モンテカルロ法，乱数，および疑似乱数[1]

杉田 洋

大阪大学大学院理学研究科数学専攻

[1]2007 年度「確率論サマースクール」講義ノート．(ver.20070725)

はじめに

モンテカルロ法は「シミュレーションや数値計算を乱数を用いて行なう手法」(Wikipedia) などといわれている．1940 年代第二次世界大戦末期に，フォン・ノイマン (von Neumann) やウラム (Ulam) らが米国のロスアラモス研究所で核分裂物質中の中性子の拡散のシミュレーションをした[1] のが，モンテカルロ法の始まりであるらしい．それ以来，コンピュータの発達に伴い，モンテカルロ法はあらゆる科学技術領域で大いに活用されてきた．しかし「コンピュータでは乱数を生成することはできない」という原理的困難があって，乱数の代わりに疑似乱数 (乱数のように見えるが、実際には確定的な計算によって求めている数列 (Wikipedia)) を用いて計算している．そのため，じつはモンテカルロ法の数学的基盤は磐石ではない．

そこで，このノートでは，モンテカルロ法にできるだけ強固な数学的基盤を提供することを目指す．

そうした試みは 1960 年代にコルモゴロフらが “計算の複雑さ” という考えを用いて乱数を定義したときにすでに芽生えていた．それは，1930 年代に始まった計算理論を駆使した画期的なものであった．コルモゴロフらの仕事からおよそ 20 年後の 1980 年代に，疑似乱数という概念を暗号理論の文脈でブラム (Blum) らが定義している．それは，より現実的な計算可能性を論じる計算量の理論を援用して誕生したが，現在において情報通信の秘密保持に関する理論の基本的部分を支えている．だが，残念なことに，コルモゴロフの乱数の概念も，ブラムの疑似乱数の概念も，モンテカルロ法の数学的基盤を強化することには繋がらなかった．前者は難解な上に現実的な解決をもたらさないこと，後者は暗号理論はモンテカルロ法と関係ないと考えられたこと，が理由でモンテカルロ法の研究者たちはそれらに注目しなかったのである．

しかし，このノートでは，むしろこれら乱数と疑似乱数の定義を拠り所として，それらと理論的整合性を持つようにモンテカルロ法の数学的定式化を行う．そのお陰で，モンテカルロ法の全貌が明確になり部分的だが強固な数学的基盤ができあがった．実際，我々の定式化の下で，次のことが見えてくる: 疑似乱数を用いたモンテカルロ法でも数学的に正当化できる可能性が残されている．とくにモンテカルロ積分 (大数の法則を利用した数理統計的手法による数値積分法) の場合には，乱数と同等の働きをする疑似乱数を具体的に構成することができる．

以下，順を追ってこのノートの内容を概観しよう．§ 1 では，上で述べたモンテカルロ法の数学的定式化を行う．そこではモンテカルロ法の目的とあり方が，乱数，および疑似乱数のおおまかな定義とともに規定され，それら相互の有機的な関係が明らかになる．この最初の § は短いけれどもきわめて重要な考え方を含んでおり，ぜひ，しっかり理解して欲しい．

§ 2 では，計算の複雑さと呼ばれる概念を用いた乱数の定義とその基本的性質を紹介する．コルモゴロフの公理論的確率論は，しばしば「ランダムとは何であるか，という根源的な問いを避けたもの」といわれているが，一方で，彼はこの § で紹介するとおり，この根源的な問いに見事な解答を与えているのである．

[1] 原子爆弾製造の目的のために．

§ 3 の前半 § 3.1 では，“計算量的に安全な疑似乱数生成器” について，その基本的な事項を紹介する．そこでは計算量的安全性がモンテカルロ法におけるサンプリングに用いられる疑似乱数生成器の性質として有用であることが述べられる．また，この性質を持つ疑似乱数生成器が存在するかどうかは未解決予想 **P** ≠ **NP** と関連することを述べ，疑似乱数生成の本質的困難の所在を明らかにする．

§ 3 の後半 § 3.2 では，筆者が見出したある疑似乱数生成器について詳しく述べる．その疑似乱数生成器は § 3.1 で述べた “次ビット予測不可能性” の実現を意図したものであり，確率論的に接近しうるものである．§ 3.3 では § 3.2 で示した諸定理の証明を (日本語の文献が手に入らないということもあるので) 紙数を顧みずできるだけ丁寧な証明を与えた．

§ 4 では，筆者が最も実用的と考えるモンテカルロ積分のためのサンプリング法について解説する．それは § 1.5.2 で紹介したモンテカルロ積分のための疑似乱数生成器と同様にペアごとに独立なサンプルを生成する方法であり，それよりも，ずっと実用的なものである．とくに § 4.4 で紹介する動的ランダム-ワイル-サンプリングは，モンテカルロ法で扱い得るすべての確率変数の数値積分に適用できる大変実用的な方法である．

§ 5 では，§ 3.2 の疑似乱数生成器と，§ 4.2，§ 4.4 のモンテカルロ積分法について C 言語による実装例を掲載する．

このノートは，筆者が今まで作成してきたいろいろなファイルを掻き集めてまとめたもので，記号の使い方など一部で不統一が残ってしまった．たとえば，文字 A は複数の § で登場するが，その意味するところは集合であったり，関数であったり，まちまちである．文脈は明白なので誤解はないと信じるが，読者の寛容を乞う次第である．

2007.7.25

杉田 洋

目 次

1　モンテカルロ法の数学的定式化

モンテカルロ法を賭けとして定式化し，そのサンプリングにおける本質的な問題である乱数の問題，およびその解決を目指す疑似乱数生成器について基本事項を述べる．この § では個々の概念等については概略を紹介するに留め，厳密な定義等は後の §§ で紹介しそれぞれ掘り下げて説明する．

最初に，全編を通じて用いられる数学記号をいくつか列挙しておく．

$$
\begin{aligned}
\mathbb{N} &:= \{1, 2, \ldots\}, \quad \text{自然数全体} \\
\mathbb{Z} &:= \{\ldots, -2, -1, 0, 1, 2, \ldots\}, \quad \text{整数全体} \\
\mathbb{R} &:= \text{実数全体} \\
\mathbf{1}_B(x) &:= \text{集合 } B \text{ の定義関数} \\
\Pr &:= \text{確率 (とくに確率空間を明示しない場合に用いる．)} \\
\mathbf{E}[X] &:= \text{確率変数 } X \text{ の平均 (期待値)} \\
\mathbf{Var}[X] &:= \text{確率変数 } X \text{ の分散}
\end{aligned}
$$

1.1　概要

§ 1 の概要を述べる．詳しいことは後で述べるので，ここでは流れを理解して欲しい．

モンテカルロ法では，それぞれの具体的な問題に応じて確率空間 $(\Omega, \mathcal{F}, P)$ — § 1 では簡単のため，有限回の硬貨投げの確率空間

$$\Omega = \{0,1\}^L \ (L \in \mathbb{N}), \quad \mathcal{F} = 2^\Omega, \quad P = P_L = \Omega \text{ 上の一様確率測度},$$

に限るが — と，その上で定義された確率変数 $S : \{0,1\}^L \to \mathbb{R}$ を設定する．そして，一つの $\omega \in \{0,1\}^L$ を選んで S の ω における値 $S(\omega)$ を算出する．以下，これを**サンプリング** (標本抽出) と呼ぶ．

モンテカルロ法は，その名の由来のとおり，**賭け**の一種である．プレーヤー (以下，アリスと呼ぶ) の目的は S の**一般的な値** — S のとる値としてごくありふれた値，例外的でない値 — をサンプリングすることである (§ 1.2.1)．アリスは S の例外的な値をサンプリングしてしまうリスクを承知の上でサンプリングを行う．そのリスクは，S の例外値の集合を B とするとき，

$$A := \left\{\omega \in \{0,1\}^L \,;\, S(\omega) \in B\right\}$$

の確率 $P_L(A)$ でもって評価される．

S が例外値をとることは滅多にないから $P_L(A) \ll 1$ である．[2] 当然，アリスは自分が S の一般的な値を手に入れることはほとんど確実だと思うだろう．L が小さいときは確かにそのとおりである．しかし，$L \gg 1$ (§ 1.2.2 の例題では $L = 2^{26}$) の場合は事情が異なる．その場合，アリスが一つの $\omega \in \{0,1\}^L$ を選ぶときの具体的な方法が問題となる．実際，任意の $\omega \in \{0,1\}^L$ を指定するには，コンピュータに L ビットのデータを入力する必要だが，

[2] $a \ll b$ は「a は b よりずっと小さい」の意，また $a \gg b$ は「a は b よりずっと大きい」の意．

$L = 2^{26}$ ともなればそれは 8MB にもなり,[3] それをキーボードから直接打ち込むのはあまりに膨大な時間と労力がかかるので事実上不可能である．何らかの工夫が必要である．ところが，じつはどのような工夫をしようとも，アリスが (膨大な時間と労力をかけずに) 自分の意志で選ぶことのできる $\omega \in \{0,1\}^L$ は非常に少数であり特殊なものに限られてしまう (§ 1.3)．そのため，$P_L(A) \ll 1$ であったとしても，アリスが S の一般的な値をほとんど確実に手に入れることができる，とはとてもいい切れないのである．

このことから，リスク評価 $P_L(A)$ に実質的な意味を持たせるためには，アリスの意志では選ぶことのできない ω — すなわち**乱数** — によるサンプリングが必要であると考えられる．しかし一方，現実の問題に現れる S は何らかの意味のある量を表す確率変数であり，任意の確率変数ではない．すなわち，アリスが自分の意志で選ぶことのできる $\omega \in \{0,1\}^L$ は確かに特殊だが，S も確率変数 ($\{0,1\}^L$ 上の関数) 全体の中ではきわめて特殊なのである．もしかしたら，特殊な S なら特殊な ω によって — つまり，乱数を用いなくても — 一般的な値をサンプリングすることができるかもしれない．それを実現しようと企てるのが疑似乱数生成器である．

疑似乱数生成器とは短い $\{0,1\}$-列を長い $\{0,1\}$-列に写す写像のことである．たとえば疑似乱数生成器 $g : \{0,1\}^n \to \{0,1\}^L$，$n < L$，を用いたモンテカルロ法では，アリスは**種**と呼ばれる $\{0,1\}$-列 $\omega' \in \{0,1\}^n$ を自分の意志で選ぶ．(ここで任意の $\omega' \in \{0,1\}^n$ をコンピュータのキーボードから人が容易に打ち込める程度に n は小さい必要がある．) コンピュータは ω' をもとに疑似乱数 $g(\omega') \in \{0,1\}^L$ を算出し，さらにこれを S に代入して $S(g(\omega'))$ を出力する．

疑似乱数生成器 g を用いることによって，アリスは「$S(g(\omega'))$ が S の一般的な値であるかどうか」という新しい別の賭けを行うことになる．そのリスクは確率 $P_n\,(g(\omega') \in A)$ によって評価される．もちろん，この確率は疑似乱数生成器 g に依存するが，もし $P_n(g(\omega') \in A) \ll 1$ であるような疑似乱数生成器 g を見つけることができたなら，アリスは S の一般的な値を膨大な時間と労力をかけずに高い確率で手に入れることができる．この場合，乱数は必要でない．このような g を A **に対して安全な疑似乱数生成器**という (§ 1.4.2).

一般の確率変数 S については，その例外値を与える ω の集合 A に対して安全な疑似乱数生成器をどのように構成したらよいかは，その候補はいくらか挙がっているものの，知られていない（§ 1.4.3，§ 3.1)．しかし，疑似乱数生成器の用途を**モンテカルロ積分**に限った場合，すなわち S がある独立同分布確率変数列の標本平均であるような場合，S の例外値を与える ω の集合 A に対して安全な疑似乱数生成器が具体的に構成できて (§ 1.5.2)，大きすぎない規模のモンテカルロ積分の場合はすでに実用化されている (§ 4.2).

1.2 賭けとしてのモンテカルロ法

数学の問題はもちろん確実な方法で解けるに越したことはない．しかし，非常に複雑な問題あるいは詳しい情報が不足しているような問題では確率的ゲーム，つまり賭け，として定式化し，正しい解が求まらないリスクを承知の上で近似解を探すことが実際的である場合がある．モンテカルロ法はまさにそのような場合の一つである．

[3] 漢字 4,194,304 文字分，400 字詰原稿用紙で約 10,000 枚分の情報量.

1.2.1 プレーヤーの目的

モンテカルロ法 (Monte-Carlo method) とは，確率変数のサンプリングを行うことによって数学的問題を数値的に解く手法をいう．数学的にはモンテカルロ法を賭けとして定式化するのが適切である．この賭けのプレーヤー，アリスの目的は与えられた確率変数の一般的な値— ごくありふれた，特殊でない，例外的でない値 — をサンプリングすることである．(それが問題解決に結び付くようにあらかじめ問題を設定しておく．詳しくは後のいくつかの例を参照せよ．) もちろん，不運にも一般的でない値，すなわち例外的な値をサンプリングしてしまうこともある．数学的には，そのような場合の起こる確率をできるだけ正しく見積もること — リスクの評価 — が要請される．

次に非常に小規模なモンテカルロ法の例を示す.

例 1　r を未知の実数とする．32 枚のカードがあり，そのうち 31 枚には実数 r が，残りの 1 枚には実数 $10r$ が書いてある．アリスがその中から 1 枚を選び，そのカードに書かれた数でもって r の値を推定する．アリスが不運にも r の値を正しく推定し損ねる確率は 1/32 である．

この例を賭けの形式で述べるならば以下のようになる:「アリスの目的は r の書かれたカード (今の場合の “一般的な値”) を選ぶことである．選んだカードの数が r ならばアリスの勝ち， $10r$ ならばアリスの負け，である．このとき，アリスの負ける確率は 1/32 である (リスクの評価).」

なお，一般にモンテカルロ法ではプレーヤーの目的が達成されたかどうかはサンプリングの後でも分からないのが普通である．たとえば例 1 では，リスクは正確に評価されるものの，アリスの推定した r の値が正しいかどうか — つまり賭けとしての勝ち負け— は，カードを選んだ後も知ることはできない.[4]

じつは例外的な値を求めたい場合もないわけではない．たとえば，複雑な確率変数 X の最小値をモンテカルロ法で探索するという場合がある．これは数学的には X の滅多に実現しない値を求めることになる．このような場合，別の確率変数 S をうまく定義して，S の一般的な値が X のその滅多に起きない値になるようにする．次の例を見よ．

例 2　$\Pr(X < c) = 1/10000$ であると仮定しよう．すなわち，X は c 未満の値を取り得るがその確率はきわめて小さい．ここで $\{X_i\}_{i=1}^{40000}$ を X の独立なコピーとし，$S := \min_{1\le i\le 40000} X_i$ とする．このとき

$$\Pr(S < c) \;=\; 1 - \left(1 - \frac{1}{10000}\right)^{40000} \;\sim\; 1 - e^{-4} \;=\; 0.981\ldots$$

となる．だから，S は十分高い確率で c 未満の値を取ることになる．

[4] 人生における様々な選択も多くの場合，賭けであろう．果たして自分の選んだ手が良い手だったかどうか，結局分からないということはよくあるではないか....

1.2.2 一つの例題

例 1 を実行するのには，32 枚のカード以外に何も特別な道具は必要ない．しかし，現実のモンテカルロ法で扱う賭けは大規模であり，高い情報処理能力を持つコンピュータが必要である．§ 1 では次の例題をモンテカルロ法で解くことを中心に考えて行く．

例題 硬貨投げを $2^7 = 128$ 回行うとき表が続けて 7 回以上出る確率 p を求めよ．

モンテカルロ法では数理統計学における推定の方法を適用する．「硬貨投げを 2^7 回行う」という試行を独立に N 回繰り返して，そのうち「表が続けて 7 回以上出る」ということが起った回数を S_N とする．このとき N が十分大きければ大数の法則により S_N/N の値が高い確率で求めたい p のよい推定値となる．より具体的に:

例 3 $N = 2^{19} = 524,288$ とする．$S_{2^{19}}/2^{19}$ の平均と分散は

$$\mathbf{E}\left[\frac{S_{19}}{2^{19}}\right] = p, \quad \mathbf{V}\left[\frac{S_{19}}{2^{19}}\right] = \frac{p(1-p)}{2^{19}} \leq \frac{1/4}{2^{19}} = \frac{1}{2^{21}}$$

なので**チェビシェフの不等式** (Chebyshev's inequality) により

$$\Pr\left(\left|\frac{S_{2^{19}}}{2^{19}} - p\right| \geq 2^{-8}\right) \leq 2^{-21} \cdot \frac{1}{(2^{-8})^2} = 2^{-5} = \frac{1}{32}, \tag{1}$$

が成り立つ．言い換えれば，$S_{2^{19}}/2^{19}$ の一般的な値をサンプリングできれば，それは p の近似値になっている．

例 3 では (1) によってリスクが評価されている，と考えてよいだろう．

もちろん，本物の硬貨を $2^7 \times 2^{19} = 2^{26} = 67,108,864$ 回投げて S_{19} の値を計算するのではない．コンピュータを用いる．数学的形式に則って考えるために，例 3 の $S_{2^{19}}$ を硬貨投げの確率空間 ($\Omega = \{0,1\}^{2^{26}}, 2^\Omega, P_{2^{26}} =$ 一様分布) 上で次のように実現しよう: まず，関数 $X : \{0,1\}^{2^7} \to \{0,1\}$ を

$$X(\xi_1, \ldots, \xi_{2^7}) := \max_{1 \leq n \leq 2^7 - 6} \prod_{k=n}^{n+6} \xi_k, \qquad (\xi_1, \ldots, \xi_{2^7}) \in \{0,1\}^{2^7},$$

と定義する．これは $(\xi_1, \ldots, \xi_{2^7})$ の中で 1 が 7 個以上続いた箇所があるとき $X = 1$ そうでないとき $X = 0$ であることを意味する．次に $S_{2^{19}} : \{0,1\}^{2^{26}} \to \mathbb{Z}$ を各 $\omega = (\omega_1, \ldots, \omega_{2^{26}}) \in \{0,1\}^{2^{26}}$ に対して

$$S_{2^{19}}(\omega) := \sum_{k=1}^{2^{19}} X(\omega_{2^7(k-1)+1}, \ldots, \omega_{2^7 k})$$

と定義する．$S_{2^{19}}$ の例外値を与える ω の集合 A_0 を

$$A_0 := \left\{\omega \in \{0,1\}^{2^{26}} ; \left|\frac{S_{2^{19}}(\omega)}{2^{19}} - p\right| \geq 2^{-8}\right\} \tag{2}$$

とすれば，(1) より $P_{2^{26}}(A_0) \leq 1/32$ となる．そこで例 3 は次のような賭けとして捉えることができる．

例 4 一つの $\omega \in \{0,1\}^{2^{26}}$ をアリスが選んだとき，$\omega \notin A_0$ ならば勝ち，$\omega \in A_0$ ならば負け，という賭けを考える．この賭けでアリスが負ける確率は 1/32 以下である．

1.3 乱数の問題

例 1 と例 4 は規模の違いを除けば，数学上の違いはない．しかし，この規模の違いこそが実際に後者の賭けを行うときに本質的な問題を引き起こす．

今，アリスが例 4 の賭けを実行するために一つの $\omega \in \{0,1\}^{2^{26}}$ を選ぼうとしているとしよう．しかし，じつは $\{0,1\}^{2^{26}}$ の元のうちアリスが自分の意志で選ぶことができる元はきわめて少数であり，従って特殊な元なのである．だから，たとえ 計算上のリスク $P_{2^{26}}(A_0) \ll 1$ であっても，アリスが実際にこの賭けに勝つことは必ずしも容易ではない．

事情を説明しよう．各々の $\omega \in \{0,1\}^{2^{26}}$ は $2^{26} = 67,108,864$ ビット，すなわち 8MB のデータであり，アリスにはこれを直接キーボードからコンピュータに入力することなど，とてもできない．たとえばアリスはせいぜい 1,000 ビットのデータ (アルファベット 125 字分) までならキーボードから直接コンピュータへ入力できると仮定しよう．コンピュータは彼女の入力をもとに $\{0,1\}^{67108864}$ の一つの元をあるアルゴリズムによって生成するとする．[5] このとき，アリスが自分の意志で選ぶことのできる $\omega \in \{0,1\}^{2^{26}}$ の個数はせいぜい $2^{2^{10}+1} = 2^{1025}$ に過ぎない．(なぜなら l-ビットのデータ全体の個数が 2^l で従って l-ビット以下のデータ全体の個数が $2^0 + 2^1 + 2^2 + \cdots + 2^l = 2^{l+1} - 1$ だから．) $\{0,1\}^{2^{26}}$ の元の個数は $2^{2^{26}} = 2^{67108864}$ もあるのだから，アリスの選ぶことのできる $\omega \in \{0,1\}^{2^{26}}$ がいかに僅かであるかが分かるだろう．

たとえ，アリスが $2^{26} - 10$ ビットのデータまで入力できると仮定しても，アリスの選ぶことのできる ω の個数はせいぜい $2^{2^{26}-9}$ であり，これは $\{0,1\}^{2^{26}}$ 全体の個数 $2^{2^{26}}$ の 1/512 に過ぎない．言い換えると $\{0,1\}^{2^{26}}$ 全体の少なくとも 511/512 は $2^{26} - 9$ ビット以上の入力がどうしても必要な $\{0,1\}$-列なのである．

その $\{0,1\}$-列自身の長さと同程度の長さの入力がなければ指定できない $\{0,1\}$-列のことをコルモゴロフは**乱数**と呼んだ．[6] 乱数を指定するには，結局，それ自身を入力するより簡潔な方法は本質的に存在しない．$L \gg 1$ のとき，$\{0,1\}^L$ の元のうち圧倒的多数は乱数である．にもかかわらず，人智ではそのうちたった一つでさえ取り出すことができない．この不条理こそ，大規模なモンテカルロ法におけるサンプリングの本質的な問題なのである．

1.4 疑似乱数生成器

引き続き，例 4 の賭けについて考える．リスク評価 $P_{2^{26}}(A_0) \leq 1/32$ において，リスクを測る確率測度 $P_{2^{26}}$ は各 $\omega \in \{0,1\}^{2^{26}}$ を同様に確からしく選ぶことができる，ということを前提にしている．しかし，前節の議論で，アリスが自分の意思で選ぶことのできる $\omega \in \{0,1\}^{2^{26}}$ は非常に少数である．このことから，リスク評価 $P_{2^{26}}(A_0) \leq 1/32$ に実質的な意味を持たせるためには，アリスは人智を超えた方法 — 物理現象を用いて ω を生成するとか (物理乱数という)[7] — で乱数を得てサンプリングするしかないように思われる．

否，じつは悲観するのはまだ早過ぎる...

