に対して (3) が成り立つ．この場合，$c(N) = (3/\log N) + (1/\log \xi) + (1/\log 2)$，$\xi = (1+\sqrt{5})/2$，という評価が知られている ([18])．

しかし f_{50} が有界変動であるといっても $||f_{50}||_{BV}$ は巨大な値であり，この場合 (3) という評価はまったく実用的な価値を持たない．それでもとにかく，$x_n = (n\alpha) \bmod 1$，$N = 10^3 \sim 10^7$ として絶対誤差 $\left| N^{-1} \sum_{n=1}^{N} f_{50}(x_n) - (1/2) \right|$ を計算してみると図 3.1 のようになる．多少の揺らぎ

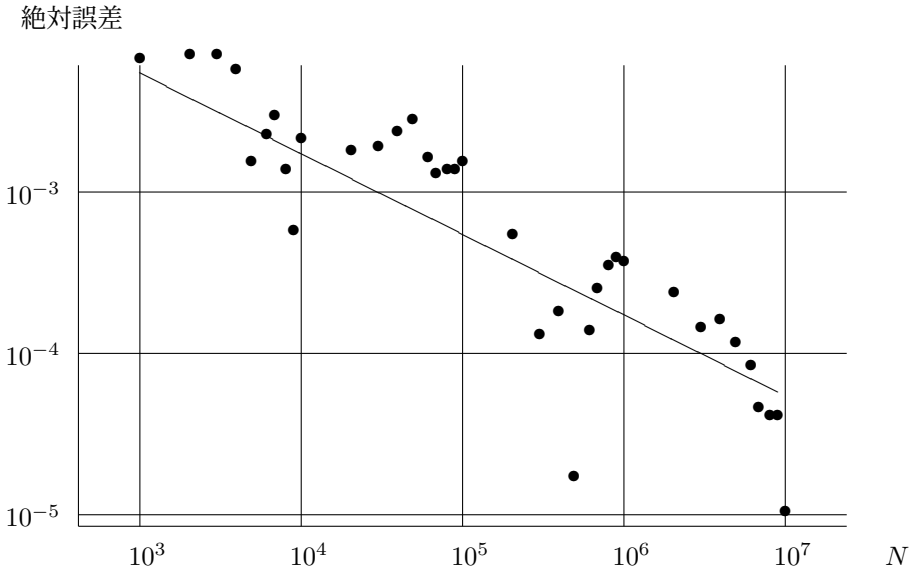

図 1: 点列 (4) による準モンテカルロ法の絶対誤差 (log-log グラフ)

はあるものの，確かに誤差は 0 に収束しているようであり，図 3.1 の左上から右下に引かれた直線の傾きは $-1/2$ であることから，その収束の早さはおよそ $O(N^{-1/2})$ である．

他の代表的な小さい差異を持つ一様分布列，ヴァン デル コープット列 (van der Corput sequence, cf. [18]) $\{v_n\}_{n=1}^{\infty}$ でも同じことを考えてみよう．ここで各 v_n は以下のように定義される: $d_{-i}(n)$ を非負の整数 n の 2 進展開における下から i 桁目のビット (i.e., $n = \sum_{i=1}^{\infty} d_{-i}(n)2^{i-1}$) として

$$v_n := \sum_{i=1}^{\infty} d_{-i}(n-1)2^{-i}, \qquad n = 1, 2, \ldots, \tag{5}$$

具体的に書き下せば

$$\{v_n\}_{n=1}^{\infty} = \left\{0, \frac{1}{2}, \frac{1}{4}, \frac{3}{4}, \frac{1}{8}, \frac{5}{8}, \frac{3}{8}, \frac{7}{8}, \frac{1}{16}, \frac{9}{16}, \frac{5}{16}, \frac{13}{16}, \ldots\right\}. \tag{6}$$

このときも，$x_n = v_n$ として (3) が成り立つ．この場合 $c(N) = \log(N+1)/(\log 2 \cdot \log N)$ でよい ([18])．$c(N)$ の比較では $x_n = (n\alpha) \bmod 1$ より $x_n = v_n$ の方が優れている．しかし，先程と同様に $N^{-1}\sum_{n=1}^{N} f_{50}(v_n)$ の計算をすると，$N < 2^{50} \quad 1.13 \times 10^{15}$ ならば $N^{-1}\sum_{n=1}^{N} f_{50}(v_n) = 1$ になってしまう．これでは現実問題として $\{v_n\}_{n=1}^{\infty}$ を用いて f_{50} を積分することはできない．

3.2 なぜ決定論的数値積分法は機能しないか

任意の $\mathcal{B}_m$-可測関数 f に適用できる数値積分法がどのようなものであるべきか，について考える．f の積分は $I[f] = 2^{-m}\sum_{y\in\Sigma^m} f(y)$ で与えられる．2^m が小さいうちは右辺の有限和を実際に計算することができる．問題は 2^m が巨大な数でこの有限和を実際に計算することができない場合である．仕方がないので，我々は $I[f]$ の近似値を 2^m よりずっと少ない N 個のサンプル点列 $\mathbf{x} := \{x_n\}_{n=1}^{N} \subset \Sigma^m$ における f の値 $\{f(x_n)\}_{n=1}^{N}$ から計算する．このような状況では決定論的数値積分法が機能しない場合があることを以下で説明する．

$\sum_{n=1}^N c_n = 1$ を満たす係数 $\mathbf{c} := \{c_n\}_{n=1}^N$ を用いて

$$I\left(f;\mathbf{c},\mathbf{x}\right) := \sum_{n=1}^N c_n f(x_n) \tag{7}$$

の形の数値積分公式を考えよう．$\mathcal{B}_m$-可測な実数値確率変数全体に内積 $\langle f,g\rangle := I[fg]$ を考えた内積空間を $L^2(\mathcal{B}_m)$ と書く．ノルムは通常どおり $||f|| := \langle f,f\rangle^{1/2}$ で与える．このとき

$$g_{\mathbf{c},\mathbf{x}}(y) := 2^m \sum_{n=1}^N c_n \mathbf{1}_{[x_n,x_n+2^{-m})}(y), \quad y \in \mathbb{T}^1 \tag{8}$$

で定まる $g_{\mathbf{c},\mathbf{x}} \in L^2(\mathcal{B}_m)$ によって $I\left(f;\mathbf{c},\mathbf{x}\right) = \langle g_{\mathbf{c},\mathbf{x}}, f\rangle$ であることに気づく．従って

$$I\left(f;\mathbf{c},\mathbf{x}\right) - I[f] \;=\; \langle g_{\mathbf{c},\mathbf{x}} - 1, f\rangle$$