---

[5] じつは，数学的にはそのアルゴリズムこそが後に述べる疑似乱数生成器と呼ばれるものである．

[6] IT 時代の今日の用語を使えば，長い $\{0,1\}$-列を短い入力で指定できるとすると，その短い入力は元の長いデータを “圧縮したもの” ということができる．この意味で乱数とは “圧縮不可能なデータ” である．

[7] 事実，たとえば [8] に付属の乱数表は本質的に物理乱数で作成されている．

1.4.1 定義と役割

例 4 の賭けを実行するとき，$S_{2^{19}}(\omega)$ を計算するためにアリスはとにかく一つの $\omega \in \{0,1\}^{2^{26}}$ を選ばなければならない．それには何らかの工夫が必要である．ここでは最もよく用いられる工夫，すなわち疑似乱数生成器を用いる場合を考えることにしよう．

定義 1　$n < L$ のとき，関数 $g : \{0,1\}^n \to \{0,1\}^L$ を**疑似乱数生成器** (pseudo-random generator) という．ここで g の入力 $\omega' \in \{0,1\}^n$ を**種** (seed)[8]，出力 $g(\omega') \in \{0,1\}^L$ を**疑似乱数** (pseudo-random number) という．

疑似乱数というのは数列であるが，数学的対象としては，それを生み出す関数の方が重要である．それを疑似乱数生成器という．疑似乱数生成器は**初期化**[9]という手続きを経て疑似乱数を生成する．初期化というのは種 $\omega' \in \{0,1\}^n$ を一つ選ぶことである．プログラムはそれをもとに疑似乱数 $g(\omega')$ を生成する．実用のためには，当然，$n \in \mathbb{N}$ は種 ω' がキーボードから入力可能な程度に小さいことが必要である．また，関数 g を実現するプログラムが許容できる程度に短く，早く動作することが必要である．

例 5　例 4 において，アリスがたとえばある疑似乱数生成器 $g : \{0,1\}^{2^8} \to \{0,1\}^{2^{26}}$ を使うとしよう．アリスは g の種 $\omega' \in \{0,1\}^{2^8}$ を一つ選んでキーボードからコンピュータに入力する．ω' は $2^8 = 256$ ビット (アルファベット 32 文字分) のデータだから，キーボードから入力するのは困難ではない．そこでコンピュータは $S_{2^{19}}(g(\omega'))$ を計算する．

疑似乱数生成器を用いる理由は，アリスの入力すべきデータ $\omega \in \{0,1\}^{2^{26}}$ がキーボードから入力するにはあまりにも長大だからであった．入力データが短くて済む場合は疑似乱数生成器は必要ない．たとえば例 1 で 32 枚のカードの中から 1 枚選ぶとき，誰が疑似乱数生成器の利用を考えるだろうか．

1.4.2 安全性

例 5 の場合を引き続き考える．アリスは疑似乱数生成器 g の種 $\omega' \in \{0,1\}^{2^8}$ を自由に選ぶことができる．今の場合，リスクは確率

$$P_{2^8}\left(\left|\frac{S_{2^{19}}(g(\omega'))}{2^{19}} - p\right| \geq 2^{-8}\right) \tag{3}$$

であり，次にこれを計算する必要がある．もちろん，確率 (3) は g に依存する．もし，この確率 — すなわちアリスの選んだ ω' から計算された $S_{2^{19}}(g(\omega'))$ が S の例外的な値である確率 — が大きいとすると，アリスは目的を達成することが困難になる．それでは困る．

そこで，次の (かなり漠然とした) 定義を設けよう．

定義 2　疑似乱数生成器 $g : \{0,1\}^n \to \{0,1\}^L$，$n < L$，が集合 $A \subset \{0,1\}^L$ に対して**安全** (secure) であるとは

$$P_L(\omega \in A) \approx P_n(g(\omega') \in A)$$

が成り立つことをいう．[10]

[8] 初期値ともいう．
[9] ランダマイズ (randomize) ともいう．
[10] $x \approx y$ は x と y の値がほぼ等しいの意．

例 5 において，もし $g : \{0,1\}^{2^8} \to \{0,1\}^{2^{26}}$ が (2) で定義された集合 A_0 に対して安全であれば，アリスが自分の意志で選ぶことのできる $\omega' \in \{0,1\}^{2^8}$ の大多数に対して $S(g(\omega'))$ は S の一般的な値を与えることが分かる．この場合，乱数は必要でない．言い換えると，S の一般的な値をサンプリングしたい際に g を用いてもリスクを大きくしないという意味で，このような g を安全な疑似乱数生成器と呼ぶわけである．

一般に，できるだけ多くの集合 A に対して安全であるような g が望ましい疑似乱数生成器といえる．しかしすべての A に対して安全であるような疑似乱数生成器は存在しない．実際，疑似乱数生成器 $g : \{0,1\}^n \to \{0,1\}^L$ が与えられたとき

$$A_g := g(\{0,1\}^n) \subset \{0,1\}^L$$

とすれば，$P_L(\omega \in A_g) \leq 2^{n-L}$ であるが，$P_n(g(\omega') \in A_g) = 1$ となるので g は A_g に対して安全ではない．従って，疑似乱数生成器の安全性を論じるときは集合 A のクラスを制限して考えなければならない．

1.4.3 計算量的に安全な疑似乱数生成器

疑似乱数生成器の安全性を論じる上で，対象となる集合の考えうる限り最も大きなクラスは，以下に述べる「計算量的に判定可能であるような集合」のクラスである．

疑似乱数生成器 $g : \{0,1\}^n \to \{0,1\}^L$，$n < L$，が集合 $A \subset \{0,1\}^L$ に対して安全であるかどうかを判定する具体的手続きについて考えよう．そのためには，何はさておき，与えられた $\omega \in \{0,1\}^L$ が $\omega \in A$ を満たすかどうかを判定するプログラムを書かなければならない．そもそも部分集合 $A \subset \{0,1\}^L$ の総数は 2^{2^L} だから，そのようなプログラムも同じ数だけ必要である．このときプログラムの長さは長いものでは 2^L ビットに達することが § 1.3 で述べた議論と同様にして分かる．さらに，大部分の A のプログラムがほぼ 2^L ビットの長さに達することも分かる．$L = 2^{26}$ の場合だと，大部分の $A \subset \{0,1\}^{2^{26}}$ のプログラムの長さはほぼ $2^{2^{26}}$ ビットである．

明らかに，そのように長いプログラムが必要となる A に対しては $\omega \in A$ であるかどうか実際には判定できない．そこで，$\omega \in A$ が計算量的に判定可能であるような A に対してのみ安全であるような疑似乱数生成器があればそれで十分である．そのような疑似乱数生成器は**計算量的に安全** (computationally secure，または**暗号理論的に安全** (cryptographically secure)) である，といわれる．

暗号理論において，疑似乱数生成器の計算量的安全性は，ここに述べた空間計算量 (プログラムの長さ) ではなく，時間計算量 (プログラムの実行時間) をもとに定式化されている (§ 3.1). [11]

例 5 の場合に g が計算量的に安全であれば，$S_{2^{19}}(\omega)$ と $S_{2^{19}}(g(\omega'))$ の分布は十分近い．なぜなら，$S_{2^{19}}$ という関数が実際に計算できる以上，たとえば各 $c_1, c_2 \in \mathbb{Z}$ に対して $A(c_1, c_2) := \{\omega;\, c_1 \leq S_{2^{19}}(\omega) \leq c_2\}$ とすれば，$\omega \in A(c_1, c_2)$ は計算量的に判定可能だからである．従っ

[11] 短いプログラムでも反復計算が多ければ実行時間が実行不可能なほど莫大になり得るから，時間計算量を基準にすることは実際的な意味がある．

て p' の値が何であっても

$$P_{2^{26}}\left(\left|\frac{S_{2^{19}}(\omega)}{2^{19}} - p'\right| \geq 2^{-8}\right) \approx P_{2^8}\left(\left|\frac{S_{2^{19}}(g(\omega'))}{2^{19}} - p'\right| \geq 2^{-8}\right)$$

が成り立つ.

計算量的に安全な疑似乱数生成器は，理論上，最も汎用的で最も完全な疑似乱数生成器といえよう．ただし，その存在は計算量理論の難しい問題の一つであり現時点では不明である (§ 3.1.3)．また，この概念は正確に述べれば漸近的性質であって，個々の具体的な問題に対して，計算量的に安全な疑似乱数生成器が確実に有用であることを保証するものではない.

1.5 モンテカルロ積分

モンテカルロ積分 (Monte-Carlo integration) とは，例 3 のように大数の法則を用いて確率変数の平均を推定する方法をいう．およそ科学的なモンテカルロ法では，確率変数の分布に関する何らかの特性量を計算することを目的とする場合がほとんどであり，それらはすべてモンテカルロ積分であるといって過言でない.

1.5.1 i.id.-サンプリング

例 3 を一般的な設定の下で述べると以下のようになる: X を m 回の硬貨投げの関数，すなわち $\{0,1\}^m$ 上の関数とし，X の平均 $\mathbf{E}[X]$ を求めることを目標としよう．X の独立な N 個のコピー $\{X_n\}_{n=1}^N$ の和を S_N とする．S_N は Nm 回の硬貨投げの関数であるが，具体的に書けば,

$$\begin{aligned} X_N(\omega) &:= X(\omega_n), \quad \omega_n \in \{0,1\}^m, \quad \omega = (\omega_1, \ldots, \omega_N) \in \{0,1\}^{Nm}, \\ S_N(\omega) &:= \sum_{n=1}^{N} X_n(\omega). \end{aligned}$$

このとき S_N/N と X の平均 (それぞれ P_{Nm} と P_m による積分) は等しく ($\mathbf{E}[S_N/N] = \mathbf{E}[X]$), 分散は $\mathbf{Var}[S_N/N] = \mathbf{Var}[X]/N$ を満す．S_N/N でもって X の平均を推定する方法を **i.i.d.-サンプリング**[12] と呼ぶ．このときのリスクをチェビシェフの不等式

$$P_{Nm}\left(\left|\frac{S_N}{N} - \mathbf{E}[X]\right| \geq \delta\right) \leq \frac{\mathbf{Var}[X]}{N\delta^2}$$

で評価しよう.[13] このことは,

$$A_1 := \left\{\omega \in \{0,1\}^{Nm}\,;\, \left|\frac{S_N(\omega)}{N} - \mathbf{E}[X]\right| \geq \delta\right\} \tag{4}$$

としたとき，$\omega \in \{0,1\}^{Nm}$ を選んで $\omega \notin A_1$ ならば勝ち，$\omega \in A_1$ ならば負け，という賭けを考えていることになる.

[12] $\{X_n\}_{n=1}^N$ が i.i.d.(独立同分布) 確率変数列なのでそう呼ぶ.

[13] $\mathbf{E}[X]$ が未知であるのと同様に $\mathbf{Var}[X]$ も未知であるのが普通だろう．その意味でこのリスク評価は完全ではない．しかし状況によっては $\mathbf{Var}[X]$ の上からの評価が得られれば (たとえば 例 3 のように，X が有界の場合など)，このリスク評価は完全になる.

1.5.2 ペアごとに独立な確率変数によるサンプリング

$\{0,1\}^m$ 上のモンテカルロ積分の場合に限れば，それ専用の安全な疑似乱数生成器は存在する．さらに暗号理論ではその具体的な構成方法も次の定理の証明にあるように以前[14]から知られていた．

定理 1 ([14] Lecture 5, cf. [10]) $N \leq 2^m$ のとき，[15] 次の性質を満たす疑似乱数生成器 $g : \{0,1\}^{2m} \to \{0,1\}^{Nm}$ が存在する: ω'，ω はそれぞれ P_{2m}，P_{Nm} に従うものとして

$$\begin{aligned} \mathbf{E}[S_N(g(\omega'))] &= \mathbf{E}[S_N(\omega)] \,(= N\mathbf{E}[X]), \\ \mathbf{Var}[S_N(g(\omega'))] &= \mathbf{Var}[S_N(\omega)] \,(= N\mathbf{Var}[X]). \end{aligned}$$

証明. g を次のように構成する: $\mathrm{GF}(2^m)$ を位数 2^m の有限体とし，任意の 2 つの全単射 $\phi : \mathrm{GF}(2^m) \to \{0,1\}^m$ と $\psi : \mathrm{GF}(2^m) \to \{1,2,3,\ldots,2^m\}$ によって，$\{0,1\}^m$ と $\{1,2,3,\ldots,2^m\}$ を $\mathrm{GF}(2^m)$ と同一視しよう．各 $\omega' := (x,\alpha) \in \mathrm{GF}(2^m) \times \mathrm{GF}(2^m) \cong \{0,1\}^{2m}$ に対して

$$Z_n(\omega') := x + n\alpha, \quad n \in \mathrm{GF}(2^m) \cong \{1,2,3,\ldots,2^m\}$$

とおき，$g : \{0,1\}^{2m} \to \{0,1\}^{Nm}$ を次で定義する．

$$g(\omega') := (Z_1(\omega'), Z_2(\omega'), \ldots, Z_N(\omega')) \in \mathrm{GF}(2^m)^N \cong \{0,1\}^{Nm}$$

このとき，ω' を $\mathrm{GF}(2^m)^2 \cong \{0,1\}^{2m}$ 上の一様分布 P_{2m} に従って選べば，各 $Z_n(\omega')$ は $\{0,1\}^m$ 上で一様分布することは容易に分かる．従って，

$$\mathbf{E}[S_N(g(\omega'))] = \mathbf{E}[S_N(\omega)].$$

さらに $\{Z_n(\omega')\}_{n=1}^{2^m}$ はペアごとに独立になる．実際，任意の $a,b \in \mathrm{GF}(2^m) \cong \{0,1\}^m$，$1 \leq n < n' \leq 2^m$ に対して

$$P_{2m}(Z_n(\omega') = a,\, Z_{n'}(\omega') = b) = P_{2m}\,(x + n\alpha = a,\, x + n'\alpha = b)$$

ここで，未知数 (x,α) に関する $\mathrm{GF}(2^m)$ での連立 1 次方程式

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

の一意解を $(x_0,\alpha_0) \in \mathrm{GF}(2^m) \times \mathrm{GF}(2^m)$ とすれば

$$\begin{aligned} P_{2m}(Z_n(\omega') = a,\, Z_{n'}(\omega') = b) &= P_{2m}\,(\{(x_0',\alpha_0')\}) = 2^{-2m} \\ &= P_{2m}(Z_n(\omega') = a) P_{2m}(Z_{n'}(\omega') = b) \end{aligned}$$

[14] 本質的には [10] で解決されているといってよく，従って $\{0,1\}^m$ 上のモンテカルロ法のための安全な疑似乱数生成器の構成法はすでに 1974 年から存在していたことになる．

[15] $\{0,1\}^m$ で定義された確率変数は，その各点での値 (全部で 2^m 個) をすべて計算すれば平均を確実に求めることができる．従ってモンテカルロ積分では $N < 2^m$ の場合だけ考えればよい．

である．このペアごとの独立性によって

$$\begin{aligned}\mathbf{Var}[S_N(g(\omega'))] &= \mathbf{E}\left[\sum_{n=1}^{N}\left(X(Z_n(\omega'))-\mathbf{E}[X]\right)^2\right]\\ &= \sum_{n=1}^{N}\sum_{n'=1}^{N}\mathbf{E}\left[\left(X(Z_n(\omega'))-\mathbf{E}[X]\right)\left(X(Z_{n'}(\omega'))-\mathbf{E}[X]\right)\right]\\ &= \sum_{n=1}^{N}\mathbf{E}\left[\left(X(Z_n(\omega'))-\mathbf{E}[X]\right)^2\right]\\ &\quad +2\sum_{1\le n<n'\le N}\mathbf{E}\left[\left(X(Z_n(\omega'))-\mathbf{E}[X]\right)\left(X(Z_{n'}(\omega'))-\mathbf{E}[X]\right)\right]\\ &= N\mathbf{Var}[X].\end{aligned}$$

以上から，g は要請された性質を持つことが分かる． □

定理 1 の疑似乱数生成器 g を用いて $S_N(g(\omega'))/N$ によって $\mathbf{E}[X]$ を推定するときのリスクは，やはりチェビシェフの不等式によって (4) の集合 A_1 に対して

$$P_{2m}(g(\omega')\in A_1) \;=\; P_{2m}\left(\left|\frac{S_N(g(\omega'))}{N}-\mathbf{E}[X]\right|\ge\delta\right) \;\le\; \frac{\mathbf{Var}[X]}{N\delta^2}$$

であり，$S_N(\omega)$ の場合と同一のリスク評価を持つことになる．従って g は A_1 に対して安全な疑似乱数生成器であることが分かる．たとえば，定理 1 の g を $g:\{0,1\}^{2^8}\to\{0,1\}^{2^{26}}$ となるように構成して例 5 を実現する場合，リスクが

$$P_{2^8}\left(\left|\frac{S_{2^{19}}(g(\omega'))}{2^{19}}-p\right|\ge 2^{-8}\right) \;\le\; \frac{1}{32}$$

のように評価される (cf.(3))．アリスは自分の意思で種 $\omega'\in\{0,1\}^{2^8}=\{0,1\}^{256}$ を容易に選ぶことができるから，この場合は乱数は必要ない．

注意 1　もっと大規模なモンテカルロ積分で $2m\gg 1$ の場合だと，再び乱数の問題によって，アリスは種 $\omega'\in\{0,1\}^{2m}$ さえ自分の意思で選ぶことができなくなる．その場合は，さらに別の補助的な疑似乱数生成器 $g':\{0,1\}^n\to\{0,1\}^{2m}$，$n\ll 2m$，を用いて $\omega'\in\{0,1\}^{2m}$ を選ぶ破目になる．この場合は，現時点では合成された疑似乱数生成器 $g\circ g':\{0,1\}^n\to\{0,1\}^{2m}\to\{0,1\}^{Nm}$ が A_1 に対して安全になるかどうかは分からない．

注意 2　例 2 で扱った確率変数 X の最小値を探索する場合にペアごとに独立な確率変数によるサンプリングを利用してみよう．$\Pr(X<c)=1/10000$ とし，$X_1,X_2,\ldots,X_{40000}$ を X のペアごとに独立なコピーとする．このとき $S:=\sum_{i=1}^{40000}\mathbf{1}_{\{X_i<c\}}$ とすれば

$$\mathbf{E}[S]=4,\qquad \mathbf{Var}[S]=40000\mathbf{Var}[\mathbf{1}_{\{X_i<c\}}]=40000\left(1-\frac{1}{10000}\right)\frac{1}{10000}\;<\;4.$$

だからチェビシェフの不等式より

$$\Pr(S\ge 1)\;\ge\;\Pr\left(|S-4|<4\right)\;\ge\;1-\frac{4}{4^2}\;=\;\frac{3}{4}.$$

これより $\min_{1\le i\le 40000}X_i$ は少なくとも確率 3/4 以上で c 以下の値をとることが分かる．

注意 3　定理 1 の証明で構成した疑似乱数生成器 g は実際の数値計算では m が少し大きいとプログラムが複雑になり (GF(2^m) における積演算が面倒)，サンプルの生成が非常に遅いので実用的ではない．そこで，ペアごとに独立なサンプルをもっと高速に生成する実用的な方法について § 4.2.1 で紹介する．なお，この § の例題はその実用的方法によって例 9 で具体的に解かれる．

1.6　数理統計学の視点から

1.6.1　無作為なサンプリング

我々はモンテカルロ法を賭けと考え，プレーヤーのアリスが自分の意志で疑似乱数の種 $\omega' \in \{0,1\}^n$ を選ぶ，という観点で論じてきた．しかし数理統計学の視点から見ると，ω' がアリスの意志で選ばれるというのは困ったことである．なぜなら，結果に客観性を持たせるために数理統計学では**無作為なサンプリング**を行うことを重要と考えるからである．実際，定理 1 の疑似乱数生成器 g の場合は，種として $\omega' = (x, \alpha) = (0,0) \in \mathrm{GF}(2^m) \times \mathrm{GF}(2^m)$ を選べば $Z_n(\omega') \equiv 0 \in \mathrm{GF}(2^m)$ となって S の一般的な値を選ぶことはほぼ絶望的になる．つまり，悪い種を選んで賭けにわざと負けることができる．プレーヤーの意思で結果が左右されることが起こり得るのである．

サンプリングの客観性を厳密に論ずることはもちろん数学の守備範囲を超えている．ここでは，たとえば，本物の硬貨の表裏の出方が誰の意思にも影響されないことを仮定した上で議論することにしよう．このとき，たとえば例 5 の ω' を選ぶときは，硬貨を硬貨を 256 回投げて，順に，表だったら 1，裏だったら 0，を記録していく．そうして長さ 256 の $\{0,1\}$-列ができたら，それを疑似乱数生成器 g の種 ω' として $S(g(\omega'))$ を計算する．これで，サンプリングが無作為に実行される．

この方法で非常に長い $\omega \in \{0,1\}^L$ を無作為に選ぶことはあまりに膨大な時間と労力がかかるので事実上不可能である．重要なのは，疑似乱数生成器の利用によって無作為に選ばなければならない $\{0,1\}$-列の長さがきわめて短くなるため，この方法が実際に実行可能になる，ということである．

注意 4　JIS[8] p.7 には乱数表の使い方について次のような記述がある．

> **出発点をランダムにきめる** 付表 1(乱数表) の任意のページの上に目をつぶって鉛筆を立てて落とし，当たった点に一番近い数字を起点として連続 3 個の数字を読み，その数字を 250 で割った余りに 1 を加えた数を行の番号とする．次にもう一度鉛筆を落として当たった点に一番近い数字を起点として連続 2 個の数字を読み，その数字を 20 で割った余りを列の番号とする．
> **乱数列を読み取る** 10 進 1 桁の乱数列または 2 桁の乱数列が必要な場合は右に進む．右端に達したら次の行の左端に移る．(以下略)

ここで，目をつぶって鉛筆を落とし乱数表の数字を読み始める位置を決めるのは，ちょうど，硬貨を投げて種 ω' を決めることに対応する．読み始める位置を決めてから決まった手順で次々と乱数を拾い出していくさまは，種 ω' から疑似乱数生成器が決まったアルゴリズムで疑似乱数を生成することに対応している．

1.6.2 疑似乱数の検定

[8] をはじめ，夥しい数の疑似乱数の検定はおよそ次の手順で行われてきた:

1. 検定項目 (連の検定，ポーカー検定など) を決める．
2. 複数の種 ω' をもとに疑似乱数生成器 g によって疑似乱数 $g(\omega')$ を生成し，その結果，棄却されるものの割合を計算する．
3. 棄却されるものの割り合いがその検定の危険率程度であれば，g を採択し，それを大きく超えるようであれば棄却する．

上の手順において，1. で選ばれた検定の棄却域を A とすれば，2. で行っていることは確率 $P_n(g(\omega') \in A)$ を推定する作業と解釈できる．そして 3. ではそれが危険率 $P_L(\omega \in A)$ と近いかどうかを調べていることになる．従ってこうした検定作業は，じつは g の集合 A に対する安全性の検査であったといえるだろう．

1.7 この§のまとめとして...

ここまで読んで，読者はモンテカルロ法の目的とあり方，乱数，および疑似乱数の定義を理解されたことと思う．そこでもう一度，§ 1.1 を読んで全体の流れを確認されることをお勧めする．

2　乱数

乱数とは，素朴には，何ら規則性のない $\{0,1\}$-列をいう．与えられた有限 $\{0,1\}$-列を出力するコンピュータプログラム (これも 0 と 1 の並びで書かれているとする) を考える．短いプログラムで生成される $\{0,1\}$-列は規則的，逆に長いプログラムが必要な $\{0,1\}$-列は不規則的であると解釈されよう．そこで，大変長いプログラムが必要な $\{0,1\}$-列を乱数と呼べばよい．もっとも，プログラムの長さはプログラム言語 (C，Pascal，...) に依存するので，これを普遍的に定義する必要がある．

このように，乱数の定義は確率と無関係のところで述べられる．しかしながら，これが硬貨投げの確率過程に対する検定と結びつくことを示したのはマルティン=レーフ (Martin-Löf) である．その際，マルティン=レーフは万能検定と呼ばれる，ある普遍的な検定の存在を示し，乱数であることと万能検定に採択されることの同値性を述べた (定理 7).

2.1　計算の複雑さ

2.1.1　帰納的部分関数と枚挙定理

コンピュータプログラムを数学的に表現するために，帰納的部分関数という概念が用いられる．これは，コンピュータで計算することのできると考えられるすべての関数を含んでいる．

定義 3　(**帰納的部分関数**，partial recursive function，[11, 27])[16]

1. (基本関数)

$$\begin{aligned} \mathrm{zero} &: \ \mathbb{N}^0 \to \mathbb{N}, \quad \mathrm{zero}(\,) := 0 \\ \mathrm{suc} &: \ \mathbb{N} \to \mathbb{N}, \quad \mathrm{suc}(x) := x+1 \\ p_i^n &: \ \mathbb{N}^n \to \mathbb{N}, \quad p_i^n(x_1,\ldots,x_n) := x_i, \qquad i = 1,\ldots,n. \end{aligned}$$

   は帰納的部分関数.

2. (合成)
   $g : \mathbb{N}^m \to \mathbb{N}$，$g_j : \mathbb{N}^n \to \mathbb{N}$ が帰納的部分関数ならば,

$$f : \mathbb{N}^n \to \mathbb{N}, \qquad f(x_1,\ldots,x_n) := g(g_1(x_1,\ldots,x_n),\ldots,g_m(x_1,\ldots,x_n))$$

   は 帰納的部分関数.

3. (帰納法)
   $g : \mathbb{N}^n \to \mathbb{N}$，$h : \mathbb{N}^{n+2} \to \mathbb{N}$ が帰納的部分関数ならば，$f : \mathbb{N}^{n+1} \to \mathbb{N}$,

$$\begin{cases} f(x_1,\ldots,x_n,0) &:= \ g(x_1,\ldots,x_n) \\ f(x_1,\ldots,x_n,y+1) &:= \ h(x_1,\ldots,x_n,y,f(x_1,\ldots,x_n,y)) \end{cases}$$

   は帰納的部分関数.

[16]部分関数とは $\mathbb{N}^n$ のある部分集合で定義された $\mathbb{N}^m$-値関数のことをいう．面倒なので定義域を明示せず，$g : \mathbb{N}^n \to \mathbb{N}^m$ のように書く.

4. (直積)
   $g_j : \mathbb{N}^{n_j} \to \mathbb{N}$, 帰納的部分関数, $j = 1, \ldots, m$, のとき,

$$g : \mathbb{N}^{n_1+\cdots+n_m} \to \mathbb{N}^m, \qquad g := (g_1, \ldots, g_m)$$

   は帰納的部分関数.

5. (最小解関数)
   $p : \mathbb{N}^{n+1} \to \mathbb{N}$ が帰納的部分関数のとき,

$$\mu_y(p(x_1, \ldots, x_n, y)) := \min\{y \in \mathbb{N} \mid p(x_1, \ldots, x_n, y) = 1\}$$

   ただし, $\min \emptyset$ は定義しない.

6. 以上, 1〜5 の有限個の組み合わせで得られる関数 ($\mathbb{N}^m \to \mathbb{N}^n$), およびそれらだけが帰納的部分関数.

1〜4 の有限個の組み合わせで得られる関数は**原始帰納的関数** (primitive recursive function) と呼ばれる. 原始帰納的関数は最小解関数を用いないので全域で定義されている. また, 最小解関数を使っていても, 条件 $\{y \in \mathbb{N} \mid p(x_1, \ldots, x_n, y) = 1\} \neq \emptyset$ が満たされるときに限って $\mu_y(p(x_1, \ldots, x_n, y))$ を用いるとき, 帰納的部分関数は全域で定義されるが, これを**帰納的関数** (recursive function) という.