である．つまり数値積分の誤差を小さくするためには f に対して $g_{\mathbf{c},\mathbf{x}} - 1$ がほぼ直交するように $\mathbf{c},\mathbf{x}$ をとればよい，ということが分かる．しかし，すべての $f \in L^2(\mathcal{B}_m)$ と $g_{\mathbf{c},\mathbf{x}} - 1$ がほぼ直交するように $\mathbf{c},\mathbf{x}$ をとることができるだろうか．明らかに，そんなことはできない．たとえば $g_{\mathbf{c},\mathbf{x}} - 1 \in L^2(\mathcal{B}_m)$ だから，$g_{\mathbf{c},\mathbf{x}} - 1$ がそれ自身と同じ方向を向いている．実際，$\langle g_{\mathbf{c},\mathbf{x}}, 1\rangle = 1$ であること，$||g_{\mathbf{c},\mathbf{x}}||^2 \geq 2^m \sum_{n=1}^N c_n^2$ であること (各 x_n が相異なれば等号成立)，および $\sum_{n=1}^N c_n^2 \geq 1/N$ であること ($c_n \equiv 1/N$ ならば等号成立) に注意すると

$$\begin{aligned} I(g_{\mathbf{c},\mathbf{x}} - 1;\mathbf{c},\mathbf{x}) - I[g_{\mathbf{c},\mathbf{x}} - 1] \;&=\; \langle g_{\mathbf{c},\mathbf{x}} - 1, g_{\mathbf{c},\mathbf{x}} - 1\rangle \;=\; ||g_{\mathbf{c},\mathbf{x}}||^2 - 1 \\ &\geq\; 2^m \sum_{n=1}^N c_n^2 - 1 \;\geq\; \frac{2^m}{N} - 1 \end{aligned} \tag{9}$$

となって，これは $N \ll 2^m$ ならば巨大な値になる．すなわちサンプル点列および係数列 $\mathbf{c},\mathbf{x}$ によるサンプリングは，それ自身に対応する関数 $g_{\mathbf{c},\mathbf{x}}$ を積分する際にとんでもなく大きい誤差を出してしまう．

以上の議論は極めて自明なことなのであるが，固定された有限個のサンプル点列で任意の複雑な関数を数値積分することの理論的な困難はこのような一種の‘自己双対性’にある．前節で見たヴァン デル コープット列 $\{v_n\}_{n=1}^\infty$ による確率変数 f_{50} の数値積分の失敗はこれに近い事態が実際に起こり得ることを示す例と言えよう．

注意 3.1 考えている被積分関数のクラスが $L^2(\mathcal{B}_m)$ の中で線形部分空間あるいはそれに準ずる薄い集合であれば，そのクラスに属するすべての関数 f と $g_{\mathbf{c},\mathbf{x}} - 1$ がほぼ直交するように $\mathbf{c},\mathbf{x}$ を取ることが可能であろう．実際，台形公式やシンプソン (Simpson) 公式などはすべて (7) の形をしており，それらの公式が適用できるような滑らかな関数族 (を離散化したもの) は $L^2(\mathcal{B}_m)$ で薄い集合になっていると考えられる．

3.3 モンテカルロ積分のシナリオ

サンプル点列として確率変数を用いて数値積分を行う手法を**モンテカルロ積分** (Monte-Carlo integration) という．ここにその考え方 [1)] を手短に整理して述べよう．

目的

まず，積分値を求めたい関数 $\tilde{f}$ と同じ (または非常に近い) 積分値を持つ確率変数 f をルベーグ確率空間上に構成する．次に，この f をもとにして，与えられた誤差限界 $\varepsilon > 0$ に対して f の積分値 $I[f]$ の近くに分布する 1 つの確率変数 X をやはりルベーグ確率空間上に構成する．小論では f はとくに 2 乗可積分であることを仮定して，平均 2 乗誤差が ε 未満，すなわち

$$I\left[|X - I[f]|^2\right] < \varepsilon \tag{10}$$

であるような確率変数 X を構成することを考える．このときチェビシェフ (Chebyshev) の不等式を用いて

$$\mathbb{P}\left(|X - I[f]| > \delta\right) < \varepsilon/\delta^2. \tag{11}$$

ただし f と X は有限のランダム性を持つか，あるいは後に述べる模倣可能な確率変数 (§ 3.7) でなければならない．

以上の要件を満たす f と X が構成された段階で，一応，理論的な目的は達成される．

ランダムサンプリング

$I[f]$ の近似値を実際に求めるために，ある $x \in \mathbb{T}^1$ における X の値 $X(x)$，を計算する．ただしその際，具体的にどの x であれば $X(x)$ が真値 $I[f]$ に近いかを前もって知ることは期待できないので (それができるくらいなら，初めから決定論的な近似値が求められるわけだ)，確実によい近似値が得られるわけではない．これは一種の賭けである (だからフォン ノイマンはカジノで有名な町に因んでモンテカルロ法と名付けたらしい)．計算の実行者 (以下，アリスとしよう) は $x \in \mathbb{T}^1$ を自由に選ぶ権利を持つと同時に，その結果責任を負う．そのとき (10) や (11) をこの賭けのリスクを客観的に評価する尺度と考える．このような考えに基づく問題解決を**ランダムサンプリング**と呼ぶ．ここで，'ランダム (= デタラメ，無作為)' という言葉にあまりとらわれない方がよい．たとえば (11) の主張する数学的内容は，X が $\mathcal{B}_M$-可測のときは，'$|X(x) - I[f]| > \delta$ を満たす $x \in \Sigma^M$ の個数が $2^M\varepsilon/\delta^2$ 未満である' ということであり，それ以上でも以下でもない．

疑似乱数生成器

ところが，X のランダム性 (=M-ビットとする) は非常に大きいことが多い．言い換えると，X の実現値を 1 つ計算するために非常に長いビット列 $x \in \Sigma^M$ の入力が必要となる．それをアリスが行うことの負担を肩代わりするために，疑似乱数生成器 $g : \Sigma^n \to \Sigma^M$ $(n \ll M)$ が用いられる．g はアリスに短いビット列の入力 $\omega \in \Sigma^n$ (種と呼ぶ) を要請し，これを長いビット列 $g(\omega) \in \Sigma^M$ (疑似乱数と呼ぶ) に変換する．それを用いてアリスは $X(g(\omega))$ を計算する．従って，実際にはアリスは疑似乱数の種 $\omega \in \Sigma^n$ を自由に選ぶ権利しか与えられない．そのときリスクはどのように評価されるべきだろうか．特別な場合を除いて $X(g(\omega))$ の分布を正確に求めることはできないにも拘らず，アリスはいつも $X(g(\omega))$ に関しても X とほぼ同じリスク評価，たとえば

$$P_n\left(|X(g(\omega)) - I[f]| > \delta\right) < \varepsilon'/\delta^2, \quad (\varepsilon' \text{ は } \varepsilon \text{ より僅かに大きくてよい})$$

が成り立つことを期待している．この期待に応えるために，疑似乱数生成器 g は 'P_n の下での $X(g(\omega))$ の分布が X の分布に十分近い' という性質を持っていなければならない．

3.4 i.i.d.-サンプリング

最も簡明でよく使われるランダムサンプリング法が，Σ^m 上の一様分布に従う独立確率変数列 $\{Z_n\}_{n=1}^N$ と均等な係数 $c_n \equiv 1/N$ を用いた **i.i.d.-サンプリング**である (i.i.d. は independently

identically distributed の略)．式で書けば

$$I_{\mathrm{iid}}(f;\{Z_n\}_{n=1}^N) := I(f;\{1/N\}_{n=1}^N,\{Z_n\}_{n=1}^N) = \frac{1}{N}\sum_{n=1}^N f(Z_n), \quad f\in L^2(\mathcal{B}_m). \tag{12}$$

ここで $\{Z_n\}_{n=1}^N$ は $Z_n = Z_n(x) := \sum_{i=1}^m 2^{-i} d_{(n-1)m+i}(x)$，$x\in\mathbb{T}^1$，としてルベーグ確率空間上に構成する．よく知られたように平均 2 乗誤差は次のようになる．

$$I\left[\left|I_{\mathrm{iid}}(f;\{Z_n\}_{n=1}^N) - I[f]\right|^2\right] = \frac{1}{N}\mathbf{Var}[f], \quad f\in L^2(\mathcal{B}_m). \tag{13}$$

ここで $\mathbf{Var}[f] := I\left[|f-I[f]|^2\right]$ は f の分散である．チェビシェフの不等式から

$$\mathbb{P}\left(\left|I_{\mathrm{iid}}(f;\{Z_n\}_{n=1}^N) - I[f]\right| > \delta\right) \le \frac{\mathbf{Var}[f]}{N\delta^2}, \quad \delta>0. \tag{14}$$

N がある程度大きければ，i.i.d.-サンプリングで得られる確率変数 (12) の分布は中心極限定理によって正規分布で近似されるから，その誤差は (14) よりもっと正確に見当をつけることができる．

f が $\mathcal{B}_m$-可測のとき，$X(x) := I_{\mathrm{iid}}(f;\{Z_n(x)\}_{n=1}^N)$ は $\mathcal{B}_{Nm}$-可測となる．$1\ll N$ のときは X のランダム性は f のそれよりもずっと大きい．

3.5 ランダムサンプリングに関するある不等式

一般のランダムサンプリングについて (13) のような明快な公式は得られないが，ある不等式が得られる．

定理 3.1 (cf. [28]) 関数系 $\{1,\psi_1,\dots\psi_{2^m-1}\}$ を $L^2(\mathcal{B}_m)$ の正規直交基底とする．このとき，任意の確率変数列 $\mathbf{X} := \{X_n\}_{n=1}^N \subset \mathbb{T}^1$，$1\le N\le 2^m$，に対して，次の不等式が成り立つ．

$$\sum_{l=1}^{2^m-1} \mathbf{E}\left[\left|I(\psi_l;\mathbf{c},\mathbf{X})\right|^2\right] \ge \frac{2^m}{N} - 1. \tag{15}$$

なお，$c_n\equiv 1/N$ でかつ確率 1 で $\{\lfloor X_n\rfloor_m\}_{n=1}^N$ がすべて異なる点であれば (15) は等式となる．

証明. 定理の主張の特別な場合として $\mathbf{X}$ が任意の決定論的数列 $\{x_n\}_{n=1}^N$ である場合も (15) を満たさなければならないし，またそうだとすると，任意の確率変数列 $\mathbf{X}\subset\mathbb{T}^1$ に対して (15) が従う．だから任意の決定論的数列 $\{x_n\}_{n=1}^N$ について示せばよい．

$\mathbf{x} := \{x_n\}_{n=1}^N \subset \Sigma^m$ としてよい．この列には (8) によって $g_{\mathbf{c},\mathbf{x}}\in L^2(\mathcal{B}_m)$ が対応している．パーセヴァル (Parseval) の等式 (ピュタゴラス (Pythagoras) の定理) から

$$||g_{\mathbf{c},\mathbf{x}}||^2 = \langle g_{\mathbf{c},\mathbf{x}},1\rangle^2 + \sum_{l=1}^{2^m-1}\langle g_{\mathbf{c},\mathbf{x}},\psi_l\rangle^2. \tag{16}$$

$\langle g_{\mathbf{c},\mathbf{x}},1\rangle^2 = 1$ および (9) に注意すれば (16) から

$$\frac{2^m}{N} \le 1 + \sum_{l=1}^{2^m-1}\left|I(\psi_l;\mathbf{c},\mathbf{x})\right|^2.$$

これは $X_n\equiv x_n$ とした場合の (15) に他ならない． □

さて，もしあるサンプリング法が — 決定論的であろうがランダムであろうが — ある性質のよい被積分関数のクラスに対して i.i.d.-サンプリングよりもずっと能率のよい数値積分法を与えるならば，そのサンプリング法はそのクラスに属さない被積分関数に安易に適用することは避けた方がよい．なぜなら，定理 3.1 によって必ず近似誤差が大きくなる被積分関数が — 不等式 (15) を満たすために — 存在するからである．これは 'High risk, high return' の原則の一つの例である．

それに対して i.i.d.-サンプリングは，以下に示すように，ほぼ最も 'Low risk, low return' なサンプリング法ということができる．規格化された平均 2 乗誤差の上限値

$$R(\mathbf{c},\mathbf{X}) := \sup_{f\in L^2(\mathcal{B}_m),\,\mathbf{Var}[f]=1} \mathbf{E}\left[\,|I(f;\mathbf{c},\mathbf{X})-I[f]|^2\,\right] \tag{17}$$

を $\mathbf{c},\mathbf{X}$ によるランダムサンプリング法の最悪時リスクと呼ぼう．i.i.d.-サンプリングの場合は (13) より，$R(\{1/N\}_{n=1}^N,\{Z_n\}_{n=1}^N)=1/N$ である．一般のランダムサンプリングの場合は (15) の両辺を ψ_l たちの個数 2^m-1 で割った不等式から，とくに興味深い $N\ll 2^m$ の場合を考えてみると，

$$R(\mathbf{c},\mathbf{X}) \;\geq\; \left(\frac{2^m}{2^m-1}\frac{1}{N}-\frac{1}{2^m-1}\right) \quad \frac{1}{N} \tag{18}$$

が導かれる．ここに最右辺は i.i.d.-サンプリングの場合の最悪時リスクに等しい．このことは，$N\ll 2^m$ のとき，i.i.d.-サンプリングは'あらゆるランダムサンプリング法 (もちろん決定論的方法も含む) の内で最悪時リスクがほぼ最小である' という特長を有することを示している．

3.6 ランダム性の削減

i.i.d.-サンプリングの場合の等式 (13) は，サンプル列 $\{f(Z_n)\}_{n=1}^N$ の独立性というより，無相関性 ($n\neq n'$ ならば $I[(f(Z_n)-I[f])(f(Z_{n'})-I[f])]=0$) に起因している．そこで，i.i.d. の代わりにたとえば 同分布で**ペアごとに独立**な確率変数列を用いても (13) と同様の等式を得ることができる．

例 3.1 (cf. [20]) Σ^m を 2 進小数展開を通して $\{0,1\}^m$ と同一視し，さらにこれをガロア (Galois) 体 $\mathrm{GF}(2^m)$ と同一視する．$\Omega:=\mathrm{GF}(2^m)\times\mathrm{GF}(2^m)$ とし，μ を Ω 上の一様確率測度とするとき，確率空間 $(\Omega,2^\Omega,\mu)$ 上の確率変数列 $\{X_n\}_{n=1}^{2^m}$ を次で定義する．

$$X_n(\omega) := x+n\alpha, \quad \omega=(x,\alpha)\in\Omega, \quad n\in\mathrm{GF}(2^m).$$

このとき μ の下で各 X_n は $\mathrm{GF}(2^m)$ 上一様分布し，$n\neq n'$ ならば X_n と $X_{n'}$ は独立である．