定理 2 (枚挙定理) $l \in \mathbb{N}$ とする. ある帰納的部分関数 $B : \mathbb{N} \times \mathbb{N}^l \to \mathbb{N}$ が存在して, 任意の帰納的部分関数 $A : \mathbb{N}^l \to \mathbb{N}$ に対して次を満たすような $e_A \in \mathbb{N}$ が存在する:

$$B(e_A, x) = A(x), \qquad x \in \mathbb{N}^l.$$

定理 2 における B は**枚挙関数** (enumerating function), e_A は A の**ゲーデル数** (Gödel number) と呼ばれる. この定理の詳しい証明は計算可能性に関する教科書 (たとえば [11, 27]) に譲るとして, ここでは証明のアイデアを述べる. まず, 与えられた帰納的部分関数 A がどのようにして基本関数から作られたのかを具体的に符号化し, それに一つの自然数を対応させる. たとえば万能チューリング機械 (universal Turing machine, 無限の記憶容量を持つ仮想的なコンピュータ, 詳しくは [11] を見よ) のプログラムを $\{0, 1\}$-列として書き, さらにそれを 2 進数自然数と見ればよい. ゲーデル数 e_A はそのような自然数なのである. 枚挙関数 $B(e, x)$ は, まず, e が帰納的部分関数のゲーデル数かどうかを判定し, もし, そうであるなら, その e の表す帰納的部分関数を基本関数から再現して, それに x を代入してその答えを返すような関数である.

注意 5 定理 2 において, "帰納的部分関数" を "帰納的関数" で置き換えることはできない. 確かに, 帰納的関数の全体も可算集合であるが, それらすべてを一つの帰納的関数で枚挙することはできないのである. もし, できたとすると矛盾が生じることを示そう. 簡単のため, $B : \mathbb{N} \times \mathbb{N} \to \mathbb{N}$ をすべての 1 変数の帰納的関数を枚挙する 2 変数の帰納的関数とする. このとき, $h(x) := B(x, x) + 1$ は 1 変数の帰納的関数であるから, ある $e_h \in \mathbb{N}$ によって $h(x) = B(e_h, x)$ と書かれる. これに $x = e_h$ を代入すれば $B(e_h, e_h) + 1 = h(e_h) = B(e_h, e_h)$ となって矛盾である (対角線論法)[17].

[17]可算集合であってもその元を具体的に数え上げるアルゴリズムが存在しない, という例である. これは

2.1.2 コルモゴロフの計算の複雑さと乱数

定義 4

1. $\{0,1\}^* := \bigcup_{n\in\mathbb{N}\cup\{0\}}\{0,1\}^n$. すなわち $\{0,1\}^*$ は長さが有限の $\{0,1\}$-列全体である．とくに長さ 0 の $\{0,1\}$-列も考えてこれを空語と呼ぶ.

2. $p \in \{0,1\}^*$ に対して，$p \in \{0,1\}^n$ となる n を $L(p) \in \mathbb{N}\cup\{0\}$ と書く (p の長さ).

3. $\{0,1\}^*$ に標準的順序 (canonical order) を入れて $\mathbb{N}$ と同一視する: $x, y \in \{0,1\}^*$ において，x の方が y より長い $\{0,1\}$-列のとき，$x \geq y$ とする．同じ長さのときは 2 進整数と見てその大きさで順序を決める.

定義 5 (アルゴリズムに依存した計算の複雑さ) $A : \{0,1\}^*\times\{0,1\}^* \to \{0,1\}^*$ は $\mathbb{N}\times\mathbb{N} \to \mathbb{N}$ と見て帰納的部分関数とする．A をアルゴリズムと呼ぶ．アルゴリズム A のもとで $x \in \{0,1\}^*$ を入力としたときの $y \in \{0,1\}^*$ の複雑さを

$$K_A(y|x) := \min\{L(p) \mid p \in \{0,1\}^*,\, A(p,x) = y\}$$

と定義する．ただし，$\min\emptyset = \infty$ とする.[18]

定義 5 において，A は今日ではプログラム言語と呼んだ方が分かりやすいだろう．A の入力の一つ p はプログラムであり，A は p を解釈しもう一つの入力 x から y を計算して出力する．従って，$K_A(y|x)$ はプログラム言語 A の下で入力 x から出力 y を得るためのプログラム p のうち，最も短いものの長さを返す関数である.

当然，K_A は A に依存するので，このままでは客観的な複雑さの指標にはならない．そこで次の定理が登場する.

定理 3 ある帰納的部分関数 $A_0 : \{0,1\}^*\times\{0,1\}^* \to \{0,1\}^*$ が存在して，任意の帰納的部分関数 $A : \{0,1\}^*\times\{0,1\}^* \to \{0,1\}^*$ に対して以下を満たす定数 $c_{A_0A} \in \mathbb{N}$ が存在する:

$$K_{A_0}(y|x) \leq K_A(y|x) + c_{A_0A}, \qquad \forall x, y \in \{0,1\}^*.$$

A_0 は**万能アルゴリズム** (universal algorithm，あるいは**漸近最適アルゴリズム** (asymptotically optimal algoruthm)) と呼ばれる.

証明. $e, p \in \{0,1\}^*$ に対して $\bar{e}\cdot p \in \{0,1\}^*$ を次で定義する: $e = (e_1, \ldots, e_m)$, $p = (p_1, \ldots, p_l)$ のとき,

$$\bar{e}\cdot p := (e_1, e_1, \ldots, e_m, e_m, 0, 1, p_1, \ldots, p_l) \in \{0,1\}^{2m+2+l} \tag{5}$$

B を定理 2 の枚挙関数 ($l = 2$) とし，アルゴリズム A_0 を次で定義する.

$$A_0(\bar{e}\cdot p, x) := B(e, p, x), \qquad e, p, x \in \{0,1\}^*.$$

「真であっても証明できない命題が存在する」というゲーデルの不完全性定理と本質的に同じ精神に基づいている.

[18] 無定義としてもよいが，後述の不等式を考慮して，ここでは $\min\emptyset = \infty$ としておく.

もし，$z = \bar{e}\cdot p$ の形をしていなかったら $A(z,x)$ は定義しない．このとき，$A_0(\bar{e}_A\cdot p, x) = A(p,x)$ であるから，

$$K_{A_0}(y|x) \le K_A(y|x) + 2L(e_A) + 2, \qquad \forall x,y \in \{0,1\}^*.$$

そこで $c_{A_0A} := 2L(e_A) + 2$ として定理の主張が従う． □

注意 6　万能アルゴリズムは一意的でないが，A_0' をもう一つの万能アルゴリズムとするとき，$c > 0$ が存在して

$$\left|K_{A_0}(y|x) - K_{A_0'}(y|x)\right| < c, \qquad \forall x,y \in \{0,1\}^*. \tag{6}$$

定義 6　万能アルゴリズム A_0 を一つ固定し，

$$K(y|x) := K_{A_0}(y|x), \qquad x,y \in \{0,1\}^*$$

を入力 x を与えたときの y の複雑さと呼ぶ．とくに，x が空語のとき，$K(y)$ と書いて y の**コルモゴロフの複雑さ** (Kolmogorov's complexity) と呼ぶ．

定義 6 は A_0 の取り方に依存するが，(6) があるので，十分大きな複雑さを扱うときは定数 c が，事実上，問題でなくなる．

定理 4　(i) ある定数 $c > 0$ が存在して，任意の $x \in \{0,1\}^n$ に対して，$K(x|n) \le n + c$.
(ii) $n > c' > 0$ のとき，[19] $\#\{x \in \{0,1\}^n \mid K(x|n) \ge n - c'\} \ge 2^n - 2^{n-c'}$.

証明. (i) x をそのまま書き出すアルゴリズム $A(x,n) := x$ を考えれば，$K_A(x|n) = n$ だから．
(ii) 長さが $n - c'$ 未満のプログラム p の個数は高々$2^{n-c'}$ 個であるから．[20] □

定理 4 の主張は $K(x)$ についてもまったく同様に成り立つ．従って，定数 c が無視できるほど n が大きいとき，大多数の $x \in \{0,1\}^n$ の複雑さ $K(x)$ はほぼ n である．このように $K(x)$ がほぼ最大の n となる $x \in \{0,1\}^n$ を**乱数** (random number) と呼ぶ．

例 6　円周率 π の公式計算記録は，2007 年 2 月現在，2 進小数で 4 兆 1228 億桁 (10 進小数で約 1 兆 2411 億桁) である．それを算出したプログラムは，4 兆 1228 億ビットよりずっと短いので，π の 2 進小数表示における 0 と 1 の並びは乱数ではない．

例 6 のように，乱数でないことが具体的に分かる $x \in \{0,1\}^*$ は存在するが，乱数であることが具体的に分かる $x \in \{0,1\}^*$ の例は不明である．実際，次の定理 5 は，$x \in \{0,1\}^*$ が与えられたとき，それが乱数であるかどうか判定するアルゴリズムは存在しないことを示す．

定理 5　$K(x)$ は帰納的関数でない．

[19] # は集合の元の個数を表す．
[20] § 1.3 で調べた個数の議論そのものである．

証明. 背理法で示す．$K(x)$ は帰納的関数であると仮定する．このとき，次のプログラム p_n について考察する．

function p_n : $\{0,1\}^*$-値関数;
begin
　$l := 1$ とし，以下，l を 1 ずつ増加させて繰り返し，
　　すべての $x \in \{0,1\}^l$ に関する繰り返し，
　　もし $K(x) \geq n$ ならば x を返す.
end;

p_n は短い $\{0,1\}$-列から始めてすべての $\{0,1\}$-列 x に対し $K(x)$ を計算して，それが n 以上となったときにその x を出力して中断する． このプログラム p_n と $K(x)$ のサブルーチンとを合わせてたプログラムの長さを L_n とする．L_n は n に依存するが，n の記述長は対数オーダーでしか大きくならないので，$L_n = O(\log n)$，従って $2L_n < n$ となる n を取ることができる．しかしこのとき、このプログラム自身が $K(x)$ よりも短い x を生成するプログラムとなってしまう．実際，$K(x) \geq n > 2L_n$．これは K の定義と矛盾する。 □

2.2 検定とマルティン=レーフの定理

2.2.1 検定の定式化と万能検定

定義 7 $\mathbb{N} \times \{0,1\}^* \supset U$ が**検定**とは,
(i) U は**帰納的可算集合** (recursively enumerable set)，すなわち，ある帰納的部分関数または原始帰納的関数 $\phi : \mathbb{N} \to \mathbb{N} \times \{0,1\}^*$ が存在し[21]，$U = \phi(\mathbb{N})$.
(ii) $U_m := \{x \in \{0,1\}^* \mid (m,x) \in U\}$ とするとき，$U_m \supset U_{m+1}$，$m \in \mathbb{N}$.
(iii) $\#(U_m \cap \{0,1\}^n) \leq 2^{n-m}$，$n > m > 0$.
が成り立つこと.

ここで U_m は危険率 2^{-m} 以下の検定の棄却域と見なされる．また，検定 U に対して関数

$$m_U(x) := \max\{m \in \mathbb{N} \mid x \in U_m\} \tag{7}$$

を定める．$m_U(x)$ が小さいほど，x は検定 U で採択されやすい．つまり，U ではランダムと判定されやすい.

定理 6 ([15]) 以下の性質を満たす検定 V (**万能検定**，universal test) が存在する: 任意の検定 U に対して，ある $c = c_{VU} \in \mathbb{N}$ が存在し，

$$U_{m+c} \subset V_m, \qquad m \in \mathbb{N}. \tag{8}$$

すなわち，U で危険率 2^{-m-c} で棄却される $x \in \{0,1\}^*$ は V で危険率 2^{-m} で棄却される.

[21] ϕ は帰納的部分関数または原始帰納的関数としても同じことであることが知られている ([27] 定理 1.7.4). ϕ が帰納的部分関数の場合は，$\phi(\mathbb{N})$ は ϕ が定義されている $\mathbb{N}$ の部分集合の像の意味である.

証明. 検定 U を定義する帰納的部分関数を ϕ_U とする．枚挙定理によって $\{\phi_U\}_{U:\text{検定}}$ を数え上げる帰納的部分関数 $\psi : \mathbb{N}\times\mathbb{N} \to \mathbb{N}\times\{0,1\}^*$ が存在する．すなわち，任意の検定 U に対して，$e_U \in \mathbb{N}$ が存在して $\psi(e_U, \bullet) = \phi_U(\bullet)$ となる．帰納的部分関数 $h : \mathbb{N} \to \mathbb{N}\times\mathbb{N}\times\{0,1\}^*$ を $h(\overline{e}\cdot n) := (e, \psi(e,n))$ で定義する．ただし，入力が $\overline{e}\cdot n$ の形 ((5) を見よ) をしていなければ無定義とする．そこで，$\widetilde{V} := h(\mathbb{N})$ とおけば，$\widetilde{V}$ は再び帰納的可算集合である．このとき，検定 U に対して

$$\widetilde{V}_{e_U} := \{(m,x) \,|\, (e_U, m, x) \in \widetilde{V}\} = \psi(e_U, \mathbb{N}) = U.$$

そこで，帰納的部分関数 $\tau : \mathbb{N}\times\mathbb{N}\times\{0,1\}^* \to \mathbb{N}\times\{0,1\}^*$ を $\tau(e, m+e, x) := (m,x)$ で定義して (入力 (e,m',x) が $m' \le e$ のときは $\tau(e,m',x)$ は定義されない)， $V := \tau(\widetilde{V})$ とすれば，これも帰納的可算集合であり，この V が求める検定である．実際，$V_m = \cup_{e=1}^{\infty}(\widetilde{V}_e)_{m+e}$ であるから，$V_m \supset V_{m+1}$ は明らか,[22] さらに,

$$\begin{aligned}\#\{x \in V_m \,|\, L(x) = n\} &\le \sum_{e=1}^{\infty} \#\{x \in (\widetilde{V}_e)_{m+e} | L(x) = n\} \\ &\le \sum_{e=1}^{n-m-1} 2^{n-(m+e)} \le 2^{n-m}\end{aligned}$$

だから，V は検定であって，任意の検定 U に対して $U_{m+e_U} \subset V_m$ が成り立つ． □

(8) を (7) を使って書けば,

$$m_U(x) \le m_V(x) + c \tag{9}$$

である．とくに，V と V' を万能検定とすると，それらだけに依存した定数 $c \in \mathbb{N}$ が存在して,

$$|m_V(x) - m_{V'}(x)| < c, \qquad x \in \{0,1\}^*,$$

が成り立つ．つまり，両者は高々定数差しかない．そこで，一つの万能検定 V を固定し，$m_V(x)$ を単に $m(x)$ と書こう．

2.2.2 マルティン=レーフの定理

次の定理は $K(x|L(x))$ が $L(x)$ に近ければ，それだけランダムに見える，ということを巧みに述べた定理である．

定理 7 ([15]) ある定数 $c \in \mathbb{N}$ が存在して

$$|L(x) - K(x|L(x)) - m(x)| \le c, \qquad \forall x \in \{0,1\}^*. \tag{10}$$

証明. (1) $L(x) - K(x|L(x)) \le m(x) + c$ の証明 :

$$U := \{(m,x) \,|\, K(x|L(x)) < L(x) - m\}$$

[22] e がどの e_U とも等しくないときは $\widetilde{V}_e = \emptyset$.

は定理 4(ii) により検定であることが分かる．明らかに,

$$m_U(x) \;=\; L(x) - K(x|L(x)) - 1$$

だから (9) によって，$m(x) + c > L(x) - K(x|L(x))$ となる．

(2) $K(x|L(x)) \le L(x) - m(x) + c$ の証明 : V を枚挙する原始帰納的関数を ϕ とする．すなわち，$\phi(\mathbb{N}) = V$．この ϕ を用いて，アルゴリズム $A : \{0,1\}^* \times \{0,1\}^* \to \{0,1\}^*$ を以下で定義する．

$\phi(i) =: (m_i, x_i) \in V$ と書く．まず,

$$A(\underbrace{0\ldots00}_{L(x_1)-m_1}, L(x_1)) \;=: x_1$$

と定める．次にもし，$(m_1, L(x_1)) = (m_2, L(x_2))$ ならば,

$$A(\underbrace{0\ldots01}_{L(x_2)-m_2}, L(x_2)) \;=: x_2 \tag{11}$$

とし，もし $(m_1, L(x_1)) \neq (m_2, L(x_2))$ ならば,

$$A(\underbrace{0\ldots00}_{L(x_2)-m_2}, L(x_2)) \;=: x_2$$

とする．以下同様にして A を定義する．V が検定であることから，$(m, L(x))$ が同じであるような (m, x) は高々 $2^{L(x)-m}$ 個しかないことが分かるから，それらは長さ $L(x) - m$ のプログラムで記述できる．従って，A は確かに定義 (well-defined) される．

明らかに

$$K_A(x|L(x)) \;=\; L(x) - m(x),$$

従って，(10) が成り立つ． □

(10) は,「c が無視できるくらい $L(x)$ が大きいとき，$K(x|L(x))$ と $L(x)$ が近いことと，$m(x)$ が小さいことが，同等になる」と読む．すなわち $K(x|L(x)) \approx L(x)$ ならば万能検定を通してみてランダムに見える，ということを意味する．

2.3 無限乱数列

有限列の場合と異なって，無限列に対する乱数の定義は明解である．硬貨投げの確率過程に関する確率 0 の計算可能 (computable) な事象 (たとえば大数の強法則の例外集合) に属するような $\{0,1\}$-列は乱数ではない，と考えるのは自然である．ところが確率 0 の事象を規定するような計算可能な手続きは可算個しかない．だから，その各々で規定される確率 0 の事象の和集合 N は再び確率 0 である．N を**最大帰納的零集合** (maximal recursive null set)，そして N の補集合の元を**無限乱数** (random sequence) と呼ぶ.

ここで，確率 0 の事象を規定するような計算可能な手続きとは何か，述べよう． 各 $y \in \{0,1\}^*$ に対して $C(y)$ を y で始まるような無限 $\{0,1\}$-列の全体 (筒集合) とし，P を公平

な硬貨投げの確率過程の分布 ($\{0,1\}^\infty$ 上の確率測度) とする．事象 $A \subset \{0,1\}^\infty$ が確率 0 であるための必要十分条件は，任意の $m \in \mathbb{N}$ に対して，可算個の有限列 $y_k^{(m)} \in \{0,1\}^*$ が存在して，

$$U_m = \bigcup_k C(y_k^{(m)}) \supset A \tag{12}$$

$$\sum_k P\left(C(y_k^{(m)})\right) = \sum_k 2^{-L(y_k^{(m)})} < 2^{-m} \tag{13}$$

が成り立つことである．

$$U = \{(m,x) \in \mathbb{N} \times \{0,1\}^* \mid x \in U_m\} \tag{14}$$

とおく．このとき，次のことを仮定しても一般性を失わない．

$$(m,x) \in U, \quad n \le m \quad \text{かつ} \quad C(y) \subset C(x) \qquad \Longrightarrow \qquad (n,y) \in U \tag{15}$$

そこで，A が帰納的零集合であるとは，条件 (12)~(15) を満たす U が帰納的可算集合であること，とする．上の U_m は危険率が 2^{-m} の無限列に対する検定の棄却域と見ることができるから，U を**列検定** (sequencial test) と呼ぶ．

前節と同様に，枚挙定理によって次の性質を満たす列検定 U が存在する：任意の列検定 V に対して，

$$V_{m+c} \subset U_m, \qquad m = 1, 2, \ldots \tag{16}$$

が成り立つ．ただし $c > 0$ は U と V にのみ依存する定数である．この U を**万能列検定** (universal sequencial test) という ([15])．万能列検定は，他の列検定と同様に，帰納的零集合を定めるが，それが最大帰納的零集合 N に他ならない．万能列検定は一意的ではないものの，N は一意的に定まる．

例 7　円周率 π という実数を 2 進小数で表したとき，それは無限乱数列か，という問には，否と答えなければならない．実際，桁数を指定したとき，その桁数まで円周率を計算するアルゴリズムが存在するからである．すなわち，そのアルゴリズムを利用して π だけからなる一点集合が帰納的零集合であることを示すことができる．

3　疑似乱数生成器

3.1　計算量的に安全な疑似乱数生成器

3.1.1　定義

疑似乱数の時間計算量 (time complexity) による定義を述べよう．基本的なアイデアは [3, 31] に見ることができる．詳しくは [14, 17] などを見よ．

定義 8

1. この § で現れる関数は帰納的部分関数 $f : \{0,1\}^* \to \{0,1\}^*$ で主として次の形をしている: 各 $n \in \mathbb{N}$ に対して $f|_{\{0,1\}^{r(n)}} =: f_n : \{0,1\}^{r(n)} \to \{0,1\}^{s(n)}$．このとき $f = \{f_n\}_n$ と書く．f を計算するチューリング機械が $f(x)$ を計算するために要するステップ数 (最小単位の操作が何回必要かということ) の $x \in \{0,1\}^{r(n)}$ を動かしたときの最大値を f_n の時間計算量といい，$T_f(n)$ で表す．

2. 自然数列 $\{\ell(n)\}_n$ が多項式パラメータであるとは，ある定数 $c > 0$ が存在して $n \to \infty$ のとき $\ell(n) = O(n^c)$ であることをいう．

3. $f = \{f_n\}_n$ が多項式時間の関数であるとは，多項式パラメータ $r(n)$, $s(n)$ が存在して，$f_n : \{0,1\}^{r(n)} \to \{0,1\}^{s(n)}$ であり，さらに f_n の時間計算量 $T_f(n)$ が多項式パラメータであることをいう．この定義は多変数関数に対しても同様に考えることが出来る．

4. $Y \in_U \{0,1\}^{s(n)}$ は Y が $\{0,1\}^{s(n)}$ 上の一様分布に従う確率変数であることを意味する．Y は他のすべての確率変数と独立であると仮定する．Y を支配する確率測度を明示するときは $\Pr_Y$ と書く．

5. $A = \{A_n\}_n$ がランダムな関数であるとは，$A_n : \{0,1\}^{r(n)} \times \{0,1\}^{s(n)} \to \{0,1\}^{t(n)}$ であって，入力 $x \in \{0,1\}^{r(n)}$ と確率変数 $Y \in_U \{0,1\}^{s(n)}$ によってランダムな出力 $A_n(x, Y)$ を返す関数である．時間計算量 $T_A(n)$ は $A_n(x, y)$ の時間計算量である．しばしば Y を省略し，単に「ランダムな関数 $A_n : \{0,1\}^{r(n)} \to \{0,1\}^{t(n)}$」と書く．この定義は多変数関数に対しても同様に考えることができる．

定義 9　$g_n : \{0,1\}^n \to \{0,1\}^{\ell(n)}$，$\ell(n) > n$，なる多項式時間関数 $g = \{g_n\}_{n \in \mathbb{N}}$ を**疑似乱数生成器** (pseudorandom generator) という．

注意 7　§ 1 の定義 1 では疑似乱数生成器を $g : \{0,1\}^n \to \{0,1\}^L$，$n < L$，なる単独の関数として定義した．特定の問題を解くための疑似乱数生成器を考える際にはそれでよい．しかし，汎用の目的のための疑似乱数生成器は定義 9 にあるように，関数の列として定義するのがよい．すなわち，具体的な問題ごとに適切な n を選んで $g_n : \{0,1\}^n \to \{0,1\}^{\ell(n)}$ を用いることを可能にするためである．

確率変数 $Z_n \in_U \{0,1\}^n$ を考え，これをプレーヤーが選ぶ種と見なす. それから関数 g_n によって $\{0,1\}^{\ell(n)}$-値の確率変数 $g_n(Z_n)$ を作り，これを疑似乱数と考える．もちろん，$\ell(n) > n$ なので $g_n(Z_n)$ は $\{0,1\}^{\ell(n)}$ 上で一様には分布しない.

次に，検定のための関数を考える．$A = \{A_n\}_n$ を $A_n : \{0,1\}^{\ell(n)} \to \{0,1\}$ なる関数 (あるいはランダムな関数) の列とし，

$$\delta_{g,A}(n) := \left|\Pr_{Z_{\ell(n)}}(A_n(Z_{\ell(n)}) = 1) - \Pr_{Z_n}(A_n(g_n(Z_n)) = 1)\right| \tag{17}$$

とおく.[23]疑似乱数生成器としては，$\ell(n)$ が n よりずっと大きいこと，関数 g_n の計算が素早くできること，なおかつ，多くの A に対して $\delta_{g,A}(n)$ が十分小さいこと，が望ましい．しかし，すべての A について $\delta_{g,A}(n)$ の値が小さい必要はない．A の時間計算量を $T_A(n)$ とするとき，

$$S_{g,A}(n) := \frac{T_A(n)}{\delta_{g,A}(n)}$$

とおく.

定義 10　すべての A に対して $S_{g,A}(n)$ が多項式増大を超えるとき，疑似乱数生成器 g は**計算量的に安全**であるという.[24]

$T_A(n)$ が多項式増大を超えれば $S_{g,A}(n)$ も必ず多項式増大を超えるので，定義 10 は実質的にそのような A による検定で棄却されることを許している．従って，計算量的に安全な疑似乱数生成器とは,「多項式時間で生成可能で，多項式時間の検定では硬貨投げと区別のつかないような疑似乱数生成器」を意味する.

計算量理論では，多項式時間の関数を「実際に計算できる関数」，それを超える時間の関数を「実際には計算できない関数」というように大雑把に考えている.[25] だから，計算量的に安全な疑似乱数生成器は，大雑把には「実際に生成可能で，実際に実行可能な検定では硬貨投げと区別のつかないような疑似乱数生成器」を意味する.

3.1.2　計算量的に安全な疑似乱数生成器とモンテカルロ法

疑似乱数は，モンテカルロ法だけではなく，暗号通信においても盛んに用いられている．暗号通信における疑似乱数の使い方は次のとおり : メッセージは，文章，音声，画像など何であれ，デジタル化され結局一つの有限な $\{0,1\}$-列に変換される．メッセージと同じ長さの $\{0,1\}$-値疑似乱数を生成し，メッセージとビットごとに XOR(eXclusive OR，排他的論理和) をとり，それを暗号文とする．復号するには，同じ疑似乱数列を暗号文と再びビットごとに XOR をとればよい．このときの疑似乱数の種が暗号・復号の共通の鍵 (パスワード) である．もし，疑似乱数列が計算量的に安全な疑似乱数生成器によるものだったら，暗号文は鍵を知らない者にとって実際には復号することができなくなる (cf. [14, 17]).

[23]もし A がランダムな関数で，ある確率変数 Y を計算途中で使っているならば，(17) における確率の計算にその Y に関する確率も考慮する.

[24]計算機科学 (暗号理論) では，安全な疑似乱数生成器，あるいは単に，疑似乱数生成器と呼んでいる.

[25]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「多項式時間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.

注意 8　疑似乱数の定義 10 において，関数 A は “敵” を意味する adversary の頭文字である．これは疑似乱数を用いた暗号通信を解読しようとする敵のことであり，ありとあらゆる手段によって攻撃を仕掛けて来る．A としてランダムな関数も許すのは，敵が当て推量で攻撃を仕掛けて来ることも考慮に入れたものである．計算量的に安全な疑似乱数は，あらゆる実行可能な攻撃にも耐え得る暗号を構成するために考え出された概念であり，しばしば**暗号理論的に安全な疑似乱数生成器** (cryptographically secure pseudorandom generator) と呼ばれる．

このように計算量的に安全な疑似乱数生成器はモンテカルロ法とは異なった目的から生れた概念であるが，もちろん，モンテカルロ法においても最も理想的な疑似乱数生成器であるということができる．このことを説明しよう．

まず，数値計算の対象となる実数値確率変数 W を $Z_{\ell(n)} \in_U \{0,1\}^{\ell(n)}$ の関数として表す : $W := \psi(Z_{\ell(n)})$．ここで ψ はコンピュータで実際に計算できる関数であることに注意しよう．さて，一般に $\ell(n)$ は大きすぎるので，$Z_{\ell(n)}$ の代わりに疑似乱数 $g_n(Z_n)$，$Z_n \in_U \{0,1\}^n$，を用いる．つまり，W の代わりに $W' := \psi(g_n(Z_n))$ を数値計算に用いるわけである．そこで問題は W' が果たしてうまく W の代役を無事に勤めることができるかどうか，である．それは W と W' の分布が十分近いかどうか，による．それで，それぞれの分布関数 $F_W(t) := \Pr_{Z_{\ell(n)}}(W \le t)$，$F_{W'}(t) := \Pr_{Z_n}(W' \le t)$ を比較することを考えよう．もし，$g = \{g_n\}_n$ が計算量的に安全な疑似乱数生成器であれば，$F_W(t)$ と $F_{W'}(t)$ は十分近いことが保証される．実際，検定の関数 A_n を $A_n(x) := \mathbf{1}_{\{\psi(x)\le t\}}$，$x \in \{0,1\}^{\ell(n)}$，とおけば，$\psi$ が「実際に計算できる」という事実から，A_n の時間計算量は十分小さいはずである．すると，計算量的に安全な疑似乱数生成器の定義より

$$|F_W(t) - F_{W'}(t)| \;=\; \left|\Pr_{Z_{\ell(n)}}(A_n(Z_{\ell(n)}) = 1) - \Pr_{Z_n}(A_n(g_n(Z_n)) = 1)\right|$$

は十分小さくなければならない．

3.1.3　存在問題

理論的観点からいえば，計算量的に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．このことを正確に述べよう．

二つの計算量のクラス **P** と **NP** の定義を紹介する．

$$\mathbf{P} \;:=\; \left\{ L \subset \{0,1\}^* \;\middle|\; \begin{array}{l} \exists A : \{0,1\}^* \to \{0,1\},\ \text{多項式時間関数, s.t.} \\ \forall x \in \{0,1\}^*\,(x \in L \iff A(x) = 1) \end{array} \right\}$$

$$\mathbf{NP} \;:=\; \left\{ L \subset \{0,1\}^* \;\middle|\; \begin{array}{l} \exists A : \{0,1\}^* \to \{0,1\},\ \text{ランダムな多項式時間関数, s.t.} \\ \forall x \in \{0,1\}^*\,(x \in L \iff \Pr(A(x) = 1) > 0) \end{array} \right\}$$

このとき，$\mathbf{P} \subset \mathbf{NP}$ は明らか．しかし，逆の包含関係は現時点では不明である．多くの研究者は「$\mathbf{P} \neq \mathbf{NP}$」と予想しているが，これは計算量理論において最も重要な予想といわれている．

疑似乱数生成器に関してじつは次の定理がある．

定理 8　**P** = **NP** ならば計算量的に安全な疑似乱数生成器は存在しない. [26]

証明. 多項式パラメータ $\ell(n) > n$ と多項式時間の関数 $g = \{g_n\}_n$，$g_n : \{0,1\}^n \to \{0,1\}^{\ell(n)}$, が任意に与えられたとせよ．$M_n : \{0,1\}^{\ell(n)} \times \{0,1\}^n \to \{0,1\}$ を

$$M_n(y,x) := \begin{cases} 1, & \text{if } g_n(x) = y \\ 0, & \text{if } g_n(x) \neq y \end{cases}$$

とおく．$M = \{M_n\}_n$ は多項式時間の関数である．

$$L := \{y \in \{0,1\}^* \mid \exists n \in \mathbb{N},\, y \in \{0,1\}^{\ell(n)},\, \exists x \in \{0,1\}^n,\, M_n(y,x) = 1\}$$

とすれば $L \in$ **NP**．そこで **P** = **NP** ならば $L \in$ **P**，つまり，多項式時間の関数 $A = \{A_n\}_n$, $A_n : \{0,1\}^{\ell(n)} \to \{0,1\}$ が存在して $y \in L \iff A_n(y) = 1$．この A によれば，$Z_n \in_U \{0,1\}^n$, $Z_{\ell(n)} \in_U \{0,1\}^{\ell(n)}$ として,

$$\Pr_{Z_n}(A_n(g_n(Z_n)) = 1) = 1, \quad \Pr_{Z_{\ell(n)}}(A_n(Z_{\ell(n)}) = 1) \leq \frac{2^n}{2^{\ell(n)}},$$

だから，$\delta_{g,A}(n) \geq 1 - 2^{n-\ell(n)}$ である．いま，$T_A(n)$ は多項式増大だから，$S_{g,A}(n) = T_A(n)/\delta_{g,A}(n)$ も多項式増大である．すなわち，g は計算量的に安全な疑似乱数生成器ではない．　□

P ≠ **NP** の真偽が不明なので，前節の計算量的に安全な疑似乱数生成器の定義は現時点ではその存在が証明できていない．従って，それはそういう性質を持つ疑似乱数生成器の理念を表したものに過ぎない.

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，計算量的に安全な疑似乱数生成器が存在すればすばらしい．もし，計算量的に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，**P≠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，計算量的に安全な疑似乱数生成器は有効である，と.

3.1.4　次ビット予測不可能性

疑似乱数生成器の次ビット予測不可能性と呼ばれるある性質に注目する.

定義 11　$g = \{g_n\}_n$，$g_n : \{0,1\}^n \to \{0,1\}^{\ell(n)}$ を疑似乱数生成器とする．確率変数 $Z \in_U \{0,1\}^n$ と $I \in_U \{1,2,\ldots,\ell(n)\}$ は確率測度 $\Pr_{I,Z}$ の下で独立とする．$\tilde{A} = \{\tilde{A}_n\}_n$ を $\tilde{A}_n : \{1,\ldots,\ell(n)\} \times \{0,1\}^{\ell(n)} \to \{0,1\}$ なる関数 (あるいはランダムな関数) とし,

$$\tilde{\delta}_{g,\tilde{A}}(n) \;=\; \Pr_{I,Z}\left(\tilde{A}_n(I, g_n(Z)_{\{1,\ldots,I-1\}}) = g_n(Z)_I\right) - \frac{1}{2}$$

とする．ここで，$g_n(Z)_i$ は第 i ビット目を表し，$g_n(Z)_{\{1,\ldots,i\}} \in \{0,1\}^{\ell(n)}$ は $g_n(Z)$ の最初の i ビットを出力し，残りのビットを 0 で埋めたものを表す．すなわち,

$$g_n(Z)_{\{1,\ldots,i\}} := (g_n(Z)_1, g_n(Z)_2, \ldots, g_n(Z)_i, \overbrace{0,\ldots,0}^{\ell(n)-i}).$$

[26]なお，**P** ≠ **NP** を仮定しても，計算量的に安全な疑似乱数生成器が存在するかどうか分からない.

このとき，任意の $\tilde{A}$ に対して

$$\tilde{S}_{g,\tilde{A}}(n) := \left| \frac{T_{\tilde{A}}(n)}{\tilde{\delta}_{g,\tilde{A}}(n)} \right|$$

が多項式増大を超えるとき，疑似乱数生成器 g は**次ビット予測不可能**であるという．

定理 9　疑似乱数生成器 $g = \{g_n\}$ が計算量的に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．

証明. (1) g は次ビット予測不可能とする．すなわち，ある $\tilde{A}$ に対して $T_{\tilde{A}}(n)$ と $\tilde{S}_{g,\tilde{A}}(n)$ が多項式パラメータであるとする．まず，$I \in_U \{1, \ldots, \ell(n)\}$ として，ランダムな関数 $A : \{0,1\}^{\ell(n)} \to \{0,1\}$ を

$$A_n(x) := \begin{cases} 1, & \text{if } \tilde{A}_n(I, x_{\{1,\ldots,I-1\}}) = x_I \\ 0, & \text{if } \tilde{A}_n(I, x_{\{1,\ldots,I-1\}}) \neq x_I \end{cases} \qquad x \in \{0,1\}^{\ell(n)},$$

と定める．このとき，

$$\begin{aligned} \delta_{g,A}(n) &= \left| \Pr_{Z_{\ell(n)}}(A_n(Z_{\ell(n)}) = 1) - \Pr_{Z_n}(A_n(g_n(Z_n)) = 1) \right| \\ &= \left| \frac{1}{2} - \Pr_{I,Z_n}\left(\tilde{A}_n(I, g_n(Z_n)_{\{1,\ldots,I-1\}}) = f_n(Z_n)_I \right) \right| \\ &= \left| \tilde{\delta}_{g,\tilde{A}}(n) \right|, \end{aligned}$$

一方，$T_A(n)$ は多項式パラメータだから，$S_{g,A}(n)$ も多項式パラメータになり，従って，g は計算量的に安全ではない．

(2) g は計算量的に安全な疑似乱数生成器でないとする．すなわち，ある A に対して $T_A(n)$ と $S_{g,A}(n)$ が多項式パラメータであるとする．まず，$Y \in_U \{0,1\}^{\ell(n)}$ と $W \in_U \{0,1\}$ を独立にとる．各 $i \in \{1, \ldots, \ell(n)\}$ および $x \in \{0,1\}^{\ell(n)}$ に対して，

$$\tilde{A}(i, x) := \begin{cases} Y_i, & \text{if } A_n(x_1, \ldots, x_{i-1}, Y_i, \ldots, Y_{\ell(n)}) = 1 \\ W, & \text{if } A_n(x_1, \ldots, x_{i-1}, Y_i, \ldots, Y_{\ell(n)}) = 0 \end{cases}$$

とすれば，$\tilde{S}_{g,\tilde{A}}(n)$ は多項式パラメータであることを示そう．そうすれば g は次ビット予測不可能でないことが分かる．

$X := g_n(Z_n)$, $\Pr := \Pr_{Z_n,Y,W}$ と略記すれば，

$$\begin{aligned} &\Pr\left(\tilde{A}_n(i, X_{\{1,\ldots,i-1\}}) = X_i\right) - \frac{1}{2} \\ &\quad = \Pr\left(X_i = Y_i,\, A_n(X_{\{1,\ldots,i-1\}}, Y_{\{i,\ldots,\ell(n)\}}) = 1\right) \\ &\qquad\quad + \Pr\left(X_i = W,\, A_n(X_{\{1,\ldots,i-1\}}, Y_{\{i,\ldots,\ell(n)\}}) = 0\right) - \frac{1}{2} \\ &\quad = \Pr\left(X_i = Y_i,\, A_n(X_{\{1,\ldots,i\}}, Y_{\{i+1,\ldots,\ell(n)\}}) = 1\right) + \frac{1}{2}\Pr\left(A_n(X_{\{1,\ldots,i-1\}}, Y_{\{i,\ldots,\ell(n)\}}) = 0\right) - \frac{1}{2} \\ &\quad = \frac{1}{2}\Pr\left(A_n(X_{\{1,\ldots,i\}}, Y_{\{i+1,\ldots,\ell(n)\}}) = 1\right) + \frac{1}{2}\left(1 - \Pr\left(A_n(X_{\{1,\ldots,i-1\}}, Y_{\{i,\ldots,\ell(n)\}}) = 1\right)\right) - \frac{1}{2} \\ &\quad = \frac{1}{2}\Pr\left(A_n(X_{\{1,\ldots,i\}}, Y_{\{i+1,\ldots,\ell(n)\}}) = 1\right) - \frac{1}{2}\Pr\left(A_n(X_{\{1,\ldots,i-1\}}, Y_{\{i,\ldots,\ell(n)\}}) = 1\right) \end{aligned}$$

従って,

$$\begin{aligned}\tilde{\delta}_{g,\tilde{A}}(n) &= \frac{1}{\ell(n)}\sum_{i=1}^{\ell(n)}\left(\Pr\left(\tilde{A}_n(i, X_{\{1,\ldots,i-1\}}) = X_i\right) - \frac{1}{2}\right)\\ &= \frac{1}{2}\cdot\frac{1}{\ell(n)}\sum_{i=1}^{\ell(n)}\left(\Pr\left(A_n(X_{\{1,\ldots,i\}}, Y_{\{i+1,\ldots,\ell(n)\}}) = 1\right)\right.\\ &\qquad\left. -\Pr\left(A_n(X_{\{1,\ldots,i-1\}}, Y_{\{i,\ldots,\ell(n)\}}) = 1\right)\right)\\ &= \frac{1}{2\ell(n)}\left(\Pr(A_n(X) = 1) - \Pr(A_n(Y) = 1)\right)\end{aligned}$$

だから,

$$\left|\tilde{\delta}_{g,\tilde{A}}(n)\right| = \frac{\delta_{g,A}(n)}{2\ell(n)}$$

$T_{\tilde{A}}(n)$ が多項式パラメータであることは明らかだから，これより $\tilde{S}_{g,\tilde{A}}(n)$ は多項式パラメータである． □

モンテカルロ法で使われている疑似乱数の中には

$$\omega_i := f(\omega_{i-n}, \ldots, \omega_{i-1})$$

のような漸化式で定義されているものが多い．これでは次にくる項が完全に予測できてしまう．従って定理 9 によれば，漸化式で定義された疑似乱数生成器と計算量的に安全な疑似乱数は対極をなしていることが分かる．

定理 9 によれば，計算量的に安全な疑似乱数生成器を作るにはそれが次ビット予測不可能であるように作ればよい．これが計算量的に安全な疑似乱数生成器を設計する際の指導原理となる．この指導原理の下で，計算量的に安全な疑似乱数生成器ではないかと思われているものが幾種類も発案されている．そのうち最初に発案されたものの一つ，**BBS 生成器** ([3]) を紹介しよう．[27] p, q は $p = 3 (\mathrm{mod}\ 4)$，$q = 3 (\mathrm{mod}\ 4)$，を満たす[28] 素数でその値は秘密であるが，その積 $N = pq$ の値は公開される．整数 $1, 2, \ldots, N-1$ の平方を N で割った余り全体を $QR(N)$ と書く．初期値 (疑似乱数の種) $x_0 \in QR(N)$ を選んで

$$\begin{aligned}x_n &:= F(x_{n-1}) = x_{n-1}^2 \bmod N, \qquad n = 1, 2, \ldots,\\ y_n &:= G(x_n) = x_n \bmod 2\end{aligned}$$

とする．p, q が大きい素数のとき，逆写像 F^{-1} の計算は p, q を知っていれば易しいが，知らないと現時点では非常に手間が掛かる．[29] そのため，種 x_0 を知らないで $\{y_n\}_{n=0}^m$ が与えられたとき，次ビット y_{m+1} を予測することは大変困難になると考えられている．それで $\{y_n\}_{n=0}^\infty$ が疑似乱数として提案されたわけである．

[27] BBS 生成器の場合，種を選ぶ集合がすぐあとで述べる $QR(N)$ なので，§ 3.1.1 の定義を少し変更して考える必要がある．

[28] $\bmod M$ は M で割ったときの余りを表す．

[29] 一般に，F は容易に計算できるが F^{-1} の計算が困難であるような関数を一方向関数という．一方向関数の存在を仮定 (それは **P** ≠ **NP** 予想より強い仮定) して，次ビット予測不可能な疑似乱数生成器を構成する一般論が知られている ([14]).

3.2 ワイル変換による疑似乱数生成器

確率論から接近できるある疑似乱数生成器を紹介する．出力される疑似乱数の任意の有限次元分布が具体的に計算できて，硬貨投げの確率過程の対応する有限次元分布に収束することが示すことができる．もちろん，計算量的に安全かどうかは分からないが，ある特別な種類の次ビット予測の成功確率と 1/2 の差が種の大きさの指数関数で減少することを示すことができる (定理 10).

3.2.1 定義

定義 12

1. 区間 $[0,1)$ 上の加法として $(x+y) \bmod 1$ を考えた群を $\mathbb{T}^1$ と書き 1 **次元トーラス** (1-dimensional torus) と呼ぶ．$d_i(x)$ を実数 $x \in \mathbb{T}^1$ の 2 進展開の第 i 桁目の数 (0 または 1) とする．[30] すなわち
$$x = \sum_{i=1}^{\infty} d_i(x) 2^{-i}. \tag{18}$$

2. $(\mathbb{T}^1, \mathcal{B}, \mathbb{P})$ をルベーグ (Lebesgue) 確率空間とする．ここで，$\mathcal{B}$ は $\mathbb{T}^1 = [0,1)$ に含まれるボレル可測集合全体からなる完全加法族，$\mathbb{P}$ はルベーグ測度である．[31] $(\mathbb{T}^1, \mathcal{B}, \mathbb{P})$ の k 直積 (k 次元ルベーグ確率空間) を $(\mathbb{T}^k, \mathcal{B}^k, \mathbb{P}^k)$ と書く．

3. $m \in \mathbb{N}$ に対して
$$D_m := \{ i 2^{-m} \mid i = 0, \ldots, 2^m - 1 \} \subset \mathbb{T}^1 \tag{19}$$
とする．集合族 $\mathcal{I}_m := \{ [a,b) \mid a, b \in D_m \}$ を含むような最小の (完全) 加法族を $\mathcal{B}_m$ と書く．すなわち，$\mathcal{B}_m$ の元は $\mathcal{I}_m$ のいくつかの元の和集合である．

4. 各 $m \in \mathbb{N}$ に対して[32]
$$\lfloor x \rfloor_m := \lfloor 2^m x \rfloor / 2^m \in D_m, \quad x \in \mathbb{T}^1. \tag{20}$$
ただし，$\lfloor x \rfloor_\infty := x$ と約束する．

関数列 $\{ Y_n^{(m)}(x;\alpha) \}_{n=0}^{\infty}$ を次のように定義する．

定義 13　各 $x, \alpha \in \mathbb{T}^1$ と $m \in \mathbb{N}$ に対して，
$$Y_n^{(m)}(x;\alpha) := \sum_{i=1}^{m} d_i\left(x + n\alpha\right) \pmod 2, \qquad n = 0, 1 \ldots \tag{21}$$

ここに，$d_i(\bullet)$ は (18) で定義した関数である．

---

[30] たとえば $1/2 = 0.1 = 0.0111\ldots$ のように 2 進展開が 2 通りある場合は，有限の 2 進展開の方を採用する．すなわち $d_1(1/2) = 1$ である．

[31] 普通は $\mathcal{B}$ を $\mathbb{P}$ で完備化した σ-加法族 (ルベーグ可測集合族) を考えるが，数値計算には $\mathcal{B}$ で十分なのでここではこれを採用している．

[32] $\lfloor t \rfloor$ は実数 t を超えない最大の整数を表す．

(21) をコンピュータプログラムで実現するためには，実数を有限 2 進小数で近似しなければならない: $x, \alpha \in \mathbb{T}^1$ に対して

$$\tilde{x} := \lfloor x \rfloor_{m+j} \in D_{m+j}, \qquad \tilde{\alpha} := \lfloor \alpha \rfloor_{m+j} \in D_{m+j} \tag{22}$$

とおき，さらに写像 $F_{m+j,\alpha} : \Omega \to \Omega$ と $G_m : \Omega \to \{0,1\}$ を

$$F(\tilde{x}) \;=\; F_{m+j,\alpha}(\tilde{x}) \;:=\; (\tilde{x} + \tilde{\alpha}) \bmod 1, \tag{23}$$

$$G(\tilde{x}) \;=\; G_m(\tilde{x}) \;:=\; \sum_{i=1}^{m} d_i(\tilde{x}) \bmod 2, \tag{24}$$

とすれば

$$Y_n^{(m)}(\tilde{x}; \tilde{\alpha}) = G(F^n(\tilde{x})), \quad n = 0, 1, \ldots, 2^j - 1.$$

となる．ここに $F^n(\tilde{x})$ は $\tilde{x}$ に F を n 回作用させることを表す.

この関数列を用いて疑似乱数生成器

$$\{Y_n^{(m)}(\bullet; \tilde{\alpha})\}_{n=0}^{2^j-1} \;: D_{m+j} \cong \{0,1\}^{m+j} \;\to\; \{0,1\}^{2^j}$$

を考えよう.[33] とくに α が無理数のとき $m \to \infty$ における振る舞いに興味があって，これを**ワイル変換による疑似乱数生成器**と呼ぶ．ワイル変換 (Weyl transformation) とは無理数 $\alpha \in \mathbb{T}^1$ に対して $\mathbb{T}^1$ 上の変換 $x \mapsto x + \alpha$ をいう．無理数回転ともいう．この疑似乱数生成器の具体的な実装については § 5 または [25] を見よ.

3.2.2 ある特殊な次ビット予測の困難性

疑似乱数生成器 $\{Y_n^{(m)}(\bullet; \tilde{\alpha})\}_{n=0}^{2^j-1}$ でパラメータ m を大きくすれば,[34] 関数 G_m の複雑性が増大し，そのため，だんだん次ビット予測が困難になっていくと想像される．このことがある特殊な次ビット予測の場合に実際に起こっていることを紹介しよう.

極限定理を述べるために α は有限 2 進小数ではなく実数で考える．$l \in \mathbb{N}$，$0 < k_1 < \ldots < k_{l-1}$ とする．$\{Y_{k_j}^{(m)}(x; \alpha)\}_{j=0}^{l-2}$ の値を知ったときに，$Y_{k_{l-1}}^{(m)}(x; \alpha)$ の値を予測する方法を考えよう．次の量を考える:

$$F^{(m)}(k_0, \ldots, k_{l-1}; \alpha) := \mathbb{P}\left(\sum_{j=0}^{l-1} Y_{k_j}^{(m)}(\,\bullet\,; \alpha) = \text{奇数} \right), \tag{25}$$

$$0 \le k_0 < \cdots < k_{l-1}, \qquad l \in \mathbb{N}.$$

あとの定理 11 で述べるとおり，この量を計算する高速なアルゴリズムが存在する．そのことを利用して，次ビットを予測する関数 $A : \{0,1\}^{l-1} \to \{0,1\}$ を次のように定義する.

$$A(y_{k_0}, \ldots, y_{k_{l-2}}) := \begin{cases} \mathbf{1}_{\{y_{k_0} + \cdots + y_{k_{l-2}} = \text{偶数}\}}, & \text{if } F^{(m)}(k_0, k_1, \ldots, k_{l-1}; \alpha) \ge \frac{1}{2}, \\ \mathbf{1}_{\{y_{k_0} + \cdots + y_{k_{l-2}} = \text{奇数}\}}, & \text{if } F^{(m)}(k_0, k_1, \ldots, k_{l-1}; \alpha) < \frac{1}{2}. \end{cases}$$

[33] $j \in \mathbb{N}$ は m に依存して大きくとるべきだが，記法が複雑になるのでここではこのように書いた.
[34] (22) より，m はおよそ種の大きさを表している.