例 3.1 の事実は，$n\neq n', a\in\mathrm{GF}(2^m)$ を与えたとき 2 元連立 1 次方程式

$$\begin{cases} x+n\alpha &= a \\ x+n'\alpha &= a' \end{cases}$$

が唯一解 $(x,\alpha)=\omega_0=(x_0,\alpha_0)$ を有するので

$$\mu(X_n=a,\,X_{n'}=a') \;=\; \mu(\{\omega_0\}) \;=\; 2^{-2m} \;=\; \mu(X_n=a)\mu(X_{n'}=a')$$

となることから従う．Σ^m-値確率変数 X_1 と X_2 が独立ならば，少なくとも $2m$ ビットのランダム性が必要なのは明らかだから，例 3.1 より小さいランダム性でペアごとに独立な確率変数列を作る

ことはできない．しかし残念ながら GF(2^m) の四則演算は多項式の演算であるので，とくに乗法は m が少し大きいと大変手間が掛かるため数値計算には馴染まない．

この他にも [2, 9, 12, 15] などでペアごとに独立な確率変数列を構成している．その中でも，ランダム性は最小ではないものの，ずっと数値計算に適した例を次に挙げる．

例 3.2 (ランダム-ワイル-サンプリング (RWS)[28, 33]) $\Omega := \Sigma^{m+j} \times \Sigma^{m+j}$, $\mu := P_{m+j} \otimes P_{m+j} (=$ Ω 上の一様確率測度) とするとき，確率空間 $(\Omega, 2^\Omega, \mu)$ 上の Σ^m-値確率変数列 $\{X_n\}_{n=1}^{2^j}$ を次で定義する．

$$X_n(\omega) := \lfloor x + n\alpha \rfloor_m, \quad \omega = (x, \alpha) \in \Omega, \quad n = 1, \ldots, 2^j.$$

このとき μ の下で各 X_n は Σ^m 上一様分布し，$1 \le n \ne n' \le 2^j$ ならば X_n と $X_{n'}$ は独立である．とくに $\{f(X_n)\}_{n=1}^{2^j}$ は f と同分布のペアごとに独立な確率変数列である．

たとえば (2) で定義した確率変数 f_{50} の数値積分について考えよう．サンプルサイズが 10^7 の i.i.d.-サンプリングでは $99 \times 10^7 =$ 約 10^9 個のランダムビットが必要であるが，同じサンプルサイズの RWS ではわずか $\lceil 99 + \log_2 10^7 \rceil \times 2 = 246$ 個のランダムビットしか必要でない．この程度のビット数であればキーボードから十分打ち込めるので，疑似乱数生成器は必要ないだろう．

しかし，もちろん，RWS を用いてもなおキーボードから打ち込めないほど長いビット列の入力が必要になる場合もあるから，その場合は疑似乱数生成器を用いる．このとき，RWS の徹底したランダム性の削減は次のような利点をもたらす:

(a) RWS は疑似乱数の質に大変鈍感である ([33] の § 4).

(b) RWS ではほとんどサンプルの生成速度を落とすことなく，遅いが精密な疑似乱数生成器 (たとえば安全なもの．§ 4.2 を参照) を用いることができる．そしてその結果は，極めて信頼性の高いものになる．

注意 3.2 § 3.1 においてワイル変換による f_{50} の数値積分が ‘幸運にも’ うまく行くことを見たが，それは $(x, \alpha) = (0, \lfloor (\sqrt{5} - 1)/2 \rfloor_{123}) \in \Sigma^{123} \times \Sigma^{123}$ とした場合の RWS を実行したのだと思えば，そのような幸運は非常にしばしば起こることが分かる．

3.7 模倣可能な確率変数とその数値積分

計算資源 (記憶容量と計算時間) の有限性から，コンピュータで厳密に扱うことのできる確率変数は有限なランダム性を持つもの (ある m に対して $\mathcal{B}_m$-可測) に限る．しかし，無限のランダム性を持つ (どんな m に対しても $\mathcal{B}_m$-可測とならない) が，確率 1 で有限個のランダム性しか必要としないような関数のあるクラスは，小さな確率を除いてコンピュータで厳密に扱うことができる．次の例を見よ．

例 3.3 (到達時刻)

$$\sigma(x) := \inf\{n \ge 1 \,|\, d_1(x) + d_2(x) + \cdots + d_n(x) = 5\}, \quad x \in \mathbb{T}^1$$

とすれば σ は硬貨を投げ続けて，表の出た回数が 5 となる最初の時刻である (ただし，$\inf \emptyset = \infty$ と解釈する). 明らかに σ は非有界で無限のランダム性を持つ．しかし，表が出た回数が 5 になった時点でその後の $d_i(x)$ の値は不要だから直ちに計算を終えて $\sigma(x)$ を算出することができる．すなわち，確率 1 で $\sigma(x)$ の計算は有限時間で終わる．

関数 $\tau : \mathbb{T}^1 \to \mathbb{N} \cup \{\infty\}$ は $\{\tau \leq m\} := \{x \in \mathbb{T}^1 \,|\, \tau(x) \leq m\} \in \mathcal{B}_m, \forall m \in \mathbb{N}$, を満たすとき，$\{\mathcal{B}_m\}_m$-**停止時刻** (cf. [5, 7]) という．$\{\mathcal{B}_m\}_m$-停止時刻 τ に対して，$\mathcal{B}$ の部分 σ-加法族 $\mathcal{B}_\tau$ を $\mathcal{B}_\tau := \{A \in \mathcal{B} \,|\, A \cap \{\tau \leq m\} \in \mathcal{B}_m, \forall m \in \mathbb{N}\}$ で定義する．関数 $f : \mathbb{T}^1 \to \mathbb{R} \cup \{\pm\infty\}$ が $\mathcal{B}_\tau$-可測関数であるための必要十分条件は $f(x) = f\left(\lfloor x \rfloor_{\tau(x)}\right), x \in \mathbb{T}^1$, となることである．

例 3.3 の σ は $\{\mathcal{B}_m\}_m$-停止時刻であり，σ 自身 $\mathcal{B}_\sigma$-可測である．一般に停止時刻 τ が $\mathbb{P}(\tau < \infty) = 1$ であれば $\mathcal{B}_\tau$-可測関数 f はコンピュータ上で小さな確率を除いて実現できる．そのような f を**模倣可能** (simulatable, cf. [7]) な確率変数と呼ぼう．

模倣可能な確率変数 f に付随している停止時刻 τ が $\mathbf{E}[\tau] < \infty$ であれば $\mathcal{B}_\tau$-可測な f に i.i.d.-サンプリングを施すことは簡単である．実際

$$\mathbf{z}_n(x) := \lfloor 2^{\sum_{i=1}^{n-1} \tau(\mathbf{z}_i(x))} x \rfloor_{\tau(\mathbf{z}_n(x))}, \quad x \in \mathbb{T}^1, \quad n \in \mathbb{N}$$

とすれば $\{f(\mathbf{z}_n)\}_{n=1}^\infty$ はルベーグ確率空間上の確率変数列として f と同分布の i.i.d. である (τ が $\mathbb{P}(\tau < \infty) = 1$ なる停止時刻なので $\mathbf{z}_n$ は確率 1 で定義される)．