そして，A によって $Y^{(m)}_{k_{l-1}}(x;\alpha)$ の値を $A(Y^{(m)}_{k_0}(x;\alpha),\ldots,Y^{(m)}_{k_{l-2}}(x;\alpha))$ と予測する．このとき，この予測の的中する確率は

$$\mathbb{P}\left(A(Y^{(m)}_{k_0},\ldots,Y^{(m)}_{k_{l_2}})=Y^{(m)}_{k_{l-1}}\right) = \left|F^{(m)}(k_0,k_1,\ldots,k_{l-1};\alpha)-\frac{1}{2}\right|+\frac{1}{2} \tag{26}$$

となる．実際，$F^{(m)}(k_0,k_1,\ldots,k_{l-1};\alpha)\geq 1/2$ の場合は

$$\begin{aligned}\mathbb{P}\left(A(Y^{(m)}_{k_0},\ldots,Y^{(m)}_{k_{l-2}})=Y^{(m)}_{k_{l-1}}\right) &= \mathbb{P}\left(Y^{(m)}_{k_0}+\cdots+Y^{(m)}_{k_{l-1}}=\text{奇数}\right)\\ &= F^{(m)}(k_0,k_1,\ldots,k_{l-1};\alpha),\end{aligned}$$

また，$F^{(m)}(k_0,k_1,\ldots,k_{l-1};\alpha)<1/2$ の場合は

$$\begin{aligned}\mathbb{P}\left(A(Y^{(m)}_{k_0},\ldots,Y^{(m)}_{k_{l-2}})=Y^{(m)}_{k_{l-1}}\right) &= \mathbb{P}\left(Y^{(m)}_{k_0}+\cdots+Y^{(m)}_{k_{l-1}}=\text{偶数}\right)\\ &= 1-F^{(m)}(k_0,k_1,\ldots,k_{l-1};\alpha),\end{aligned}$$

となって，いずれの場合も (26) が成り立つ．

関数 A による予測は 1/2 以上の確率で的中するが，その的中確率 (26) について，次の定理が成り立つ．

定理 10　$\mathbb{P}$ に関してほとんどすべての $\alpha\in\mathbb{T}^1$ について，任意の $l\in\mathbb{N}$，$0<k_1<\ldots<k_{l-1}$，に対して，ある $0<\rho<1$ が存在し，

$$\mathbb{P}\left(A(Y^{(m)}_{k_0},\ldots,Y^{(m)}_{k_{l_2}})=Y^{(m)}_{k_{l-1}}\right)-\frac{1}{2} = \left|F^{(m)}(k_0,k_1,\ldots,k_{l-1};\alpha)-\frac{1}{2}\right| = O(\rho^m),\quad m\to\infty.$$

この定理は A による次ビット予測が m の増加とともに指数関数的に困難になっていくことを示している．このような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．

なお，$l=1$ のときには，任意の $\rho>\sqrt{(1+\sqrt{17})/8}=0.80024\ldots$ について定理 10 の主張が成り立つ (§ 3.3.5 定理 14).

3.2.3　有限次元分布の計算公式と従属性の消滅

ルベーグ確率空間で定義された確率過程 $\{Y^{(m)}_n(\bullet;\alpha)\}_{n=0}^{\infty}$ の有限次元分布を計算するアルゴリズムが存在する．

補題 1　(i) $\epsilon_n\in\{0,1\}$，$n=0,1,\ldots,k-1$，として次の等式が成り立つ．

$$\begin{aligned}&\mathbb{P}\left(Y^{(m)}_n(\bullet;\alpha)=\epsilon_n,\quad n=0,\ldots,k-1\right)\\ &=2^{-k}\left(\sum_{l=1}^{k}\sum_{0\leq k_0<\cdots<k_{l-1}\leq k-1}\prod_{j=0}^{l-1}(1-2\epsilon_{k_j})\left(1-2F^{(m)}(k_0,\ldots,k_{l-1};\alpha)\right)+1\right).\end{aligned}$$

(ii) $l\in\mathbb{N}$ が奇数ならば $F^{(m)}(k_0,\ldots,k_{l-1};\alpha)=1/2$ である．
(iii) $F^{(m)}(k_0,\ldots,k_{l-1};\alpha)=F^{(m)}(0,k_1-k_0,\ldots,k_{l-1}-k_0;\alpha)$．すなわち，$k_0=0$ の場合だけ求められれば十分である．

以下，l は偶数とし，$F^{(m)}(0,k_1,\dots,k_{l-1};\alpha)$ を求めるアルゴリズムを紹介する．そのために，いくつかの記号を導入する．$\alpha\in\mathbb{T}^1$ に対して，

$$\alpha_j := \langle k_j\alpha\rangle,\quad j=1,\dots,l-1,$$

とおき，[35]

$$\begin{cases} \alpha_j^{(m)L} &:= \lfloor\alpha_j\rfloor_m,\\ \alpha_j^{(m)U} &:= \lfloor\alpha_j\rfloor_m + 2^{-m} \pmod 1,\\ \beta_j^{(m)} &:= 2^m(\alpha_j-\alpha_j^{(m)L}),\end{cases}$$

とする．次に集合 $\{1,\dots,l-1\}$ 上の 置換 $\sigma(m,\bullet)$ を以下のように定める．[36]

$$1>\beta^{(m)}_{\sigma(m,1)}\ge\beta^{(m)}_{\sigma(m,2)}>\cdots\ge\beta^{(m)}_{\sigma(m,l-1)}\ge 0. \tag{27}$$

さらに，$\beta^{(m)}_{\sigma(m,0)}:=1$, $\beta^{(m)}_{\sigma(m,l)}:=0$ と約束しておく．そして

$$\alpha^{(m),s}_{\sigma(m,j)} := \begin{cases}\alpha^{(m)U}_{\sigma(m,j)}, & (\,j\le s\,)\\ \alpha^{(m)L}_{\sigma(m,j)}, & (\,j>s\,)\end{cases}$$

とした上で

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

とおく．最後に，有限 2 進小数全体の集合を以下のように定義する．

$$D := \bigcup_{m\in\mathbb{N}} D_m.$$

定理 11

$$F^{(m)}(0,k_1,\dots,k_{l-1};\alpha) \;=\; \sum_{s=0}^{l-1}\left(\beta^{(m)}_{\sigma(m,s)}-\beta^{(m)}_{\sigma(m,s+1)}\right)B(\boldsymbol{\alpha}^{(m),s}). \tag{28}$$

ここに $B(\bullet)$ 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

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

および次の漸化式で計算される．

$$B(\boldsymbol{\alpha}^{(m),s}) \;=\; \begin{cases}\dfrac{1}{2}B(\boldsymbol{\alpha}^{(m-1),s_2})+\dfrac{1}{2}B(\boldsymbol{\alpha}^{(m-1),s_1+s_2}) & (\,s_1\text{ は偶数}\,)\\[2ex] \dfrac{1}{2}\left(1-B(\boldsymbol{\alpha}^{(m-1),s_2})\right)+\dfrac{1}{2}\left(1-B(\boldsymbol{\alpha}^{(m-1),s_1+s_2})\right) & (\,s_1\text{ は奇数}\,)\end{cases}$$

ただし，s_1, s_2 は次で与えられる．

$$s_1 := \sum_{j=1}^{l-1} d_m(\alpha_j^{(m),s}),\qquad s_2 := \sum_{j=1}^{s} d_m(\alpha_{\sigma(m,j)}). \tag{29}$$

[35] $\langle t\rangle$ は実数 $t\ge 0$ の小数部分を表す．すなわち，$\langle t\rangle = t-\lfloor t\rfloor$.

[36] (27) において等号が成り立つ場合は $\sigma(m,j)$ が一意に定まらないが，(28) の右辺の値は一意に定まるので問題はない．

定理 10 と補題 1 から，次の従属性消滅定理[37]が従う．

定理 12 ([19, 34]) $\mathbb{P}$ に関してほとんどすべての $\alpha \in \mathbb{T}^1$ に対して，$\{Y_n^{(m)}(\bullet\,;\alpha)\}_{n=0}^{\infty}$ の各有限次元分布は $m \to \infty$ のとき，硬貨投げの確率過程に対応する分布に指数的に収束する．すなわち，任意の $\epsilon_n \in \{0,1\}$，$n = 0, 1, \ldots, k-1$ に対して，ある $0 < \rho < 1$ が存在して

$$\left| \mathbb{P}\left(Y_n^{(m)}(\bullet\,;\alpha) = \epsilon_n, \quad n = 0, \ldots, k-1 \right) - 2^{-k} \right| \;=\; O(\rho^m), \quad m \to \infty.$$

この定理とは別に次の定理も成り立つ．

定理 13 ([35]) すべての無理数 $\alpha \in \mathbb{T}^1$ に対して，$\{Y_n^{(m)}(\bullet\,;\alpha)\}_{n=0}^{\infty}$ の各有限次元分布は $m \to \infty$ のとき，硬貨投げの確率過程に対応する分布に収束する．すなわち，任意の $\epsilon_n \in \{0,1\}$，$n = 0, 1, \ldots, k-1$ に対して

$$\lim_{m\to\infty} \mathbb{P}\left(Y_n^{(m)}(\bullet\,;\alpha) = \epsilon_n, \quad n = 0, \ldots, k-1 \right) \;=\; 2^{-k}.$$

3.2.4 有限次元分布の事前評価

定理 11 を用いると $\{Y_n^{(m)}(\bullet\,;\alpha)\}_{n=0}^{\infty}$ の有限次元分布に関する統計的性質を調べることができる．

はじめに，二項間の相関について，

$$N^{(m)}(k;\alpha) \;:=\; \frac{1}{16\left(F^{(m)}(0,k;\alpha) - \frac{1}{2}\right)^2} \tag{30}$$

と定義する．このとき，以下のように疑似乱数の使用限界を推定することができる．まず

$$\begin{cases} \eta_{n;k}^{(m)}(\bullet\,;\alpha) \;:=\; Y_n^{(m)}(\bullet\,;\alpha) + Y_{n+k}^{(m)}(\bullet\,;\alpha) \pmod 2 \\ S_{N;k}^{(m)}(\bullet\,;\alpha) \;:=\; \dfrac{1}{N}\displaystyle\sum_{n=0}^{N-1} \eta_{n;k}^{(m)}(\bullet\,;\alpha) \end{cases}$$

とおく．$\{Y_n^{(m)}(\bullet\,;\alpha)\}_{n=0}^{\infty}$ が硬貨投げの確率過程だったと仮定すれば S_N の分散は $\sigma_N^2 := 1/(4N)$ である．各 $k \in \mathbb{N}$ に対して次の仮説

$$\mathbf{E}\left[S_{N;k}^{(m)}(\bullet\,;\alpha) \right] \;\equiv\; F^{(m)}(0,k;\alpha) \;=\; \frac{1}{2} \tag{31}$$

の検定を行うために

$$\left| S_{N;k}^{(m)}(\bullet\,;\alpha) - \frac{1}{2} \right| \;<\; 2\sigma_N \;=\; \frac{1}{\sqrt{N}} \tag{32}$$

となる確率を求める．$\{Y_n^{(m)}(\bullet\,;\alpha)\}_{n=0}^{\infty}$ が硬貨投げの確率過程であるという仮説の下では (32) の確率は，中心極限定理によって正規分布で近似すれば，約 95% である．

[37] 安富は一連の論文 [32, 33, 34, 35] によって定理 12 を様々に拡張した諸定理を証明している．このノートでは，それらの論文の証明の一部のアイデアを，ここでの議論の文脈に合わせて紹介する．

命題 1　m が十分大きいとき，$N = N^{(m)}(k;\alpha)$ (以下) ならば，事象 (32) の確率は約 92% (以上) である．

証明. m が十分大きければ疑似乱数 $\{Y_n^{(m)}(\bullet\,;\alpha)\}_{n=0}^{\infty}$ が十分ランダムであるので，S_N の分散は $\{\eta_n^{(m)}(\bullet\,;\alpha)\}_{n=0}^{\infty}$ を硬貨投げの確率過程と考えたときの $S_{N;k}^{(m)}(\bullet\,;\alpha)$ の分散 σ_N^2 にほぼ等しいであろう．N が十分大きければ，$S_{N;k}^{(m)}(\bullet\,;\alpha)$ の分布は中心極限定理により，ほぼ $N(1/2+a, \sigma_N^2)$ に従う．ただし，$a = F^{(m)}(0,k;\alpha) - 1/2$ である．いま $N = N^{(m)}(k;\alpha) = 1/(16a^2)$ とすれば，$|a| = \sigma_N/2$ であるから，

$$\left|S_{N;k}^{(m)} - \frac{1}{2}\right| < 2\sigma_N \iff \begin{cases} -\dfrac{5\sigma_N}{2} < S_{N;k}^{(m)} - \left(\dfrac{1}{2}+a\right) < \dfrac{3\sigma_N}{2} & (a>0) \\ -\dfrac{3\sigma_N}{2} < S_{N;k}^{(m)} - \left(\dfrac{1}{2}+a\right) < \dfrac{5\sigma_N}{2} & (a<0) \end{cases}$$

だから，いずれにしろ，上の確率は簡単な変数変換によって

$$\int_{-5/2}^{3/2} \frac{1}{\sqrt{2\pi}} e^{-x^2/2}\,dx \;=\; 0.926983$$

に等しい．これが「約 92%」の理由である．　□

命題 1 を根拠に，(31) の検定に対し，疑似乱数 $\{Y_n^{(m)}(\bullet\,;\alpha)\}_{n=0}^{\infty}$ の使用限界が臨界サンプル数 $N^{(m)}(k;\alpha)$ 程度と考えることができよう．命題 1 では「m が十分大きいとき」とあるが，実際はそれほど大きくない m でも命題 1 の評価はほぼ正しいことが，次の例から分かる．

例 8　$\alpha = (\sqrt{5}-1)/2$, $m = 40$ および $k = 305$ として定理 11 を適用すれば，$F^{(40)}(0,305;\alpha) = 0.5029834$ であることが分かる．このとき，

$$\frac{1}{16\left(F^{(40)}(0,305;\alpha) - \frac{1}{2}\right)^2} = \frac{1}{16\times(0.0029834)^2} = 7021.94 \sim 7022$$

となる．そこで，

$$\left|S_{7022;305}^{(40)} - \frac{1}{2}\right| \;<\; \frac{1}{\sqrt{7022}}, \tag{33}$$

となる確率を数値的に求めてみた．計算方法は次の通りである:

$$\{Y_n^{(40)}(0;\alpha)\}_{n=1}^{7022\times 10^6+305}$$

をコンピュータで生成し，$i = 1, 2, \ldots, 10^6$ に対して，

$$p_i := \frac{1}{7022}\#\{7022(i-1)+1 \le j \le 7022\,i \,|\, \eta_{j;305}^{(40)}(0;\alpha) = 1\},$$

を計算した．このとき，

$$\left(p_i - \frac{1}{2}\right) \text{ の平均} \;=\; 10^{-6}\sum_{i=1}^{10^6}(p_i - 1/2) \;=\; 0.002983535$$

$$p_i \text{ の分散} \;=\; 10^{-6}\sum_{i=1}^{10^6}(p_i - 0.502983535)^2 \;=\; 0.0000370605$$

を得た．平均の方は理論値 0.0029834 に近い．分散は $\{Y_n^{(m)}\}_{n=0}^{\infty}$ が硬貨投げの確率過程と仮定したときの値 $1/(4\times 7022) = 0.0000356024$ と比べて 4% ほど大きい．最後に

$$\left| p_i - \frac{1}{2} \right| < \frac{1}{\sqrt{7022}}$$

を満たす i は 921514 個，すなわち (33) の確率として 92.1514% という近似値を得た．

以下の例でも，ワイル変換に用いる無理数として，引き続き黄金分割の比として知られる次の数を採用した．

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

命題 1 を踏まえて，二項間の相関について $K \in \mathbb{N}$ に対し，

$$a^{(m)}(K) := \max_{1\le k\le K} \left| F^{(m)}(0,k;\alpha) - \frac{1}{2} \right|, \qquad N_c^{(m)}(K) := \frac{1}{16\,(a^{(m)}(K))^2}, \tag{34}$$

とする．$N_c^{(m)}(K)$ を臨界サンプル数と呼ぶ.[38] (34) を $K = 10,000$ のときに計算し，表にしたのが表 1 の左半分である．$a^{(m)}(K)$ 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．

表 1: 二項間・多項間分布の評価

m	$a^{(m)}(10000)$	(k)	$N_c^{(m)}(10000)$	$b^{(m)}(16)$	$k_1,\ldots$
10	0.4860680	(5473)	2.6×10^{-1}	0.1099945	1,9,10
20	0.1084934	(1449)	5.3×10^{0}	0.0053298	9
30	0.0435756	(305)	3.3×10^{1}	0.0008288	9
40	0.0029834	(305)	7.0×10^{3}	0.0000769	9
50	0.0001943	(610)	1.7×10^{6}	8.0×10^{-6}	9
60	0.0000136	(8484)	3.4×10^{8}	6.1×10^{-7}	9
70	1.2×10^{-6}	(7264)	4.1×10^{10}	5.9×10^{-8}	1
80	2.0×10^{-7}	(7697)	1.6×10^{12}	6.8×10^{-9}	16
90	8.5×10^{-9}	(165)	8.7×10^{14}	2.1×10^{-9}	16
100	2.9×10^{-9}	(5201)	7.7×10^{15}	3.0×10^{-10}	1

次に一般の有限次元分布 (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

$$F^{(m)}(0,k_1,\ldots,k_{l-1};\alpha), \qquad 1\le k_1<\cdots<k_{l-1}\le K,$$

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．表 1 の右半分は $K = 16$ の場合を計算したものである．左の欄は，

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

[38] [19] 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．

を表わし，右の欄はその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．表 1 の右半分からは次の仮説が成り立つように見受けられる．

仮説 1　各 $K \in \mathbb{N}$ に対して m が十分大きいとき，

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

実は仮説 1 は定理 12 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 (注意 10)，現在のところ厳密な証明はない．もし仮説 1 が正しければ，我々は二項間の相関の最大値さえ評価すればよいことになる．

3.3　ワイル変換による疑似乱数生成器に関する定理の証明

補題 1 と定理 11，定理 13，定理 10 をこの順序で証明する．ただし，ここでの証明は確率過程 $\{Y_n^{(m)}\}_{n=0}^{\infty}$ ではなくて，それと同等の，(35) で定義される $\{-1, 1\}$-値確率過程 $\{X_n^{(m)}\}_{n=0}^{\infty}$ について行う．$\{Y_n^{(m)}\}_{n=0}^{\infty}$ の方がコンピュータプログラムで実現しやすいが，$\{X_n^{(m)}\}_{n=0}^{\infty}$ の方が定理を証明しやすいからである．

$\{r_i\}_{i=1}^{\infty}$ を**ラデマッハ関数列** (Rademacher functions)，すなわち

$$r_i(x) := 1 - 2d_i(x), \quad x \in \mathbb{T}^1, \quad i \in \mathbb{N},$$

とする．無理数 $\alpha \in [0, 1)$ と自然数 m に対して，

$$X_n^{(m)}(x; \alpha) := \prod_{i=1}^{m} r_i(x + n\alpha), \qquad n \in \mathbb{N}, \tag{35}$$

と定義する．$\{X_n^{(m)}\}_{n=0}^{\infty}$ と $\{Y_n^{(m)}\}_{n=0}^{\infty}$ は次の関係にある．

$$X_n^{(m)}(x; \alpha) = 1 - 2Y_n^{(m)}(x; \alpha), \qquad Y_n^{(m)}(x; \alpha) = \frac{1}{2}\left(1 - X_n^{(m)}(x; \alpha)\right).$$

次のような性質に注意する: 任意の $k, h \in \mathbb{N}$ と任意の $\boldsymbol{\epsilon} \in \{-1, 1\}^k$ に対して

$$\mathbb{P}\left((X_0^{(m)}(\bullet; \alpha), \dots, X_{k-1}^{(m)}(\bullet; \alpha)) = \boldsymbol{\epsilon}\right) = \mathbb{P}\left((X_h^{(m)}(\bullet; \alpha), \dots, X_{k-1+h}^{(m)}(\bullet; \alpha)) = \boldsymbol{\epsilon}\right) \tag{36}$$

この性質は定常性 (詳しくは強定常性) と呼ばれる．(36) の証明はルベーグ測度の平行移動不変性 — すなわち，部分区間の長さは平行移動によって変わらないこと — を使えば容易に導かれる．

3.3.1　補題 1 の証明

補題 1 を $\{X_n^{(m)}\}_{n=0}^{\infty}$ の言葉で表すと次のようになる．

補題 1′ (i) 任意の有限次元分布は次の量から算出することができる.

$$E^{(m)}(k_0,\dots,k_{l-1};\alpha) := \mathbf{E}\left[\prod_{j=0}^{l-1} X_{k_j}^{(m)}(\bullet;\alpha)\right], \qquad 0\le k_0<\dots<k_{l-1}, \quad l\in\mathbb{N}.$$

実際, $\epsilon_n\in\{-1,1\}$ として次の等式が成り立つ.

$$\begin{aligned}&\mathbb{P}\left(X_n^{(m)}(\bullet;\alpha)=\epsilon_n, \quad n=0,\dots,k-1\right)\\ &\quad = 2^{-k}\left(\sum_{l=1}^{k}\sum_{0\le k_0<\dots<k_{l-1}\le k-1}\prod_{j=0}^{l-1}\epsilon_{k_j}E^{(m)}(k_0,\dots,k_{l-1};\alpha)+1\right).\end{aligned} \tag{37}$$

(ii) $l\in\mathbb{N}$ が奇数ならば $E^{(m)}(k_0,\dots,k_{l-1};\alpha)=0$.
(iii) $E^{(m)}(k_0,\dots,k_{l-1};\alpha)=E^{(m)}(0,k_1-k_0,\dots,k_{l-1}-k_0;\alpha)$.

証明. (i) $\epsilon_i\in\{-1,1\}$ として, 次の等式に注意する.

$$\sum_{l=1}^{k}\sum_{0\le k_0<\dots<k_{l-1}\le k-1}\prod_{j=0}^{l-1}\left(\epsilon_{k_j}X_{k_j}^{(m)}(x;\alpha)\right) = \prod_{n=0}^{k-1}\left(1+\epsilon_nX_n^{(m)}(x;\alpha)\right) - 1. \tag{38}$$

ここで左辺を平均すれば,

$$\sum_{l=1}^{k}\sum_{0\le k_0<\dots<k_{l-1}\le k-1}\prod_{j=0}^{l-1}\epsilon_{k_j}E^{(m)}(k_0,\dots,k_{l-1};\alpha). \tag{39}$$

一方, 右辺の平均は

$$\mathbf{E}\left[\prod_{n=0}^{k-1}\left(1+\epsilon_nX_n^{(m)}(x;\alpha)\right)\right]-1. \tag{40}$$

ところが, (40) の $\mathbf{E}[\,\bullet\,]$ の中は, すべての $n=0,\dots,k-1$ に対して $X_n^{(m)}(x;\alpha)=\epsilon_n$ のときは 2^k, それ以外のときは 0 になるから, (40) の値は次に等しいことが分かる.

$$2^k\mathbb{P}\left(X_n^{(m)}(\bullet;\alpha)=\epsilon_n, \quad n=0,\dots,k-1\right)-1. \tag{41}$$

(39) と (41) が等しいことより (37) が従う.

(ii) $r_1(x+\frac{1}{2})=-r_1(x)$, $r_i(x+\frac{1}{2})=r_i(x)$, $i\ge 2$, だから

$$X_k^{(m)}\left(x+\frac{1}{2};\alpha\right) = -X_k^{(m)}(x;\alpha), \qquad x\in[0,1), \tag{42}$$

であることがすぐ分かる. それで, l が奇数のとき,

$$X_0^{(m)}(x;\alpha)\times\cdots\times X_{k_{l-1}}^{(m)}(x;\alpha) = -1$$

と

$$X_0^{(m)}\left(x+\frac{1}{2};\alpha\right)\times\cdots\times X_{k_{l-1}}^{(m)}\left(x+\frac{1}{2};\alpha\right) = 1$$

は同値であるから, それぞれの確率も等しい. ところが, ルベーグ測度の平行移動不変性によって後者の確率は

$$X_0^{(m)}(x;\alpha)\times\cdots\times X_{k_{l-1}}^{(m)}(x;\alpha) = 1$$

の確率に等しいから, これらの確率はすべて 1/2 でなくてはならない. これより補題 1′ (ii) の主張が従う. (iii) は定常性 (36) により明らか. □

3.3.2 定理 11 の証明

定理 11 を $\{X_n^{(m)}\}_{n=0}^{\infty}$ の言葉で述べると次のようになる.

定理 11′

$$E^{(m)}(0, k_1, \ldots, k_{l-1}; \alpha) \;=\; \sum_{s=0}^{l-1} \left(\beta^{(m)}_{\sigma(m,s)} - \beta^{(m)}_{\sigma(m,s+1)}\right) A(\boldsymbol{\alpha}^{(m),s}). \tag{43}$$

ここに $A(\bullet)$ 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，$A(\boldsymbol{\alpha}^{(m),s})$ の値は

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

および次の漸化式で計算される.

$$A(\boldsymbol{\alpha}^{(m),s}) \;=\; \frac{(-1)^{s_1}}{2} \left(A(\boldsymbol{\alpha}^{(m-1),s_2}) + A(\boldsymbol{\alpha}^{(m-1),s_1+s_2})\right).$$

ただし，s_1, s_2 などの記号は定理 11 と同じ.

定理 11′ の証明を与えよう．以下ではずっと，l が偶数であることを仮定する．定義によって,

$$E^{(m)}(0, k_1, \ldots, k_{l-1}; \alpha) \;=\; \mathbf{E}\left[\prod_{i=1}^{m} r_i(x) r_i(x + k_1\alpha) \times \cdots \times r_i(x + k_{l-1}\alpha)\right].$$

このことを念頭に入れて，一般の $\boldsymbol{\alpha} = (\alpha_1, \ldots, \alpha_{l-1}) \in [0,1)^{l-1}$ に対して次の定義を設ける.

$$A^{(m)}(\boldsymbol{\alpha}) \;:=\; \mathbf{E}\left[\prod_{i=1}^{m} r_i(x) r_i(x + \alpha_1) \times \cdots \times r_i(x + \alpha_{l-1})\right] \tag{44}$$

補題 2　各 $\boldsymbol{\alpha} = (\alpha_1, \ldots, \alpha_{l-1}) \in (D_m)^{l-1}$ に対して[39]

$$A^{(m')}(\boldsymbol{\alpha}) \;=\; A^{(m)}(\boldsymbol{\alpha}), \qquad m' > m.$$

が成り立つ.

証明. $A^{(m')}(\boldsymbol{\alpha})$ を次のように表す.

$$\begin{aligned} A^{(m')}(\boldsymbol{\alpha}) \;\;=\;\; \mathbf{E}\Bigg[&\prod_{i=1}^{m} r_i(x) r_i(x + \alpha_1) \times \cdots \times r_i(x + \alpha_{l-1}) \\ &\times \prod_{i=m+1}^{m'} r_i(x) r_i(x + \alpha_1) \times \cdots \times r_i(x + \alpha_{l-1})\Bigg]. \end{aligned}$$

$\boldsymbol{\alpha} \in (D_m)^{l-1}$ ならば，$i > m$ のとき次が成り立つ.

$$r_i(x) = r_i(x + \alpha_j), \quad j = 1, \ldots, l-1.$$

[39] D_m は (19) で定義された $[0,1)$ の部分集合.

すると，l は偶数なので後半の積は

$$\prod_{i=m+1}^{m'} r_i(x)r_i(x+\alpha_1)\times\cdots\times r_i(x+\alpha_{l-1}) \;=\; \prod_{i=m+1}^{m'} r_i(x)^l = 1$$

となって全体の積 $\prod_{i=1}^{m'}$ に何の影響も及ぼさない．従って $A^{(m')}(\boldsymbol{\alpha}) = A^{(m)}(\boldsymbol{\alpha})$. □

定義 14　各 $\boldsymbol{\alpha}\in D^{l-1}$ に対して，次のように定義する．

$$A(\boldsymbol{\alpha}) \;:=\; \lim_{m\to\infty} A^{(m)}(\boldsymbol{\alpha}).$$

定義 14 は 補題 2 によって正当化される．以下の補題 3 で関数 A の値がある漸化式によって求められることを示す．そのための準備をしよう．私たちは定理 11 において $\boldsymbol{\alpha}^{(m),s}$ など無理数を成分とするベクトル $\boldsymbol{\alpha}$ について定義したが，ここでは一般のベクトル $\boldsymbol{\alpha}\in[0,1)^{l-1}$ に対しても同様のものを定義する．

各 $\boldsymbol{\alpha}=(\alpha_1,\dots,\alpha_{l-1})\in[0,1)^{l-1}$ に対して

$$\boldsymbol{\alpha}^{(m)L} := (\alpha_1^{(m)L},\dots,\alpha_{l-1}^{(m)L}),\quad \alpha_j^{(m)L} := \lfloor\alpha_j\rfloor_m,$$

と定義する．このとき $\boldsymbol{\alpha}^{(m)L}\in(D_m)^{l-1}$ は明らか．次に

$$\alpha_j^{(m)U} \;:=\; \begin{cases} \lfloor\alpha_j\rfloor_m + 2^{-m} \pmod 1 & (\alpha_j\notin D_m)\\ \alpha_j & (\alpha_j\in D_m)\end{cases}$$

と定義し，さらに

$$\boldsymbol{\alpha}^{(m)U} := (\alpha_1^{(m)U},\dots,\alpha_{l-1}^{(m)U})$$

と定義する．このとき $\boldsymbol{\alpha}^{(m)U}\in(D_m)^{l-1}$ は明らか.

以上の定義の下で次の補題が成り立つ.

補題 3　(i) $A(\overbrace{0,\dots,0}^{l}) = 1$.

(ii) 各 $\boldsymbol{\alpha}=(\alpha_1,\dots,\alpha_{l-1})\in(D_m)^{l-1}$ に対して $r := \sum_{j=1}^{l-1} d_m(\alpha_j)$ と置く．このとき，次が成り立つ.

$$A(\boldsymbol{\alpha}) \;=\; \frac{(-1)^r}{2}\Big(A(\boldsymbol{\alpha}^{(m-1)U}) + A(\boldsymbol{\alpha}^{(m-1)L})\Big). \tag{45}$$

証明. (i) l は偶数なので

$$A(\overbrace{0,\dots,0}^{l}) \;=\; \mathbf{E}\left[\prod_{i=1}^{m}\overbrace{r_i(x)\times\cdots\times r_i(x)}^{l}\right] \;=\; 1.$$

(ii) を示そう． 次が成り立つことに注意せよ．

$$\prod_{i=1}^{m} r_i(x+\alpha_j) \;=\; \begin{cases} \displaystyle\prod_{i=1}^{m-1} r_i\left(x+\alpha_j+2^{-m}\right) & (\,d_m(\alpha_j)=1,\quad d_m(x)=1\,) \\ \displaystyle -\prod_{i=1}^{m-1} r_i\left(x+\alpha_j-2^{-m}\right) & (\,d_m(\alpha_j)=1,\quad d_m(x)=0\,) \\ \displaystyle -\prod_{i=1}^{m-1} r_i\left(x+\alpha_j\right) & (\,d_m(\alpha_j)=0,\quad d_m(x)=1\,) \\ \displaystyle\prod_{i=1}^{m-1} r_i\left(x+\alpha_j\right) & (\,d_m(\alpha_j)=0,\quad d_m(x)=0\,) \end{cases}$$

実際，もし $d_m(\alpha_j)=0$ ならば $r_m(x+\alpha_j)=r_m(x)$ であるから

$$\prod_{i=1}^{m} r_i(x+\alpha_j) \;=\; \prod_{i=1}^{m-1} r_i(x+\alpha_j)\times r_m(x).$$

これより，第 3 と第 4 の場合が分かる．

それでは $d_m(\alpha_j)=1$ と仮定してみよう．このときは $r_m(x+\alpha_j)=-r_m(x)$ である．さらに $d_m(x)=1$ と仮定してみる．この場合は $d_m(x+\alpha_j)=0$ なので，$i=1,\ldots,m-1$ に対して $d_i(x+\alpha_j)=d_i(x+\alpha_j+2^{-m})$，つまり $r_i(x+\alpha_j)=r_i(x+\alpha_j+2^{-m})$ である．従って，

$$\prod_{i=1}^{m} r_i(x+\alpha_j) \;=\; \prod_{i=1}^{m-1} r_i(x+\alpha_j)\times r_m(x+\alpha_j) \;=\; \prod_{i=1}^{m-1} r_i\left(x+\alpha_j+2^{-m}\right)$$

が分かるが，これは第 1 の場合を示している．

最後に $d_m(\alpha_j)=1$ かつ $d_m(x)=0$ である場合について検討しよう．今度は $d_m(x+\alpha_j)=1$ なので，$i=1,\ldots,m-1$ に対して $d_i(x+\alpha_j)=d_i(x+\alpha_j-2^{-m})$，つまり $r_i(x+\alpha_j)=r_i(x+\alpha_j-2^{-m})$ である．従って，

$$\prod_{i=1}^{m} r_i(x+\alpha_j) \;=\; \prod_{i=1}^{m-1} r_i(x+\alpha_j)\times r_m(x+\alpha_j) \;=\; -\prod_{i=1}^{m-1} r_i\left(x+\alpha_j-2^{-m}\right)$$

が分かるが，これは第 2 の場合を示している．

記号を簡単にするため，次の状況を仮定しよう．

$$d_m(\alpha_j) \;=\; \begin{cases} 1, & 1\le j\le r, \\ 0, & r+1\le j\le l-1. \end{cases}$$

$r=\sum_{j=1}^{l-1} d_m(\alpha_j)$ となることに注意せよ．このとき，

$$A(\boldsymbol{\alpha}) \;=\; \mathbf{E}\left[\prod_{i=1}^{m}\left(r_i(x)\prod_{j=1}^{r} r_i(x+\alpha_j)\prod_{j=r+1}^{l-1} r_i(x+\alpha_j)\right)\right]$$

$$= \mathbf{E}\left[-\prod_{i=1}^{m-1} r_i(x) \prod_{j=1}^{r} \prod_{i=1}^{m-1} r_i\left(x+\alpha_j+2^{-m}\right) \prod_{j=r+1}^{l-1} \left(-\prod_{i=1}^{m-1} r_i(x+\alpha_j) \right) ;\, d_m(x)=1 \right]$$

$$+\mathbf{E}\left[\prod_{i=1}^{m-1} r_i(x) \prod_{j=1}^{r} \left(-\prod_{i=1}^{m-1} r_i\left(x+\alpha_j-2^{-m}\right) \right) \prod_{j=r+1}^{l-1} \prod_{i=1}^{m-1} r_i(x+\alpha_j) ;\, d_m(x)=0 \right].$$

それぞれの被積分関数は与えられた事象 $\{d_m(x)=\epsilon\}$ ($\epsilon=0$ または 1) とは独立だから,

$$= \frac{1}{2}\mathbf{E}\left[\prod_{i=1}^{m-1} r_i(x) \prod_{j=1}^{l-1} \prod_{i=1}^{m-1} r_i(x+\alpha_j^{(m-1)U}) \times (-1)^{l-r} \right]$$

$$+\frac{1}{2}\mathbf{E}\left[\prod_{i=1}^{m-1} r_i(x) \prod_{j=1}^{l-1} \prod_{i=1}^{m-1} r_i(x+\alpha_j^{(m-1)L}) \times (-1)^{r} \right].$$

となることが分かる．さて，r が偶数であれば $l-r$ も偶数だから,

$$A(\boldsymbol{\alpha}) = \frac{1}{2}A(\boldsymbol{\alpha}^{(m-1)U}) + \frac{1}{2}A(\boldsymbol{\alpha}^{(m-1)L}),$$

一方 r が奇数であれば $l-r$ も奇数だから,

$$A(\boldsymbol{\alpha}) = -\frac{1}{2}A(\boldsymbol{\alpha}^{(m-1)U}) - \frac{1}{2}A(\boldsymbol{\alpha}^{(m-1)L}).$$

これで証明が終了する． □

補題 3 によって各 $\boldsymbol{\alpha} \in D^{l-1}$ に対する $A(\boldsymbol{\alpha})$ の値は具体的に計算することができる．

定理 11′ の証明.

$$C_j := \bigcup_{s=1}^{2^m} \left[\frac{s}{2^m} - \frac{\beta^{(m)}_{\sigma(m,j)}}{2^m},\ \frac{s}{2^m} - \frac{\beta^{(m)}_{\sigma(m,j+1)}}{2^m} \right), \quad j=0,1,\ldots,l-1,$$

とおけば，$i=1,\ldots,m$ に対して

$$x \in C_j \implies d_i(x+\alpha_{\sigma(m,p)}) = \begin{cases} d_i\left(x+\alpha^{(m)U}_{\sigma(m,p)}\right), & 1 \le p \le j, \\ d_i\left(x+\alpha^{(m)L}_{\sigma(m,p)}\right), & j+1 \le p \le l-1, \end{cases}$$

となるから，次が成り立つ.

$$E^{(m)}(0,k_1,\ldots,k_{l-1};\alpha)$$

$$= \sum_{j=0}^{l-1} \mathbf{E}\left[\prod_{i=1}^{m} \left(r_i(x) \prod_{p=1}^{j} r_i(x+\alpha^{(m)U}_{\sigma(m,p)}) \prod_{p=j+1}^{l-1} r_i(x+\alpha^{(m)L}_{\sigma(m,p)}) \right) ;\, C_j \right]$$

ここで被積分関数は C_j と独立だから

$$= \sum_{j=0}^{l-1} \mathbb{P}(C_j)\mathbf{E}\left[\prod_{i=1}^{m}\left(r_i(x)\prod_{p=1}^{j} r_i(x+\alpha_{\sigma(m,p)}^{(m)U})\prod_{p=j+1}^{l-1} r_i(x+\alpha_{\sigma(m,p)}^{(m)L})\right)\right]$$

$$= \sum_{j=0}^{l-1} \mathbb{P}(C_j)A(\boldsymbol{\alpha}^{(m),j}).$$

定理 11′ の前半部分は $\mathbb{P}(C_j) = \beta_{\sigma(m,j)}^{(m)} - \beta_{\sigma(m,j+1)}^{(m)}$ であることを用いれば証明が終わる．

定理 11′ の後半部分は補題 3 に訴えて示すことができる．そのためには

$$\left(\boldsymbol{\alpha}^{(m),s}\right)^{(m-1)U} = \boldsymbol{\alpha}^{(m-1),s_1+s_2}, \tag{46}$$

$$\left(\boldsymbol{\alpha}^{(m),s}\right)^{(m-1)L} = \boldsymbol{\alpha}^{(m-1),s_2}, \tag{47}$$

を示せばよい．

はじめに (47) を示そう．示すべきことは,

$$\#\left\{ j \,\middle|\, \left(\alpha_j^{(m),s}\right)^{(m-1)L} = \alpha_j^{(m-1)U} \right\} = s_2 \tag{48}$$

である．次の四つの implications を確認しよう．

$$\begin{aligned} \alpha_j^{(m),s} \neq \alpha_j^{(m)U} &\implies \alpha_j^{(m),s} = \alpha_j^{(m)L} \\ &\implies \left(\alpha_j^{(m),s}\right)^{(m-1)L} = \alpha_j^{(m-1)L} \\ &\implies \left(\alpha_j^{(m),s}\right)^{(m-1)L} \neq \alpha_j^{(m-1)U}, \end{aligned} \tag{49}$$

$$\alpha_j^{(m),s} = \alpha_j^{(m)U} \implies \left(\alpha_j^{(m),s}\right)^{(m-1)U} = \alpha_j^{(m-1)U}, \tag{50}$$

$$\begin{aligned} d_m(\alpha_j^{(m),s}) = 1 &\iff \left(\alpha_j^{(m),s}\right)^{(m-1)L} \neq \left(\alpha_j^{(m),s}\right)^{(m-1)U} \\ &\implies \left(\alpha_j^{(m),s}\right)^{(m-1)L} \neq \alpha_j^{(m-1)U}, \end{aligned} \tag{51}$$

$$d_m(\alpha_j^{(m),s}) = 0 \iff \left(\alpha_j^{(m),s}\right)^{(m-1)L} = \left(\alpha_j^{(m),s}\right)^{(m-1)U}. \tag{52}$$

(49) と (51) の対偶をとって

$$\left(\alpha_j^{(m),s}\right)^{(m-1)L} = \alpha_j^{(m-1)U} \implies \begin{cases} \alpha_j^{(m),s} = \alpha_j^{(m)U} \\ \quad \text{かつ} \\ d_m(\alpha_j^{(m),s}) = 0 \end{cases}$$

が分かる．この逆も (50) と (52) より成り立つことが分かり，

$$\left(\alpha_j^{(m),s}\right)^{(m-1)L} = \alpha_j^{(m-1)U} \iff \begin{cases} \alpha_j^{(m),s} = \alpha_j^{(m)U} \\ \quad \text{かつ} \\ d_m(\alpha_j^{(m),s}) = 0 \end{cases}$$

従って

$$\begin{aligned} \left\{ j \,\middle|\, \left(\alpha_j^{(m),s}\right)^{(m-1)L} = \alpha_j^{(m-1)U} \right\} &= \left\{ j \,\middle|\, \alpha_j^{(m),s} = \alpha_j^{(m)U},\ d_m(\alpha_j^{(m),s}) = 0 \right\} \\ &= \left\{ j \,\middle|\, \alpha_{\sigma(m,j)}^{(m),s} = \alpha_{\sigma(m,j)}^{(m)U},\ d_m(\alpha_{\sigma(m,j)}^{(m),s}) = 0 \right\} \\ &= \left\{ j \,\middle|\, 1 \le j \le s,\ d_m(\alpha_{\sigma(m,j)}^{(m),s}) = d_m(\alpha_{\sigma(m,j)}^{(m)U}) = 0 \right\} \\ &= \left\{ j \,\middle|\, 1 \le j \le s,\ d_m(\alpha_{\sigma(m,j)}^{(m)L}) = d_m(\alpha_{\sigma(m,j)}) = 1 \right\}. \end{aligned}$$

最後の集合の要素の個数は s_2 に等しい．従って (48) が示された．

では (46) を示そう．示すべきことは，

$$\#\left\{ j \,\middle|\, \left(\alpha_j^{(m),s}\right)^{(m-1)U} = \alpha_j^{(m-1)U} \right\} = \#\left\{ j \,\middle|\, \left(\alpha_j^{(m),s}\right)^{(m-1)U} \neq \alpha_j^{(m-1)L} \right\} = s_1 + s_2 \qquad (53)$$

である．明らかに，

$$\left(\alpha_j^{(m),s}\right)^{(m-1)U} \neq \alpha_j^{(m-1)L} \iff \begin{cases} \left(\alpha_j^{(m),s}\right)^{(m-1)U} \neq \left(\alpha_j^{(m),s}\right)^{(m-1)L} = \alpha_j^{(m-1)L} \\ \quad \text{または} \\ \left(\alpha_j^{(m),s}\right)^{(m-1)U} = \left(\alpha_j^{(m),s}\right)^{(m-1)L} \neq \alpha_j^{(m-1)L} \end{cases}$$

$$\iff \begin{cases} \left(\alpha_j^{(m),s}\right)^{(m-1)U} \neq \left(\alpha_j^{(m),s}\right)^{(m-1)L} \\ \quad \text{または} \\ \left(\alpha_j^{(m),s}\right)^{(m-1)L} \neq \alpha_j^{(m-1)L} \end{cases}$$

ところが，(52) より，

$$\left\{ j \,\middle|\, \left(\alpha_j^{(m),s}\right)^{(m-1)U} \neq \left(\alpha_j^{(m),s}\right)^{(m-1)L} \right\} = \left\{ j \,\middle|\, d_m(\alpha_j^{(m),s}) = 1 \right\},$$

であるからこの集合の要素の個数は s_1 に等しい．一方，(48) より

$$\#\left\{ j \,\middle|\, \left(\alpha_j^{(m),s}\right)^{(m-1)L} \neq \alpha_j^{(m-1)L} \right\} = \#\left\{ j \,\middle|\, \left(\alpha_j^{(m),s}\right)^{(m-1)L} = \alpha_j^{(m-1)U} \right\} = s_2,$$

なので，結局，(53) が従う． □

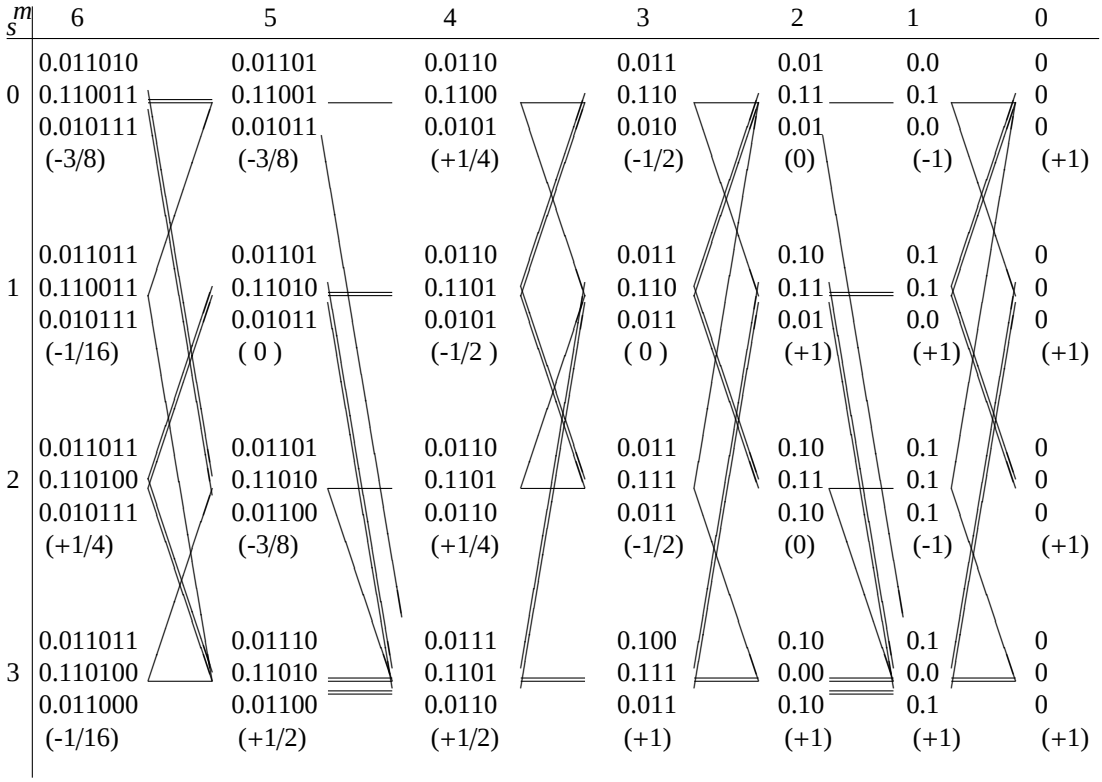
3.3.3 定理 13 の証明

定理 13 の証明は補題 1′ に基づいて次を示すことに帰着される：$l \in \mathbb{N}$ を任意の偶数とし，任意の l 個の自然数 $k_1 < \cdots < k_{l-1}$ に対して

$$\lim_{m\to\infty} E^{(m)}(0, k_1, \ldots, k_{l-1}\,;\,\alpha) \;=\; 0$$

が任意の無理数 α に対して成り立つ.