さらに，そのような f に対してもペアごとに独立なサンプリングを行うことが可能である．

定義 3.1 (動的ランダム-ワイル-サンプリング (DRWS), [29, 30]) $j, K \in \mathbb{N}$ とし，$\{(x_l, \alpha_l)\}_{l \in \mathbb{N}}$ を $\Sigma^{K+j} \times \Sigma^{K+j}$-値で一様分布する i.i.d. 確率変数列とする．τ を $\mathbf{E}[\tau] < \infty$ なる $\{\mathcal{B}_m\}_m$-停止時刻とし，$\mathbb{T}^1$-値確率変数列 $\{\mathbf{x}_n\}_{n=1}^{2^j}$ を

$$\begin{aligned} \mathbf{x}_n &:= \sum_{l=1}^{\lceil \tau(\mathbf{x}_n)/K \rceil} 2^{-(l-1)K} \left\lfloor x_l + \nu_{n,l} \alpha_l \,(\mathrm{mod}\, 1) \right\rfloor_K, \\ \nu_{n,l} &:= \#\{\, 1 \leq i \leq n \mid \tau(\mathbf{x}_i) > (l-1)K \,\} \end{aligned}$$

で定義する (τ が $\mathbf{E}[\tau] < \infty$ なる停止時刻なので $\nu_{n,l}$ および $\mathbf{x}_n$ は確率 1 で定義される)．

定理 3.2 ([29]) ルベーグ確率空間 $(\mathbb{T}^1, \mathcal{B}, \mathbb{P})$ 上の確率変数 f が $\mathcal{B}_\tau$-可測のとき，各 $f(\mathbf{x}_n)$，$1 \leq n \leq 2^j$，は f と同分布で，$1 \leq n \neq n' \leq 2^j$ ならば $f(\mathbf{x}_n)$ と $f(\mathbf{x}_{n'})$ は独立である．

4 疑似乱数生成器

4.1 一般的枠組み

パラメータ $n \in \mathbb{N}$ を伴った関数 $g_n : \Sigma^n \to \Sigma^{\ell(n)}$ は $n < \ell(n)$ のとき，**疑似乱数生成器** (pseudo-random generator) と呼ばれる．$\omega_n \in \Sigma^n$ は Σ^n 上の一様分布 P_n に従う確率変数で，**種** (seed) と呼ばれ計算の実行者がランダムに選ぶとし，これを g_n の入力とする．そして出力 $g_n(\omega_n) \in \Sigma^{\ell(n)}$ を**疑似乱数** (pseudo-random bits, pseudo-random numbers) という．$n < \ell(n)$ なので $g_n(\omega_n)$ は硬貨投げの確率過程ではない．

疑似乱数の検定を定式化しよう．$\Sigma^{\ell(n)}$ の部分集合 A で $P_{\ell(n)}(A) \ll 1$ なるものを選べば，それを棄却域とする検定が考えられる．すなわち，$P_n(g_n(\omega_n) \in A)$ が危険率 $P_{\ell(n)}(A)$ に近ければ疑似乱数 $g_n(\omega_n)$ はこの検定に採択される．このことを少し拡張して以下のように検定を定式化する．パラメータ $n \in \mathbb{N}$ を伴った関数 $A_n : \Sigma^{\ell(n)} \times \Sigma^{s(n)} \to \{0, 1\}$ に対して

$$\delta(g_n, A_n) := \left| P_n \otimes P_{s(n)} \left(A_n(g_n(\omega_n), \omega_{s(n)}) = 1 \right) - P_{\ell(n)} \otimes P_{s(n)} \left(A_n(\omega_{\ell(n)}, \omega_{s(n)}) = 1 \right) \right|$$

を定義する．ここで $\otimes$ は確率測度の直積を表す．$\delta(g_n, A_n)$ は関数 A_n によって疑似乱数 $g_n(\omega_n)$ と硬貨投げ $\omega_{\ell(n)}$ がどれだけ区別できるかを表す．ω_n と独立で $P_{s(n)}$ に従う確率変数 $\omega_{s(n)}$ を導

入したのはランダムな検定をも許すことを意味する．当て推量でもよいから疑似乱数を公平な硬貨投げから区別するよう試みよ，というわけである．

4.2 安全な疑似乱数生成器

4.2.1 定義

$n < \ell(n)$ だから $\delta(g_n, A_n) \geq 1/2$ となる関数 A_n が必ず存在する (A_n が g_n の値域の定義関数の場合を考えよ)．従ってすべての検定，すなわちすべての関数 A_n，を考えたのではもとより望みがない．そこで，A_n のクラスを制限して考えよう．このとき，各々のクラスに応じて，そのクラスに属する検定には採択されるがそうでない検定には棄却されることを許されるような疑似乱数のクラスが定義される．

それでは検定のクラスは何を基準に定めればよいか．暗号理論では，その基準を計算量におく．すなわち，A_n の時間計算量を $T(A_n)$ とおくとき

$$S(g_n, A_n) := \frac{T(A_n)}{\delta(g_n, A_n)}$$

を定義する．すべての A_n に対して $S(g_n, A_n)$ が十分大きければ，疑似乱数生成器 g_n は‘優れている’と考えられる．すなわち，$T(A_n)$ が非常に大きくて現実問題として A_n による検定が不可能である場合には $\delta(g_n, A_n)$ は小さくなくてもよく，一方，$T(A_n)$ があまり大きくない A_n，すなわち実際に実行できる検定，に対しては $\delta(g_n, A_n)$ は十分小さくてはならない，というとき g_n を優れていると考えるのである．

理論的な研究の対象として，次の定義で規定される疑似乱数生成器のクラスが最も簡単で，かつよく調べられている．

定義 4.1 疑似乱数生成器 g_n の定義に現れる $\ell(n)$ および $T(g_n)$ は n について高々多項式増大であると仮定する．もし $S(g_n, A_n)$ がどのような A_n に対しても多項式増大を超えるならば，g_n は**安全な**疑似乱数生成器 (または疑似乱数 $g_n(\omega_n)$ は**安全**である) と言われる．

注意 4.1 たとえば，$e^{t/1000}$ はあらゆる多項式より早く増大するが，それは t が十分大きいときのことで，比較的小さい t のときは t^{100} の方がずっと大きい．従って実際用いられる程度のサイズの問題では，必ずしも‘時間計算量が多項式増大の関数は実際に計算できる関数であり，それを超える時間の関数は実際には計算できない関数’というわけではない．実際に計算ができるかどうかは別途調査する必要があるだろう．

4.2.2 安全な疑似乱数生成器とモンテカルロ法

‘暗号通信に用いたときに非常に高い確率で通信の秘密を守ることができる’というのが‘安全’の元来の意味であるが，モンテカルロ法においても，硬貨投げの確率過程の代わりにその疑似乱数を用いて確率変数を実現したとき分布の違いが検出される確率は非常に小さい，という意味で安全である．このことを説明しよう．

§ 3.3 で述べたように，一般にモンテカルロ法で扱われる確率変数 X はランダム性の大きいものが多い．いま，X のランダム性は $\ell(n)$-ビットであるとしよう．さて，一般に $\ell(n)$ は大きすぎるので，$\omega_{\ell(n)} \in \Sigma^{\ell(n)}$ の代わりに疑似乱数 $g_n(\omega_n)$，$\omega_n \in \Sigma^n$，を用いる．つまり，X の代わりに