図 1: **補題 3 のダイアグラム.**



証明に入る前に証明のアイデアを述べておく．まず，定理 11′ のアルゴリズムを具体的な場合に書き下してみよう．簡単のため $l = 4$ とし，$\boldsymbol{\alpha} = (\alpha_1, \alpha_2, \alpha_3)$ が次のように与えられているとする.

$$\begin{cases} \alpha_1 &=& 0.011010110\ldots, \\ \alpha_2 &=& 0.110011101\ldots, \\ \alpha_3 &=& 0.010111011\ldots. \end{cases}$$

ただし，これらは 2 進小数として書かれている．このとき，$m = 6$ として

$$\begin{cases} \alpha_1^{(6)L} &=& 0.011010, & \alpha_1^{(6)U} &=& 0.011011, \\ \alpha_2^{(6)L} &=& 0.110011, & \alpha_2^{(6)U} &=& 0.110100, \\ \alpha_3^{(6)L} &=& 0.010111, & \alpha_3^{(6)U} &=& 0.011000. \end{cases}$$

さらに

$$1 > \beta_1^{(6)} = 0.110\ldots > \beta_2^{(6)} = 0.101\ldots > \beta_3^{(6)} = 0.011\ldots > 0$$

であるから，今の場合，$\sigma(6, j) = j$，$j = 1, 2, 3$，である．これらより，次のようになる (ここでは縦ベクトルで書く).

$$\boldsymbol{\alpha}^{(6),j} = \begin{pmatrix} 0.011010 \\ 0.110011 \\ 0.010111 \end{pmatrix}, \begin{pmatrix} 0.011011 \\ 0.110011 \\ 0.010111 \end{pmatrix}, \begin{pmatrix} 0.011011 \\ 0.110100 \\ 0.010111 \end{pmatrix}, \begin{pmatrix} 0.011011 \\ 0.110100 \\ 0.011000 \end{pmatrix}, \quad j = 0, 1, 2, 3.$$

さて，定理 11′ および補題 3 を忠実に図化すると図 1 のようなダイヤグラムができる．これについて説明しよう．

図 1 のダイヤグラムの右から m 番目 ($m = 0, \ldots, 6$)，上から s 番目 ($s = 0, \ldots, 3$) のベクトルは $\boldsymbol{\alpha}^{(m),s}$ を表す．たとえば $\boldsymbol{\alpha}^{(6),0}$ は左上角のベクトルである．各ベクトルのすぐ下の ( ) の中の数は，関数 $A(\boldsymbol{\alpha}^{(m),s})$ の値を表す．補題 3 によれば $A(\boldsymbol{\alpha}^{(m),s})$ は $m - 1$ の二つの同様の量によって (45) のように書き表されるが，そのとき (45) の符号 $(-1)^r$ が正のときは二重線で，負のときは一重線で記した．このダイヤグラムから，たとえば

$$A(\boldsymbol{\alpha}^{(6),2}) = \frac{1}{2}\left(A(\boldsymbol{\alpha}^{(5),1}) + A(\boldsymbol{\alpha}^{(5),3})\right) \tag{54}$$

が分かる．

定理 13 を証明するために，$m \to \infty$ のとき，$|A(\boldsymbol{\alpha}^{(m),s})| \to 0$ となることを示す．その基本的なアイデアを，このダイヤグラムを用いて説明する．

たとえば (54) で挙げた $A(\boldsymbol{\alpha}^{(6),2})$ の「先祖」をダイヤグラム上で辿ってみる．$m = 2$ まで戻るとすると，取り得るルートは全部で $2^{6-2} = 2^4$ 個ある．そのうち，図 2 で示すような二つのルートでキャンセルが起こっていることが分かる．すなわち，図 2 の下側のルートでは $A(\boldsymbol{\alpha}^{(2),1})$ が $A(\boldsymbol{\alpha}^{(6),2})$ の計算に $2^{-4}A(\boldsymbol{\alpha}^{(2),1})$ だけ関与しているのに対し，上側のルートでは逆に $-2^{-4}A(\boldsymbol{\alpha}^{(2),1})$ だけ関与していて，それらが打ち消し合っているのである．このことから

$$|A(\boldsymbol{\alpha}^{(6),2})| \le \left(1 - \frac{1}{2^4} \times 2\right) \times \max_{s=0,\ldots,3} |A(\boldsymbol{\alpha}^{(2),s})|$$

が分かる．

定理 13 の証明は，このようにキャンセルするルートを無限個見つけ出し，上の評価を繰り返し用いて，$m \to \infty$ のとき，$|A(\boldsymbol{\alpha}^{(m),s})| \to 0$ を示す (補題 8 を見よ) ことにより達成される．

定理 13 を証明するためにいくつかの補題を準備する．

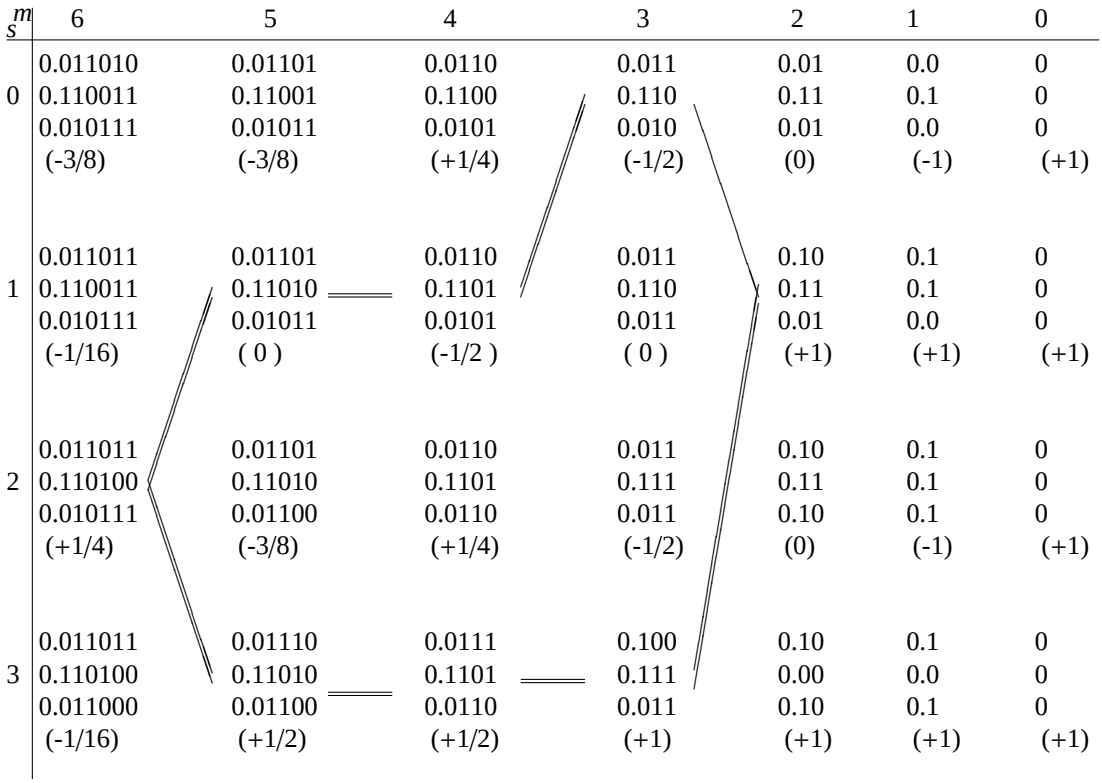
補題 4 $\boldsymbol{\alpha} = (\alpha_1, \ldots, \alpha_{l-1}) \in ([0, 1) \setminus D)^{l-1}$ とする．$m \ge 1$ のとき

(i) $d_m(\alpha_j^{(m)U}) = d_m(\alpha_j^{(m),l-1}) \ne d_m(\alpha_j^{(m),0}) = d_m(\alpha_j^{(m)L}) = d_m(\alpha_j), \quad \forall j.$

(ii) $\left(\boldsymbol{\alpha}^{(m),0}\right)^{(m-1)L} = \boldsymbol{\alpha}^{(m-1),0}.$

(iii) $\left(\boldsymbol{\alpha}^{(m),l-1}\right)^{(m-1)U} = \boldsymbol{\alpha}^{(m-1),l-1}.$

図 2: **キャンセルする二つのルート.**



(iv) $\displaystyle\sum_{j=1}^{l-1} d_m(\alpha_j^{(m),0}) \neq \sum_{j=1}^{l-1} d_m(\alpha_j^{(m),l-1}) \pmod 2.$

(v) $\left(\boldsymbol{\alpha}^{(m),l-1}\right)^{(m-1)L} = \left(\boldsymbol{\alpha}^{(m),0}\right)^{(m-1)U}.$

証明. (i) 次の関係より明らか.

$$\begin{cases} \alpha_j^{(m),l-1} = \alpha_j^{(m)U}, \\ \alpha_j^{(m)U} \neq \alpha_j^{(m)L}, \\ \alpha_j^{(m),0} = \alpha_j^{(m)L}, \quad d_m(\alpha_j) = d_m(\alpha_j^{(m)L}). \end{cases}$$

(ii) s_2 の定義 (29) より, $s = 0$ ならば $s_2 = 0$. これは (ii) を示している.

(iii) (i) より, $s = l - 1$ のとき

$$s_1 = \sum_{j=1}^{l-1} d_m(\alpha_j^{(m),l-1}) = \sum_{j=1}^{l-1} \left(1 - d_m(\alpha_j)\right) = l - 1 - \sum_{j=1}^{l-1} d_m(\alpha_j).$$

一方

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

従って，$s = l-1$ のとき $s_1 + s_2 = l-1$，これより (iii) の主張が従う．

(iv) (i) より

$$\begin{aligned}\sum_{j=1}^{l-1} d_m(\alpha_j^{(m),0}) &= \sum_{j=1}^{l-1}\left(1 - d_m(\alpha_j^{(m),l-1})\right)\\ &= l-1-\sum_{j=1}^{l-1} d_m(\alpha_j^{(m),l-1}) \end{aligned} \tag{55}$$

$l-1$ が奇数であることから (iv) の主張が分かる．

(v) $p, q \in \mathbb{N}$ を

$$\begin{cases} \left(\boldsymbol{\alpha}^{(m),l-1}\right)^{(m-1)L} &= \boldsymbol{\alpha}^{(m-1),p},\\ \left(\boldsymbol{\alpha}^{(m),0}\right)^{(m-1)U} &= \boldsymbol{\alpha}^{(m-1),q}, \end{cases}$$

を満たすようにとれば，(29) および (i) によって

$$\begin{aligned} p &= \sum_{j=1}^{l-1} d_m(\alpha_{\sigma(m,j)}) = \sum_{j=1}^{l-1} d_m(\alpha_j),\\ q &= \sum_{j=1}^{l-1} d_m(\alpha_j^{(m),0}) = \sum_{j=1}^{l-1} d_m(\alpha_j), \end{aligned}$$

すなわち，$p = q$ である． □

定義 15　記法を簡潔にするため，次のような写像を導入する．

$$\begin{cases} \mathcal{L}: \quad (D_m)^{l-1} \ni \boldsymbol{\alpha} \mapsto \boldsymbol{\alpha}^{(m-1)L} \quad \in (D_{m-1})^{l-1}\\ \mathcal{U}: \quad (D_m)^{l-1} \ni \boldsymbol{\alpha} \mapsto \boldsymbol{\alpha}^{(m-1)U} \quad \in (D_{m-1})^{l-1} \end{cases}$$

さらに，$\mathcal{L}^p$ と $\mathcal{U}^p$ は $(D_m)^{l-1}$ から $(D_{m-p})^{l-1}$ への写像と考える．

補題 5　$\boldsymbol{\alpha} = (\alpha_1, \ldots, \alpha_{l-1}) \in ([0,1) \setminus D)^{l-1}$ であり，$r \in \mathbb{N}$ とする．

(i) $\mathcal{L}^r \boldsymbol{\alpha}^{(m+r),l-1} = \boldsymbol{\alpha}^{(m),0}$ ならば $\forall s$, $\mathcal{L}^r \boldsymbol{\alpha}^{(m+r),s} = \boldsymbol{\alpha}^{(m),0}$.

(ii) $\mathcal{U}^r \boldsymbol{\alpha}^{(m+r),0} = \boldsymbol{\alpha}^{(m),l-1}$ ならば $\forall s$, $\mathcal{U}^r \boldsymbol{\alpha}^{(m+r),s} = \boldsymbol{\alpha}^{(m),l-1}$.

(iii)

$$\forall j = 1, \ldots, l-1, \quad m+1 \le \exists p \le m+r, \quad d_{m+p}(\alpha_j) = 0 \tag{56}$$

ならば $\forall s$, $\mathcal{L}^r \boldsymbol{\alpha}^{(m+r),s} = \boldsymbol{\alpha}^{(m),0}$.

(iv)

$$\forall j = 1, \ldots, l-1, \quad m+1 \le \exists p \le m+r, \quad d_{m+p}(\alpha_j) = 1 \tag{57}$$

ならば $\forall s$, $\mathcal{U}^r \boldsymbol{\alpha}^{(m+r),s} = \boldsymbol{\alpha}^{(m),l-1}$.

証明. (i) (29) の s_2 の定義より，s_2 は s の増加関数である．このことより (i) の結論が従う．
(ii) (53) より，$s_1 + s_2$ は s の増加関数である．このことより (ii) の結論が従う．
(iii) 条件 (56) と補題 4(i) より，各 j に対してある p が存在し，$d_{m+p}(\alpha_j^{(m+r),l-1}) = 1$ である．これより $\alpha_j^{(m+p-1),l-1} > \alpha_j^{(m+p)L}$ である．このことと，補題 4(ii) から $\mathcal{L}^r\boldsymbol{\alpha}^{(m+r),l-1} = \boldsymbol{\alpha}^{(m),0}$ であるから，(i) によって (iii) の結論が従う．
(iv) 条件 (57) より，各 j に対してある p が存在し，$\alpha_j^{(m+p-1),0} < \alpha_j^{(m+p)U}$ である．このことと，補題 4(iii) から $\mathcal{U}^r\boldsymbol{\alpha}^{(m+r),0} = \boldsymbol{\alpha}^{(m),l-1}$ であるから，(ii) によって (iv) の結論が従う． □

補題 6 ([35]) $r := 3k_{l-1}$ とする．任意の無理数 α に対して $\alpha_j := \langle k_j\alpha\rangle$ とするとき，(56) と (57) を同時に満たすような m が無限個存在する．

証明. 背理法による．(56) と (57) の両方を満たす m が有限個しかないと仮定する．すなわち，ある $N \in \mathbb{N}$ が存在して $m > N$ ならば，ある j_m が存在して $d_{m+1}(\alpha_{j_m}) = \ldots = d_{m+r}(\alpha_{j_m}) = 0$ または $d_{m+1}(\alpha_{j_m}) = \ldots = d_{m+r}(\alpha_{j_m}) = 1$ である，と仮定する．このとき，α は有理数であることを示したい．そのためには α の 2 進展開の第 $N+1$ 桁以下が周期的になることを示せばよい．

Step 1. $m > N$ を固定するとき，有限列 $d_{m+1}(\alpha), d_{m+2}(\alpha), \ldots, d_{m+r-k_{j_m}}(\alpha)$ は周期的でその周期は高々 k_{j_m} である，ことを示す．

$k_{j_m}\langle 2^m\alpha\rangle$ を k_{j_m} で割って α の 2 進展開について調べる．まず，$R_1 := \lfloor k_{j_m}\langle 2^m\alpha\rangle\rfloor$ とすれば

$$R_1 + \langle 2^m k_{j_m}\alpha\rangle = \lfloor k_{j_m}\langle 2^m\alpha\rangle\rfloor + \langle k_{j_m}\langle 2^m\alpha\rangle\rangle = k_{j_m}\langle 2^m\alpha\rangle.$$

両辺を 2 倍して

$$2R_1 + d_{m+1}(k_{j_m}\alpha) + \langle 2^{m+1}k_{j_m}\alpha\rangle = k_{j_m}d_{m+1}(\alpha) + k_{j_m}\langle 2^{m+1}\alpha\rangle.$$

さて

$$\begin{aligned} k_{j_m}\langle 2^{m+1}\alpha\rangle - \langle 2^{m+1}k_{j_m}\alpha\rangle &= \lfloor k_{j_m}\langle 2^{m+1}\alpha\rangle\rfloor + \langle k_{j_m}\langle 2^{m+1}\alpha\rangle\rangle - \langle 2^{m+1}k_{j_m}\alpha\rangle \\ &= \lfloor k_{j_m}\langle 2^{m+1}\alpha\rangle\rfloor \end{aligned}$$

だから $0 \le k_{j_m}\langle 2^{m+1}\alpha\rangle - \langle 2^{m+1}k_{j_m}\alpha\rangle < k_{j_m}$ なので，$2R_1 + d_m(k_{j_m}\alpha)$ を k_{j_m} で割ったときの商を Q_1，余りを R_2 とすれば，

$$\begin{aligned} Q_1 &= d_{m+1}(\alpha), \\ R_2 &= k_{j_m}\langle 2^{m+1}\alpha\rangle - \langle 2^{m+1}k_{j_m}\alpha\rangle. \end{aligned}$$

次に $2R_2 + d_{m+1}(k_{j_m}\alpha)$ を k_{j_m} で割ったときの商を Q_2，余りを R_3 とおけば，$R_2 + \langle 2^{m+1}k_{j_m}\alpha\rangle = k_{j_m}\langle 2^{m+1}\alpha\rangle$ なので，上と同様にして

$$\begin{aligned} Q_2 &= d_{m+2}(\alpha), \\ R_3 &= k_{j_m}\langle 2^{m+2}\alpha\rangle - \langle 2^{m+2}k_{j_m}\alpha\rangle. \end{aligned}$$

以下同様にして $2R_u+d_{m-1+u}(k_{j_m}\alpha)=k_{j_m}Q_u+R_{u+1}$，$0\le R_{u+1}<k_{j_m}$，を満たすように (Q_u,R_{u+1}) をとる．このとき，仮定より $d_{m+1}(k_{j_m}\alpha)=\ldots=d_{m+r}(k_{j_m}\alpha)$ であるから，(Q_u,R_{u+1}) は R_u にのみ依存し，R_u の取り得る値が高々 k_{j_m} 個だから，この列は周期的でその周期は高々 k_{j_m} である．とくに $Q_u=d_{m+u}(\alpha)$ は周期高々 k_{j_m} なる周期列である．

Step 2. 一般に，数列 $a(0),a(1),\ldots,a(p-1)$ の最小周期を w とする．ただし $2k\le p$ とする．もし，部分数列 $a(q),a(q+1),\ldots,a(q+p'-1)$，$0\le q<q+p'\le p$，の長さ p' が $2w$ 以上ならば，この部分数列の最小周期は w に等しい，ことを示す．

$a(q),a(q+1),\ldots,a(q+p'-1)$ の最小周期を w' とすれば明らかに $w'\le w$．任意の $1\le u\le p-w'$ に対して $u-q=wj+v$，$j\in\mathbb{Z}$，$0\le v<w$，とできるので，$v+w'<2k\le p'$ より

$$a(u)=a(q+wj+v)=a(q+v)=a(q+v+w')=a(q+wj+v+w')=a(u+w').$$

よって w' はもとの数列 $a(0),a(1),\ldots,a(p-1)$ の周期となり，w の最小性から $w'=w$ である．

Step 3. 補題の主張を示す．$m>N$ とする．Step 1 より，$d_{m+1}(\alpha),\ldots,d_{m+r-k_{j_m}}(\alpha)$ は周期高々 k_{j_m} なる周期列である．とくに $d_{m+1}(\alpha),\ldots,d_{m+r-k_{l-1}}(\alpha)$ も周期的でその最小周期を w_m とする．同様に $d_{m+2}(\alpha),\ldots,d_{m+r+1-k_{l-1}}(\alpha)$ も周期的でその最小周期は w_{m-1} である．Step 2 により，$w_m=w_{m-1}$ でこれは $d_{m+1}(\alpha),\ldots,d_{m+r+1-k_{l-1}}(\alpha)$ の最小周期 w に等しい．この操作を続ければ α の小数点以下 $N+1$ 桁目以下は周期 w で循環することが分かる． □

従属性消滅定理の証明を続けよう．補題 1′ によって，定理 13 を証明するために我々のすべきことは，任意の偶数 l と任意の $1\le k_1<\cdots<k_{l-1}$ に対して

$$\lim_{m\to\infty}E^{(m)}(0,k_1,\ldots,k_{l-1};\alpha)\ =\ 0 \tag{58}$$

を示すことであった．α を無理数とし，再び

$$\boldsymbol{\alpha}:=(\alpha_1,\ldots,\alpha_{l-1}),\qquad \alpha_j:=\langle k_j\alpha\rangle, \tag{59}$$

とおく．このとき，定理 11′ によれば (58) を示すには

$$\lim_{m\to\infty}\max_{0\le s\le l-1}\left|A(\boldsymbol{\alpha}^{(m),s})\right|\ =\ 0 \tag{60}$$

を示せばよい．

補題 3 によって

$$A(\boldsymbol{\alpha}^{(m),s})\ =\ \pm\frac{1}{2}\left\{A(\mathcal{U}\boldsymbol{\alpha}^{(m),s})+A(\mathcal{L}\boldsymbol{\alpha}^{(m),s})\right\} \tag{61}$$

である．次の補題は (61) から直ちに導かれる．

補題 7 $\max_{1\le q\le l-1}|A(\boldsymbol{\alpha}^{(m'),q})|\ \le\ \max_{1\le q\le l-1}|A(\boldsymbol{\alpha}^{(m),q})|,\qquad m'>m.$

そこで次の鍵になる補題が示される．

補題 8　α を無理数，$r = 3k_{l-1}$ とする．$\{m_n\}_{n=0}^{\infty}$ を補題 6 で述べられた無限個の $m \geq 2$ のうち $m_n + r + 2 \leq m_{n+1}$ となるものをとって並べたものとする．このとき

$$\max_{1\leq s\leq l-1}\left|A(\boldsymbol{\alpha}^{(m_n+r),s})\right| \leq \left(1-\frac{1}{2^{r+1}}\right)\max_{1\leq s\leq l-1}\left|A(\boldsymbol{\alpha}^{(m_n-2),s})\right|.$$

証明. (61) を r 回用いれば次の表現を得る．

$$\begin{aligned} A(\boldsymbol{\alpha}^{(m_n+r),s}) &= \frac{1}{2^r}\epsilon_{\mathcal{U}^r}A(\mathcal{U}^r\boldsymbol{\alpha}^{(m_n+r),s}) + \frac{1}{2^r}\epsilon_{\mathcal{L}U^{r-1}}A(\mathcal{L}U^{r-1}\boldsymbol{\alpha}^{(m_n+r),s}) + \\ &\quad\vdots \\ &\quad + \frac{1}{2^r}\epsilon_{\mathcal{U}L^{r-1}}A(\mathcal{U}L^{r-1}\boldsymbol{\alpha}^{(m_n+r),s}) + \frac{1}{2^r}\epsilon_{\mathcal{L}^r}A(\mathcal{L}^r\boldsymbol{\alpha}^{(m_n+r),s}), \end{aligned} \tag{62}$$

ここで $\epsilon_{\mathcal{U}^r},\ldots,\epsilon_{\mathcal{L}^r} = \pm 1$ である．補題 5 によって，$\forall s$ に対して

$$\mathcal{U}^r\boldsymbol{\alpha}^{(m_n+r),s} = \boldsymbol{\alpha}^{(m_n),l-1}, \qquad \mathcal{L}^r\boldsymbol{\alpha}^{(m_n+r),s} = \boldsymbol{\alpha}^{(m_n),0}, \tag{63}$$

である．

Case 1. まず，$\epsilon_{\mathcal{U}^r} = \epsilon_{\mathcal{L}^r}$ の場合．$\epsilon, \epsilon' = \pm 1$ として (61) より

$$\begin{cases} \epsilon_{\mathcal{U}^r}A(\mathcal{U}^r\boldsymbol{\alpha}^{(m_n+r),s}) &= \epsilon_{\mathcal{U}^r}\epsilon\left\{\frac{1}{2}A(\mathcal{U}^{r+1}\boldsymbol{\alpha}^{(m_n+r),s}) + \frac{1}{2}A(\mathcal{L}U^r\boldsymbol{\alpha}^{(m_n+r),s})\right\}, \\ \epsilon_{\mathcal{L}^r}A(\mathcal{L}^r\boldsymbol{\alpha}^{(m_n+r),s}) &= \epsilon_{\mathcal{L}^r}\epsilon'\left\{\frac{1}{2}A(\mathcal{U}L^r\boldsymbol{\alpha}^{(m_n+r),s}) + \frac{1}{2}A(\mathcal{L}^{r+1}\boldsymbol{\alpha}^{(m_n+r),s})\right\}. \end{cases} \tag{64}$$

補題 3 と補題 4(iv) によって，$\epsilon \neq \epsilon'$ である．(63) と補題 4(v) から

$$\mathcal{L}U^r\boldsymbol{\alpha}^{(m_n+r),s} = \mathcal{U}L^r\boldsymbol{\alpha}^{(m_n+r),s} \tag{65}$$

が従うことに注意せよ． さて，(64) を用いて (62) をもう一度展開すれば

$$\begin{aligned} A(\boldsymbol{\alpha}^{(m_n+r),s}) &= \frac{1}{2^{r+1}}\epsilon_{\mathcal{U}^r}\epsilon A(\mathcal{U}^{r+1}\boldsymbol{\alpha}^{(m_n+r),s}) + \frac{1}{2^{r+1}}\epsilon_{\mathcal{U}^r}\epsilon A(\mathcal{L}U^r\boldsymbol{\alpha}^{(m_n+r),s}) + \\ &\quad\vdots \\ &\quad + \frac{1}{2^{r+1}}\epsilon_{\mathcal{L}^r}\epsilon' A(\mathcal{U}L^r\boldsymbol{\alpha}^{(m_n+r),s}) + \frac{1}{2^{r+1}}\epsilon_{\mathcal{L}^r}\epsilon' A(\mathcal{L}^{r+1}\boldsymbol{\alpha}^{(m_n+r),s}) \end{aligned} \tag{66}$$

が得られる．いま，$\epsilon_{\mathcal{U}^r}\epsilon \neq \epsilon_{\mathcal{L}^r}\epsilon'$ だから (65) より

$$\frac{1}{2^{r+1}}\epsilon_{\mathcal{U}^r}\epsilon A(\mathcal{L}U^r\boldsymbol{\alpha}^{(m_n+r),s}) + \frac{1}{2^{r+1}}\epsilon_{\mathcal{L}^r}\epsilon' A(\mathcal{U}L^r\boldsymbol{\alpha}^{(m_n+r),s}) = 0. \tag{67}$$

従って次の評価が得られる．

$$\left|A(\boldsymbol{\alpha}^{(m_n+r),s})\right| \leq \left(1-\frac{1}{2^{r+1}}\times 2\right)\max_{0\leq q\leq l-1}\left|A(\boldsymbol{\alpha}^{(m_n-1),q})\right|. \tag{68}$$

Case 2. $\epsilon_{\mathcal{U}^r} \neq \epsilon_{\mathcal{L}^r}$ の場合．このときは，展開 (66) において $\epsilon_{\mathcal{U}^r}\epsilon = \epsilon_{\mathcal{L}^r}\epsilon'$ だから，(62) の代わりに (66) から出発して Case 1 と同じ方法で，評価

$$\left|A(\boldsymbol{\alpha}^{(m_n+r),s})\right| \leq \left(1-\frac{1}{2^{r+2}}\times 2\right)\max_{0\leq q\leq l-1}\left|A(\boldsymbol{\alpha}^{(m_n-2),q})\right| \tag{69}$$

を得ることができる．

補題 7 によれば，(68) も評価 (69) を導く．すなわち，いずれの場合にも評価 (69) を得る．□

もはや定理 13 の証明は容易である．自明な評価

$$\max_{0\le s\le l-1}\left|A(\boldsymbol{\alpha}^{(\bullet),s})\right| \le 1,$$

と (69) から，結局

$$\begin{aligned}
\max_{0\le s\le l-1}\left|A(\boldsymbol{\alpha}^{(m_n+r),s})\right| &\le \left(1-\frac{1}{2^{r+1}}\right)\max_{0\le s\le l-1}\left|A(\boldsymbol{\alpha}^{(m_{n-1}),s})\right| \\
&\le \left(1-\frac{1}{2^{r+1}}\right)^2\max_{0\le s\le l-1}\left|A(\boldsymbol{\alpha}^{(m_{n-2}),s})\right| \\
&\le \cdots\cdots\cdots \\
&\le \left(1-\frac{1}{2^{r+1}}\right)^n\max_{0\le s\le l-1}\left|A(\boldsymbol{\alpha}^{(m_0),s})\right| \\
&\le \left(1-\frac{1}{2^{r+1}}\right)^n \longrightarrow 0, \qquad \text{as } n\to\infty. \qquad (70)
\end{aligned}$$

が得られる．これを補題 7 と合わせれば，(60) が分かる．□

注意 9　大雑把にいって，ほとんどすべての α に対して $\{m_n\}_{n=0}^{\infty}$ はほぼ「等差数列」であろうから，最後の評価 (70) は，ほとんどすべての α に対して収束 (60) が指数的に早いことを意味している (定理 10′ 参照).