$X' := X(g_n(\omega_n))$ を数値計算に用いるわけである．そこで問題は X と X' の分布が十分近いかどうか，である．それで，それぞれの分布関数 $F_X(t) := P_{\ell(n)}(X \leq t)$，$F_{X'}(t) := P_n(X' \leq t)$ を比較することを考えよう．もし g_n が安全な疑似乱数生成器であれば，n を十分大きくとるとき，$F_X(t)$ と $F_{X'}(t)$ は十分近いことが保証される．実際，検定の関数 A_n を $A_n(\omega_{\ell(n)}) := \mathbf{1}_{\{X(\omega_{\ell(n)})\leq t\}}$，$\omega_{\ell(n)} \in \Sigma^{\ell(n)}$，とおけば，‘$X$ が実際に計算できる’ という事実から，$T(A_n)$ は十分小さいはずである．すると，安全な疑似乱数生成器の定義より

$$|F_X(t) - F_{X'}(t)| \;=\; \left|P_{\ell(n)}\left(A_n(\omega_{\ell(n)}) = 1\right) - P_n\left(A_n(g_n(\omega_n)) = 1\right)\right|$$

は十分小さくなければならない．

4.2.3 存在問題

理論的観点からいえば，安全な疑似乱数生成器は，疑似乱数生成器のクラスの中で最も簡単で自然なクラスであろう．しかしながら，残念なことに，果たしてそれが存在するかどうかが厳密には分かっていないのである．それは，計算量理論において最も重要な予想の 1 つ ‘$\mathbf{P} \neq \mathbf{NP}$ 予想’ に関連する．すなわち，もし $\mathbf{P} = \mathbf{NP}$ ならば安全な疑似乱数生成器は存在しない [2)]．

$\mathbf{P} \neq \mathbf{NP}$ の真偽が不明なので，安全な疑似乱数生成器は現時点ではその存在が証明できていない．しかしながら，研究者たちは楽観的である．彼等は考える．もちろん，安全な疑似乱数生成器が存在すればすばらしい．一方，もし安全な疑似乱数生成器だと思われていたものがそうでなかったら，$\mathbf{P}\neq\mathbf{NP}$ 予想に進展が見られるだろう．いずれにしろ，そうでないと分かるまでは，その疑似乱数生成器は有効である，と．我々も，決着が付くまでは，安全な疑似乱数生成器の存在を仮定しよう．

4.2.4 次ビット予測

安全な疑似乱数生成器をどのようにして構成すべきか，について重要な手掛りになるのが，次ビット予測の概念である．

定義 4.2 $g_n : \Sigma^n \to \Sigma^{\ell(n)}$ を疑似乱数生成器とする．集合 $\{1, 2, \ldots, \ell(n)\} \times \Sigma^n$ 上の一様確率測度を $P_{\ell(n),n}$ で表す．以下では確率変数 $(I, \omega_n) \in \{1, 2, \ldots, \ell(n)\} \times \Sigma^n$ は $P_{\ell(n),n}$ に従うとする．関数 $\tilde{A} : \{1, 2, \ldots, \ell(n)\} \times \Sigma^{\ell(n)} \times \Sigma^{s(n)} \to \{0, 1\}$ に対して

$$\tilde{\delta}(g_n, \tilde{A}_n) \;:=\; P_{\ell(n),n} \otimes P_{s(n)}\left(\tilde{A}_n\left(I, \lfloor g_n(\omega_n)\rfloor_{I-1}, \omega_{s(n)}\right) = d_I(g_n(\omega_n))\right) - \frac{1}{2}$$

とする．これは $\tilde{A}_n$ が $\lfloor g_n(\omega_n)\rfloor_{I-1}$ から次のビット $d_I(g_n(\omega_n))$ を正確に予測する確率である．このとき任意の $\tilde{A}_n$ に対して

$$\tilde{S}(g_n, \tilde{A}_n) \;:=\; \left|\frac{T(\tilde{A}_n)}{\tilde{\delta}(g_n, \tilde{A}_n)}\right|$$

が多項式増大を超えるとき，g_n は**次ビット予測不可能**という．

次ビット予測は § 4.2.1 で述べた検定の特殊なものと見なすことができるから，もちろん，安全な疑似乱数生成器は次ビット予測不可能である．ところがこの逆も成り立つのである．

定理 4.1 (cf. [20, 25]) 疑似乱数生成器 g_n が安全であるための必要十分条件は，それが次ビット予測不可能であることである．

定理 4.1 によって，安全な疑似乱数生成器を構成するためには，次のビットが予測されにくいようにさえ工夫すればよいことが分かる．一般論としては，一方向関数 (h の計算は易しいが逆関数 h^{-1} の計算が著しく大変なもの) の存在 (これは $\mathbf{P} \neq \mathbf{NP}$ より強い) を仮定して，次ビット予測不可能な疑似乱数生成器を構成する方法が知られている．このことを利用した疑似乱数生成器として最初に提案されたのは BBS 生成器と呼ばれるものである ([3, 4, 25, 36])．

4.3 特殊な用途の疑似乱数

今までは暗黙のうちに，様々な用途に用いることが可能な疑似乱数，いわば汎用疑似乱数について述べてきた．しかしながら，とくにモンテカルロ法においては，もっぱら特殊な用途に疑似乱数を用いることがある．用途を限った場合，'小さいランダム性でありながら大きなランダム性と同じ働きをする疑似乱数' は，$\mathbf{P}\neq\mathbf{NP}$ 予想などとは無関係に，実現されても不思議でない．なぜなら，用途を限る，ということは，採択されることが要求される検定のクラスが予め決まっている，ということであり，それは計算量理論を論拠としないかも知れないからである．

そうした例はすでに数値積分に見られる．(D)RWS は平均 2 乗誤差において，i.i.d.-サンプリングと同一の性能を有する数値積分法であった．なおかつ，前者は後者よりもずっと小さいランダム性しか持たない．つまり，(D)RWS の確率変数列は，もちろん，すべての検定に採択される訳ではないが，'数値積分において i.i.d.-サンプリングと同一の性能を有するか' という検定においては，平均 2 乗誤差を見る限り，採択されるのである．

このような意味で，(D)RWS は '数値積分専用の疑似乱数生成器' ということができる．

5 複雑性と疑似乱数の安全性

5.1 従属性の消滅

§ 3.1 の図 1 で，我々は $|N^{-1}\sum_{n=1}^{N} f_{50}(n\alpha) - (1/2)|$ が $O(N^{-1/2})$ のオーダーで減少するかのような状況を見た．これは定理 3.1 によればほぼ平均的に見られる状況であると思えるだろう．実は，被積分関数が非常に複雑なときには，準モンテカルロ法の収束がほぼ $O(N^{-1/2})$，i.i.d.-サンプリングと同じ速さ，であるかのように観察されることが広く知られている ([7])．この現象は直感的には以下のように説明される: 被積分関数が非常に複雑なときには，そもそも数値積分を正確に行うことが本質的に困難であり，どのような方法も目立って能率が上がらない．そのため，実際にはまったくランダムな i.i.d.-サンプリングと同程度の能率しか得られない．

ならば，複雑な被積分関数に準モンテカルロ法を適用したときのサンプル列がランダムに見えるかも知れない，と考えるのは自然であろう．実際，そのような '従属性の消滅' が起こっていることがいくつかの例で示されている．たとえば，

定理 5.1 ([10]) $S_m(x)$ を (1) で定義した関数とする．このとき，$\mathbb{P}$-a.e. α に対して，ルベーグ確率空間 $(\mathbb{T}^1, \mathcal{B}, \mathbb{P})$ 上の確率過程 $\{2m^{-1/2}(S_m(\cdot + n\alpha) - (m/2))\}_{n=1}^{\infty}$ は $m \to \infty$ のとき，標準正規分布に従う i.i.d. 確率変数列に有限次元分布の意味で収束する．

$m \to \infty$ のとき，この確率過程の各 1 次元分布は中心極限定理によって正規分布に収束する．確率変数列としてガウス系 (Gaussian system) に収束することを見るのは，すぐにはできないが，

その代わり，ここでは相関が 0 に収束する仕組みを見よう．$r_i(x) := 1 - 2d_i(x)$ はラデマッハ (Rademacher) 関数列と呼ばれるもので $S_m(x) - (m/2) = (r_1(x) + \cdots + r_m(x))/2$ である．

$$\varphi(\alpha) := \frac{1}{4} I\big[r_1(\cdot) r_1(\cdot + \alpha)\big] = \left| \frac{1}{2} - \alpha \right| - \frac{1}{4} \tag{19}$$

とおく．

$$r_i(x) = r_1(2^{i-1}x), \qquad I\big[r_i(\cdot) r_j(\cdot + \beta)\big] = 0, \quad \forall \beta, \quad i \neq j,$$

に注意して共分散を計算してみよう．

$$\begin{aligned} & I\left[m^{-1/2} \left(S_m(\cdot) - \frac{m}{2} \right) \cdot m^{-1/2} \left(S_m(\cdot + n\alpha) - \frac{m}{2} \right) \right] \\ &= \frac{1}{4m} \sum_{i=1}^{m} \sum_{j=1}^{m} I\big[r_i(\cdot) r_j(\cdot + n\alpha)\big] = \frac{1}{4m} \sum_{i=1}^{m} I\big[r_i(\cdot) r_i(\cdot + n\alpha)\big] \\ &= \frac{1}{4m} \sum_{i=1}^{m} I\big[r_1(2^{i-1}\cdot) r_1(2^{i-1} \cdot + 2^{i-1} n\alpha)\big] = \frac{1}{4m} \sum_{i=1}^{m} I\big[r_1(\cdot) r_1(\cdot + 2^{i-1} n\alpha)\big] \\ &= \frac{1}{m} \sum_{i=1}^{m} \varphi(2^{i-1} n\alpha). \end{aligned} \tag{20}$$

ここで最後の式は，変換 $\alpha \mapsto 2\alpha$ のエルゴード性のために，$m \to \infty$ のとき，ほとんどすべての α に対して積分値 $I[\varphi] = 0$ に収束する ([35])．もっとも (20) の収束は遅く，ほぼ $O(m^{-1/2})$ 程度である ([10, 16])．

次の定理では，もっと急速に従属性消滅が起こっている．

定理 5.2 ([1, 26, 27, 34, 37, 38, 39]) $(\mathbb{T}^1, \mathcal{B}, \mathbb{P})$ 上の $\{0,1\}$-値確率過程 $\{X_n^{(m)}(\cdot\,; \alpha)\}_{n=1}^{\infty}$，$\alpha \in \mathbb{T}^1$，を

$$X_n^{(m)}(x; \alpha) := S_m(x + n\alpha) \bmod 2, \quad x \in \mathbb{T}^1, \quad n = 1, 2, \ldots, \tag{21}$$

によって定義する．ここで $S_m(x + n\alpha) \bmod 2$ は $S_m(x + n\alpha)$ を 2 で割った余りを表す．このとき $\mathbb{P}$-a.e. α に対して [3] $\{X_n^{(m)}(\cdot\,; \alpha)\}_{n=1}^{\infty}$ は $m \to \infty$ のとき，公平な硬貨投げの確率過程に分布収束する．さらに $\mathbb{P}$-a.e. α について $\{X_n^{(m)}(\cdot\,; \alpha)\}_{n=1}^{\infty}$ の各有限次元分布への収束は指数関数的である．

5.2 次ビット予測との関係

定理 5.2 を計算量の観点から見ることも興味深い．つまり，$\{X_n^{(m)}(\cdot\,; \alpha)\}_{n=1}^{\infty}$ に対して次ビット予測を考えるとき，m の増加に対して関数 $S_m(x) \bmod 2$ の複雑さが増すので，次ビットの予測がより困難になるであろうことが想像できる．このことの確率論的反映が定理 5.2 と考えられる．

少し詳しく解説しよう．まず，$1 \leq k_0 < k_1 < \ldots < k_{l-1}$ に対して次の確率を明示的に求めるアルゴリズムが存在する ([26, 27])．

$$F^{(m)}(k_0, k_1, \ldots, k_{l-1}; \alpha) := \mathbb{P}\left(\sum_{j=0}^{l-1} X_{k_j}^{(m)}(\,\cdot\,; \alpha) = \text{奇数} \right).$$

そこで次のような次ビット検定を考える．

$$\tilde{A}(x_1, x_2, \ldots, x_{k_{l-1}-1}) := \begin{cases} \mathbf{1}_{\{\sum_{j=0}^{l-2} x_{k_j} = \text{偶数}\}}, & \text{if } F^{(m)}(k_0, k_1, \ldots, k_{l-1}; \alpha) > \frac{1}{2} \\ \mathbf{1}_{\{\sum_{j=0}^{l-2} x_{k_j} = \text{奇数}\}}, & \text{if } F^{(m)}(k_0, k_1, \ldots, k_{l-1}; \alpha) < \frac{1}{2} \end{cases}$$

とするとき，容易に分かるように

$$\mathbb{P}\left(\tilde{A}(X_1^{(m)}(\cdot\,;\alpha),\ldots,X_{k_{l-1}-1}^{(m)}(\cdot\,;\alpha))=X_{k_{l-1}}^{(m)}(\cdot\,;\alpha)\right)=\left|F^{(m)}(k_0,k_1,\ldots,k_{l-1};\alpha)-\frac{1}{2}\right|+\frac{1}{2}$$

が成り立つ．すなわち $\tilde{A}$ による次ビット予測は 1/2 より

$$\left|F^{(m)}(k_0,k_1,\ldots,k_{l-1};\alpha)-\frac{1}{2}\right|$$

だけ大きい確率で的中する．ところがこの量は $\mathbb{P}$-a.e. α について指数関数的に 0 に収束するのである．このことは，上の次ビット予測が m の増加とともに急速に困難になっていくことを示している．このように，m の増加とともに急速に次ビットが予測されにくくなる様子を解析的に見ることができることは，特殊な場合ではあるとはいえ，大変興味深い．

以上のような事実から，(21) で定義された確率過程 $\{X_n^{(m)}(\cdot\,;\alpha)\}_{n=1}^{\infty}$ は安全な疑似乱数かも知れない，と期待するのは虫が良すぎるだろうか．

なお，定理 5.1 より，$\mathbb{P}$-a.e. α に対して (2) で定義した $\{f_m(\cdot+n\alpha)\}_{n=1}^{\infty}$ もルベーグ確率空間上の確率変数列として硬貨投げの確率過程に分布の意味で収束するが，相関の消滅が $O(m^{-1/2})$ 程度なので，これは疑似乱数として安全ではないことが分かる．

6　おわりに

小論では表題も含めて ‘複雑な関数’ という言葉を何度も使った．これは単に ‘全変動が大きい関数’ という意味ではない．実際，$d_{100}(x)$ は全変動が非常に大きいのにも拘らず，無理数回転の軌道 (4) による数値積分は容易である．それは漠然と ‘計算量が大きい’ という意味なのであるが，筆者が表現したい ‘複雑さ’ は既存の計算量の概念で測ることができるのか，それとも新しい概念が必要なのか，筆者自身にも不明である．知りたいのは ‘複雑さ’ を表す量と解析学 (とくに確率論) 的現象とを結ぶ定量的関係である．たとえば，ルベーグ確率空間上の 2 つの確率変数列 $\{f_{50}(\cdot+n\alpha)\}_{n=1}^{\infty}$ と $\{S_{50}(\cdot+n\alpha) \bmod 2\}_{n=1}^{\infty}$ は，§ 5.2 で述べたことからも分かるように，後者の方がずっと硬貨投げの確率過程に近い．こうしたことが，関数 f_{50} と $S_{50} \bmod 2$ の複雑さとどのような定量的関係にあるのか，といったことに強く興味を引かれる．

注

1) ‘筆者が最も合理的と思っている考え方’ である．異なる考え方を持つ人たちもいるだろう．
2) $\mathbf{P} \neq \mathbf{NP}$ を仮定しても，安全な疑似乱数生成器が存在するかどうか分からない．
3) 最近，安富健児 ([40]) によって，すべての無理数 $\alpha \in \mathbb{T}^1$ に対して $\{X_n^{(m)}(\cdot\,;\alpha)\}_{n=1}^{\infty}$ が $m \to \infty$ のとき，公平な硬貨投げの確率過程に分布収束することが示された．

参考文献

[1] 秋根善孝，従属性消滅定理に関する収束速度の精密評価，九州大学大学院数理学研究院修士論文，(2002).

[2] N.S. Bahvalov, Optimal convergence bounds for quadrature processes and integration methods of Monte Carlo type for classes of functions, (Russian), Ž. Vyčisl. Mat. i Mat. Fiz., **4-4** (1964), suppl., 5–63.

[3] L. Blum, M. Blum and M. Shub, A simple unpredictable pseudorandom number generator, SIAM J. Comput., **15-2** (1986), 364–383.

[4] M. Blum and S. Macali, How to generate cryptographically strong sequences of pseudo-random bits, SIAM J. on Computing, **13** (1984), 850–864. A preliminary version appears in Proceedings of the IEEE Foundations of Comput. Sci., (1982), 112–117.

[5] P. Billingsley, Probability and measure, 3rd edition, John Wiley & Sons, (1995).

[6] E. Borel, Sur les probabilités dénombrables et leurs applications arithmétiques, Circ. Mat. d. Palermo, **29** (1909), 247–271.

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

[8] G.J. Chaitin, Algorithmic information theory, IBM J. Res. Develop., **21** (1977), 350–359.

[9] B. Chor and O. Goldreich, On the power of two-point based sampling, J. Complexity, **5-1** (1989), 96–106.

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

[11] K. Fukuyama, Riesz-Raikov sums and Weyl transform, Monte Carlo Methods and Applications, VSP, **2-4** (1996), 271–293.

[12] O. GoldreichOded and A. Wigderson, Tiny families of functions with random properties: a quality-size trade-off for hashing, Proceedings of the Workshop on Randomized Algorithms and Computation (Berkeley, CA, 1995). Random Structures Algorithms, **11-4** (1997), 315–343.

[13] 伏見正則，乱数，東京大学出版会，(1989)

[14] JIS Z 9031 乱数発生及びランダム化の手順，日本規格協会，2001 年改正.

[15] A. Joffe, On a set of almost deterministic k-independent random variables, Ann. Probability, **2-1** (1974), 161–162.

[16] M. Kac, On the distribution of values of sums of type $\sum f(2^k t)$, Ann. Math., **47** (1946), 33–49.

[17] A.N. Kolmogorov, Logical basis for information theory and probability theory, IEEE Trans. on Inform. Theo., vol.IT-**14-5**, Sept. (1968), 662–664.

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

[19] D.E. Knuth, 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)

[20] M. Luby, Pseudorandmness and cryptographic applications, Princeton Computer Science Notes, Princeton University Press, (1996).

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

[22] M. Matsumoto and T. Nishimura, Mersenne twister: a 623-dimensionally equidistributed uniform pseudo-random number generator, ACM. Trans. Model. Comput. Simul., **8-1**, (1998) 3–30.

[23] H. Niederreiter, Quasi-Monte Carlo methods and pseudo-random numbers, Bull. Amer. Math. Soc., **84** (1978), 957–1041.

[24] J. von Neumann, Various techniques used in connection with random digits, U.S. Natl. Bur. Stand. Appl. Math. Ser., **12** (1951), 36–38.

[25] D.R. Stinson, Cryptography (Theory and practice), CRC Press, Boca Raton/ Ann Arbor / London / Washington,D.C., (1995).

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

[27] 杉田洋，無理数回転による疑似乱数生成，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).

[28] H. Sugita, Robust numerical integration and pairwise independent random variables, Jour. Comput. Appl. Math., **139** (2002), 1–8.

[29] H. Sugita, Dynamic random Weyl sampling for drastic reduction of randomness in Monte Carlo integration, Math. Comput. Simulation, **62** (2003), 529–537.

[30] H. Sugita, The Random Sampler, 疑似乱数生成と動的ランダム-ワイル-サンプリングのための C/C++ 言語ライブラリ，下記にて公開:
http://idisk.mac.com/hiroshi_sugita/Public/imath/mathematics.html.

[31] H. Sugita and S. Takanobu, Limit theorem for symmetric statistics with respect to Weyl transformation: Disappearance of dependency, J. Math. Kyoto Univ., **38-4** (1998), 653–671.

[32] H. Sugita and S. Takanobu, Limit theorem for Weyl transformation in infinite-dimensional torus: Disappearance of dependency, J. Math. Sci. Univ. Tokyo, **7** (2000), 99–146.

[33] H. Sugita and S. Takanobu, Random Weyl sampling for robust numerical integration of complicated functions, Monte Carlo Methods and Appl., **6-1** (1999), 27–48.

[34] S. Takanobu, On the strong-mixing property of skew product of binary transformation on 2-dimensional torus by irrational rotation, Tokyo J. Math., **25-1** (2002), 1–15.

[35] P. Walter, An introduction to ergodic theory, Springer, (1981).

[36] A. Yao, Theory and applications of trapdoor functions, Proceedings of the IEEE Foundations of Comput. Sci., (1982), 80–91.

[37] 安富健児，Weyl 変換に関する従属性消滅定理のエルゴード論的証明，修士論文 (神戸大学大学院自然科学研究科), (2001).

[38] K. Yasutomi, A limit theorem for sequences generated by Weyl transformation: Disappearance of dependence, Probab. Theory Related Fields, **124-2** (2002), 178–188.

[39] K. Yasutomi, A direct proof of dependence vanishing theorem for sequences generated by Weyl transformation，to appear in Jour. Math. Kyoto Univ.

[40] K. Yasutomi, A dependence vanishing theorem for sequences generated by Weyl transformation, preprint.

(2003 年 4 月 14 日提出)

(すぎた　ひろし・大阪大学大学院理学研究科)