注意 10　$l \ge 4$ に対して上の証明で述べたキャンセル (67) は非常に特殊なもので，展開 (62) においてはほかにも様々なキャンセルが起こっているのが普通である．ただし，$l = 2$ のときは上の証明で述べたキャンセルが起こり得るすべてのキャンセルである．

3.3.4　定理 10 の証明

エルゴード理論を用いて定理 10 を示す．ただし，ここでは証明するのは，$\{Y_n^{(m)}\}_{n=0}^{\infty}$ ではなく，それと同等の $\{X_n^{(m)}\}_{n=0}^{\infty}$ に対応する次の定理である．

定理 10′　$l \in \mathbb{N}$ を偶数とし，任意に $0 < k_1 < \cdots < k_{l-1} \in \mathbb{N}$ をとる．このとき，ある $0 < \rho < 1$ が存在して，ほとんどすべての無理数 α に対して

$$\mathbf{E}\left[X_0^{(m)}(\bullet;\alpha)X_{k_1}^{(m)}(\bullet;\alpha)\times\cdots\times X_{k_{l-1}}^{(m)}(\bullet;\alpha)\right] = o(\rho^m), \quad \text{as } m\to\infty. \qquad (71)$$

群拡大による定式化

2 次元トーラス上の関数 f を

$$f(x,\alpha) := r_1(x)r_1(x+k_1\alpha)\times\cdots\times r_1(x+k_{l-1}\alpha), \qquad (x,\alpha)\in\mathbb{T}^2, \tag{72}$$

と定義する．ここで $r_1(x)$ はラデマッハ関数列の最初の関数である．このとき

$$\mathbf{E}\left[X_0^{(m)}(\bullet;\alpha)X_{k_1}^{(m)}(\bullet;\alpha)\times\cdots\times X_{k_{l-1}}^{(m)}(\bullet;\alpha)\right] = \int_{\mathbb{T}^1} dx\prod_{i=1}^{m} f(2^{i-1}x,2^{i-1}\alpha)$$

である．

3 次元トーラス上の 2 進変換 $\beta:\mathbb{T}^3\to\mathbb{T}^3$

$$\beta(x,y,\alpha) := (2x,2y,2\alpha) \tag{73}$$

の群拡大 (group extension または skew product) を次のように定義する．

定義 16　$\Omega := \mathbb{T}^3\times\{-1,1\}^2$，$\mu$ を Ω 上の一様確率測度，すなわち

$$\mu := \mathbb{P}^3\otimes\frac{\delta_{-1}+\delta_1}{2}\otimes\frac{\delta_{-1}+\delta_1}{2} \tag{74}$$

とする．変換 $T_f:\Omega\to\Omega$ を

$$T_f(x,y,\alpha,\epsilon_1,\epsilon_2) := (2x,2y,2\alpha,\epsilon_1 f(x,\alpha),\epsilon_2 f(y,\alpha)) \tag{75}$$

で定義する．[40]

明らかに T_f は μ を保存する．$\mathbb{T}^3$ の部分集合 C を

$$C := \{(x,y,\alpha)\,|\,(x,y,\alpha)\text{ は }f(x,\alpha)\text{ または }f(y,\alpha)\text{ の不連続点}\} \tag{76}$$

とすれば，明らかに C は測度 0 の β-不変集合である．$\mathbb{T}^3-C$ の各連結成分を E_j，$j=1,\ldots,J$，とする．$\Omega\supset F_j$ を次のようにとる．

$$\begin{aligned}\{F_j\}_{j=1}^{4J} := {} & \left\{E_j\times\{-1\}\times\{-1\}\right\}_{j=1}^{J}\bigcup\left\{E_j\times\{-1\}\times\{1\}\right\}_{j=1}^{J}\\ & \bigcup\left\{E_j\times\{1\}\times\{-1\}\right\}_{j=1}^{J}\bigcup\left\{E_j\times\{1\}\times\{1\}\right\}_{j=1}^{J}\end{aligned}$$

このとき，$\Omega=\bigcup_{j=1}^{4J}F_j$, μ-a.e.

定義 17　(Ω,μ) 上で定義された $\{1,2,3,\ldots,4J\}$-値確率過程 $\{\zeta_m\}_{m=0}^{\infty}$ を次で定義する．

$$\zeta_m(x,y,\alpha,\epsilon_1,\epsilon_2) = j \quad :\Longleftrightarrow \quad T_f^m(x,y,\alpha,\epsilon_1,\epsilon_2)\in F_j.$$

[40] ここで紹介する群拡大の方法では安富のアイデア [32, 33] を参考にした．なお，先駆的仕事として高信 [28] では，$\mathbb{T}^2\times\{-1,1\}$ 上の力学系 $T:(x,\alpha,\epsilon)\mapsto(2x,2\alpha,\epsilon f(x,\alpha))$ の強混合性を一様確率測度の下で示している．

補題 9　$\{\zeta_m\}_{m=1}^{\infty}$ は

$$p(i,j) := \mu\left(T_f^{-1}(F_j)\,\middle|\, F_i\right), \quad i,j = 1,\dots,4J,$$

を推移確率行列とする既約で非周期的な定常マルコフ連鎖である．[41]

$p^m(i,j) := \mu(\zeta_m = j \mid \zeta_0 = i)$ とおく．すると，補題 9 より直ちに次の系を得る (cf. [2] Theorem 8.9).

系 1　任意の $i,j = 1,\dots,4J$ に対して

$$p^m(i,j) \longrightarrow \mu(F_j), \quad \text{as } m \to \infty,$$

が成り立ち，しかもこの収束は指数関数的である．

補題 9 は後に示す．補題 9 によって定理 10′ は以下のようにして示される．まず，次の 4 つの写像を定義する．: $i = 1,2$ に対して

$$\begin{aligned} &\Phi_i : \Omega \to \{-1,1\}, && \Phi_i(x,y,\alpha,\epsilon_1,\epsilon_2) := \epsilon_i, \\ &\widetilde{\Phi}_i : \{1,\dots,4J\} \to \{-1,1\}, && \widetilde{\Phi}_i(j) := \Phi_i(F_j) = F_j \text{ の } \epsilon_i\text{-成分}. \end{aligned}$$

このとき

$$\begin{aligned} &\int_{\mathbb{T}^1} dx \prod_{i=1}^{m} f(2^{i-1}x, 2^{i-1}\alpha) \\ &\quad = \int_{\mathbb{T}^1} dx \Phi_1(x,y,\alpha,\epsilon_1,\epsilon_2)\Phi_1(T_f^m(x,y,\alpha,\epsilon_1,\epsilon_2)) \\ &\quad = \int_{\mathbb{T}^1} dx \widetilde{\Phi}_1(\zeta_0(x,y,\alpha,\epsilon_1,\epsilon_2))\widetilde{\Phi}_1(\zeta_m(x,y,\alpha,\epsilon_1,\epsilon_2)) \end{aligned} \tag{77}$$

と表される．なお，(77) の右辺は y, ϵ_2 によらないことに注意せよ．次のような計算をする．

$$\begin{aligned} &\int_{\mathbb{T}^1} d\alpha \left(\int_{\mathbb{T}^1} dx \widetilde{\Phi}_1(\zeta_0(x,y,\alpha,\epsilon_1,\epsilon_2))\widetilde{\Phi}_1(\zeta_m(x,y,\alpha,\epsilon_1,\epsilon_2)) \right)^2 \\ &\quad = \int_{\mathbb{T}^1} d\alpha \left(\int_{\mathbb{T}^1} dx \widetilde{\Phi}_1(\zeta_0(x,y,\alpha,\epsilon_1,\epsilon_2))\widetilde{\Phi}_1(\zeta_m(x,y,\alpha,\epsilon_1,\epsilon_2)) \right. \\ &\qquad \left. \times \int_{\mathbb{T}^1} dy \widetilde{\Phi}_2(\zeta_0(x,y,\alpha,\epsilon_1,\epsilon_2))\widetilde{\Phi}_2(\zeta_m(x,y,\alpha,\epsilon_1,\epsilon_2)) \right) \end{aligned}$$

α を固定すれば，$\Phi_1(\zeta_m(x,y,\alpha,\epsilon_1,\epsilon_2))$ と $\Phi_2(\zeta_m(x,y,\alpha,\epsilon_1,\epsilon_2))$ は $(x,y,\epsilon_1,\epsilon_2)$ に関する確率変数と見て独立だから，上の値は

$$\begin{aligned} &= \int_{\mathbb{T}^1} d\alpha \left(\int_{\mathbb{T}^2} dxdy \widetilde{\Phi}_1(\zeta_0(x,y,\alpha,\epsilon_1,\epsilon_2))\widetilde{\Phi}_1(\zeta_m(x,y,\alpha,\epsilon_1,\epsilon_2)) \right. \\ &\qquad \left. \times \widetilde{\Phi}_2(\zeta_0(x,y,\alpha,\epsilon_1,\epsilon_2))\widetilde{\Phi}_2(\zeta_m(x,y,\alpha,\epsilon_1,\epsilon_2)) \right) \\ &= \int_{\Omega} d\mu \widetilde{\Phi}_3(\zeta_0(x,y,\alpha,\epsilon_1,\epsilon_2))\widetilde{\Phi}_3(\zeta_m(x,y,\alpha,\epsilon_1,\epsilon_2)) \\ &= \sum_{i,j} \widetilde{\Phi}_3(i)\widetilde{\Phi}_3(j)p^m(i,j)\mu(F_i), \end{aligned}$$

[41] このとき，分割 $\{F_j\}_j$ はマルコフ分割，力学系 (Ω, T_f) はマルコフ変換という．なお，マルコフ連鎖に関する用語 (既約 (irreducible)，非周期的 (aperiodic)，定常 (stationary)) などは，たとえば [2] Section 8 を見よ．

ここで，$\widetilde{\Phi}_3 := \widetilde{\Phi}_1 \times \widetilde{\Phi}_2$ とおいた．系 1 によれば，$m \to \infty$ のとき

$$\sum_{i,j} \widetilde{\Phi}_3(i)\widetilde{\Phi}_3(j)p^m(i,j)\mu(F_i) \longrightarrow \sum_{i,j} \widetilde{\Phi}_3(i)\widetilde{\Phi}_3(j)\mu(F_j)\mu(F_i) = \left(\sum_i \widetilde{\Phi}_3(i)\mu(F_i)\right)^2$$
$$= \left(\int_\Omega \epsilon_1\epsilon_2 d\mu\right)^2 = 0$$

であり，この収束は指数関数的である．よって

$$\sum_{m=1}^{\infty} \int_{\mathbb{T}^1} d\alpha \left(\int_{\mathbb{T}^1} dx \widetilde{\Phi}_1(\zeta_0(x,y,\alpha,\epsilon_1,\epsilon_2))\widetilde{\Phi}_1(\zeta_m(x,y,\alpha,\epsilon_1,\epsilon_2))\right)^2 \tag{78}$$
$$= \sum_{m=1}^{\infty} \sum_{i,j} \widetilde{\Phi}_3(i)\widetilde{\Phi}_3(j)p^m(i,j)\mu(F_i) < \infty$$

だが，(78) の各項が指数減衰であることから，ある $0 < \rho_1 < 1$ が存在して，

$$\sum_{m=1}^{\infty} \rho_1^{-m} \int_{\mathbb{T}^1} d\alpha \left(\int_{\mathbb{T}^1} dx \widetilde{\Phi}_1(\zeta_0(x,y,\alpha,\epsilon_1,\epsilon_2))\widetilde{\Phi}_1(\zeta_m(x,y,\alpha,\epsilon_1,\epsilon_2))\right)^2 < \infty. \tag{79}$$

従って

$$\int_{\mathbb{T}^1} d\alpha \sum_{m=1}^{\infty} \left(\rho_1^{m/2} \int_{\mathbb{T}^1} dx \widetilde{\Phi}_1(\zeta_0(x,y,\alpha,\epsilon_1,\epsilon_2))\widetilde{\Phi}_1(\zeta_m(x,y,\alpha,\epsilon_1,\epsilon_2))\right)^2 < \infty.$$

よって

$$\rho_1^{-m/2} \int_{\mathbb{T}^1} dx \widetilde{\Phi}_1(\zeta_0(x,y,\alpha,\epsilon_1,\epsilon_2))\widetilde{\Phi}_1(\zeta_m(x,y,\alpha,\epsilon_1,\epsilon_2)) \to 0, \quad \text{as } m \to \infty, \qquad \text{a.e.}\alpha. \tag{80}$$

このことは，ほとんどすべての α に対して一様な指数減衰の評価 (71) ができることを示す．

□

マルコフ性の証明

補題 9 の証明を考えよう．T_f が μ を保存するので $\{\zeta_m\}_m$ の定常性は明らかである．定常性より，

$$\mu(\zeta_m = j \,|\, \zeta_{m-1} = i) = p(i,j), \qquad i,j = 1,\dots,4J, \quad m \in \mathbb{N}.$$

以下の証明では

$$T_f^{-m} := (T_f^m)^{-1}, \qquad \beta^{-m} := (\beta^m)^{-1} \tag{81}$$

と略記する．

補題 10 $T_f^{-m}F_j$，$j = 1,\dots,4J$，の連結集合全体のなす Ω の分割を Δ^{-m} とする．このとき，$m' < m$ ならば Δ^{-m} は $\Delta^{-m'}$ の細分，すなわち，任意の $A \in \Delta^{-m}$，$A' \in \Delta^{-m'}$ に対して，$A \subset A'$ または $A \cap A' = \emptyset$ のどちらか一方が成り立つ．

証明. $\tilde{C} := \cup_{\epsilon_1,\epsilon_2=-1,1} C\times\{\epsilon_1\}\times\{\epsilon_2\}$ とおく．$\tilde{C}$ は T_f-不変集合，すなわち，$T_f\tilde{C}\subset\tilde{C}$ である．もし $A\in\Delta^{-m}$ が $A', B'\in\Delta^{-m'}$ $(B'\neq A')$ に対して $A\cap A'\neq\emptyset$ かつ $A\cap B'\neq\emptyset$ ならば $A\cap\tilde{C}^{-m'}\neq\emptyset$ となる．この両辺に T_f^m を作用させれば，$T_f^m\tilde{C}^{-m'} = T_f^{m-m'}\tilde{C}\subset\tilde{C}$ によって

$$T_f^m A\cap\tilde{C}\supset T_f^m A\cap T_f^m\tilde{C}^{-m'}\supset T_f^m(A\cap\tilde{C}^{-m'})\neq\emptyset.$$

しかし，$T_f^m(A)$ はある F_j に他ならないので，これは矛盾である． □

それでは $\{\zeta_m\}_{m=0}^{\infty}$ のマルコフ性を証明しよう．$\mu(\zeta_0=i_0,\dots,\zeta_{m-1}=i_{m-1})>0$ を仮定する．

$$\begin{aligned}
&\mu(\zeta_m=j\,|\,\zeta_0=i_0,\dots,\zeta_{m-1}=i_{m-1})\\
&\quad=\mu\left(T_f^{-m}F_j\,\middle|\,F_{i_0}\cap T_f^{-1}F_{i_1}\cap\cdots\cap T_f^{-m+1}F_{i_{m-1}}\right)\\
&\quad=\frac{\mu\left(F_{i_0}\cap T_f^{-1}F_{i_1}\cap\cdots\cap T_f^{-m+1}F_{i_{m-1}}\cap T_f^{-m}F_{i_m}\right)}{\mu\left(F_{i_0}\cap T_f^{-1}F_{i_1}\cap\cdots\cap T_f^{-m+1}F_{i_{m-1}}\right)}\\
&\quad=\frac{\mu\left(F_{i_0}\cap T_f^{-1}F_{i_1}\cap\cdots\cap T_f^{-m+1}\left(F_{i_{m-1}}\cap T_f^{-1}F_{i_m}\right)\right)}{\mu\left(F_{i_0}\cap T_f^{-1}F_{i_1}\cap\cdots\cap T_f^{-m+1}F_{i_{m-1}}\right)}.
\end{aligned}$$

$T_f^{-m+1}F_{i_{m-1}}$ は 8^{m-1} 個の同測度の連結成分よりなり，補題 10 によれば，そのうちのいくつか，l 個としよう，が $F := F_{i_0}\cap T_f^{-1}F_{i_1}\cap\cdots\cap T_f^{-m+2}F_{m-2}$ にすっぽり含まれ，他の $8^{m-1}-l$ 個は F と共通部分を持たない．この事情は $T_f^{-m+1}\left(F_{i_{m-1}}\cap T_f^{-1}F_{i_m}\right)$ でも同様である．従って

$$\begin{aligned}
\mu\left(F\cap T_f^{-m+1}\left(F_{i_{m-1}}\cap T_f^{-1}F_{i_m}\right)\right) &= \frac{l}{8^{m-1}}\mu\left(T_f^{-m+1}\left(F_{i_{m-1}}\cap T_f^{-1}F_{i_m}\right)\right),\\
\mu\left(F\cap T_f^{-m+1}F_{i_{m-1}}\right) &= \frac{l}{8^{m-1}}\mu\left(T_f^{-m+1}F_{i_{m-1}}\right).
\end{aligned}$$

これより，T_f の μ-不変性を用いて

$$\begin{aligned}
\mu(\zeta_m=j\,|\,\zeta_0=i_0,\dots,\zeta_{m-1}=i_{m-1}) &= \frac{\mu\left(T_f^{-m+1}\left(F_{i_{m-1}}\cap T_f^{-1}F_{i_m}\right)\right)}{\mu\left(T_f^{-m+1}F_{i_{m-1}}\right)}\\
&= \frac{\mu\left(F_{i_{m-1}}\cap T_f^{-1}F_{i_m}\right)}{\mu(F_{i_{m-1}})}\\
&= \mu(\zeta_m=j\,|\,\zeta_{m-1}=i_{m-1})
\end{aligned}$$

が従い，$\{\zeta_m\}_m$ のマルコフ性が分かる． □

エルゴード性の証明

$\{\zeta_m\}_m$ の既約性は T_f のエルゴード性から従う．次の補題から始めよう．

補題 11 $\phi_i:\mathbb{T}^3\to\mathbb{C}$, $i=1,2$, は可測関数で

$$\phi_1(x,y,\alpha) = \phi_1(2x,2y,2\alpha)f(x,\alpha),\qquad \text{a.e.} \tag{82}$$

$$\phi_2(x,y,\alpha) = \phi_2(2x,2y,2\alpha)f(x,\alpha)f(y,\alpha),\qquad \text{a.e.} \tag{83}$$

を満たすとする．このとき，$\phi_1=\phi_2=0$, a.e. である．

証明. 以下 $i = 1, 2$ とする．まず，f は 0 点を持たないことより，$\phi_i(x, y, \alpha)$ の 0 点集合は $\phi_i(2x, 2y, 2\alpha)$ の 0 点集合と一致するので，それは 2 進変換 β に関して不変，従ってその測度はエルゴード性によって 0 か 1 である．その測度が 1 なら証明終わり．それで $\phi_i \neq 0$, a.e.，と仮定しよう．さらに，ϕ_i の実部または虚部の符合がやはり同じ関係式を満たすから，はじめから，$\phi_i \in \{-1, 1\}$ としてよい．

次の領域 $A \subset \mathbb{T}^3$ に注目する．

$$A := \left\{ (x, y, \alpha) \;\middle|\; \begin{array}{lll} \frac{1}{2} < x < 1, & \frac{1}{2} < x + n_{k-2}\alpha < 1, & 1 < x + n_{k-1}\alpha < \frac{3}{2}, \\ \frac{1}{2} < y < 1, & \frac{1}{2} < y + n_{k-1}\alpha < 1 & \end{array} \right\}. \tag{84}$$

容易に分かるように，A は空集合でない開集合である．$(x, y, \alpha) \in A$ ならば

$$\begin{aligned} r_1(x) &= r_1(x + n_1\alpha) = \cdots = r_1(x + n_{k-2}\alpha) = -1, \quad r_1(x + n_{k-1}\alpha) = 1, \\ r_1(y) &= r_1(y + n_1\alpha) = \cdots = r_1(y + n_{k-1}\alpha) = -1, \end{aligned}$$

だから，k が偶数であることと関数 f の定義 (72) より

$$f(x, \alpha) = -1, \quad f(y, \alpha) = 1, \qquad (x, y, \alpha) \in A, \tag{85}$$

が分かる．(81) の略記法を思い出しておこう．

$$\beta^{-m} A := \{ (x, y, \alpha) \in \mathbb{T}^3 \,|\, \beta^m(x, y, \alpha) \in A \}.$$

$\beta^{-1}A$ の各連結成分は A に相似であるが，とくに

$$B_0^{(-1)} := \left\{ (x, y, \alpha) \;\middle|\; \begin{array}{lll} \frac{3}{4} < x < 1, & \frac{3}{4} < x + n_{k-2}\alpha < 1, & 1 < x + n_{k-1}\alpha < \frac{5}{4}, \\ \frac{3}{4} < y < 1, & \frac{3}{4} < y + n_{k-1}\alpha < 1 & \end{array} \right\}.$$

は A の部分集合である．従って (85) より $B_0^{(-1)}$ 上で $f(x, \alpha) = -1$, $f(y, \alpha) = 1$ である．

さて，与えられた関係式 (82)(83) より

$$\phi_i(x, y, \alpha)\phi_i(2x, 2y, 2\alpha) = -1, \qquad (x, y, \alpha) \in B_0^{(-1)}, \tag{86}$$

である．もし，A 上で $\phi_i(x, y, \alpha) \equiv 1$，a.e. ならば，$B_0^{(-1)}$ 上で $\phi_i(2x, 2y, 2\alpha) \equiv 1$，a.e. であるから，(86) と矛盾する．だから，A 上で $\phi_i \not\equiv 1$．同様に A 上で $\phi_i(x) \not\equiv -1$．従ってとくに

$$\frac{1}{|A|} \int_A \phi_i(x, y, \alpha) dx dy d\alpha =: a_i \in (-1, 1) \tag{87}$$

である．ここに $|A|$ は A のルベーグ測度．

次に各 m につき，$\beta^{-m}A$ の任意の連結成分 $B^{(-m)}$ 上で f は一定符号であることに注意しよう．これは補題 10 と同様に示せる．実際，$f(x, \alpha)$ と $f(y, \alpha)$ の不連続点全体 C は 2 進変換 β に関して不変なので，もし，$f(x, \alpha)$ あるいは $f(y, \alpha)$ が $B^{(-m)}$ 上で符合を変えるならば，$D := B^{(-m)} \cap C \neq \emptyset$ である．これに β^m を作用させれば，$\beta^m D \subset A$ かつ $\beta^m D \subset \beta^m C = C$ となり，これは $A \cap C = \emptyset$，すなわち，$f(x, \alpha)$ あるいは $f(y, \alpha)$ が A で符合を変えることを意味するから，矛盾である．

従って，各 m につき，$\beta^{-m}A$ の任意の連結成分 $B^{(-m)}$ 上で，(82)(83) より

$$\phi_i(x,y,\alpha)\phi_i(2x,2y,2\alpha) \equiv 1 \quad \text{または} \quad \phi_i(x,y,\alpha)\phi_i(2x,2y,2\alpha) \equiv -1. \tag{88}$$

そこで

$$\frac{1}{\left|B^{(-m)}\right|}\int_{B^{(-m)}} \phi_i(x,y,\alpha)dxdyd\alpha = \pm a_i \tag{89}$$

であることを帰納法で示そう．

まず，$m=1$ のときは (88) と変数変換 $x'=2x$, $y'=2y$, $\alpha'=2\alpha$ によって，

$$\int_{B^{(-1)}} \phi_i(x,y,\alpha)dxdyd\alpha = \pm\int_{B^{(-1)}} \phi_i(2x,2y,2\alpha)dx = \pm\frac{1}{8}\int_A \phi_i(x',y',\alpha')dx'dy'd\alpha' \tag{90}$$

である．$|B^{(-1)}| = \frac{1}{8}|A|$ だから

$$\frac{1}{\left|B^{(-1)}\right|}\int_{B^{(-1)}} \phi_i(x,y,\alpha)dxdyd\alpha = \pm a_i. \tag{91}$$

$m-1$ まで (89) が正しい仮定して m のときは (88) より (90) のようにして

$$\int_{B^{(-m)}} \phi_i(x,y,\alpha)dxdyd\alpha = \pm\int_{B^{(-m)}} \phi_i(2x,2y,2\alpha)dxdyd\alpha = \pm\frac{1}{8}\int_{\beta B^{(-m)}} \phi_i(x,y,\alpha)dxdyd\alpha.$$

ここで，$\beta B^{(-m)}$ は $\beta^{-m+1}A$ の連結成分の一つ ($B^{(-m+1)}$ としよう) である．やはり $|B^{(-m)}| = \frac{1}{8}|B^{(-m+1)}|$ だから

$$\frac{1}{\left|B^{(-m)}\right|}\int_{B^{(-m)}} \phi_i(x,y,\alpha)dxdyd\alpha = \pm\frac{1}{\left|B^{(-m+1)}\right|}\int_{B^{(-m+1)}} \phi_i(x,y,\alpha)dxdyd\alpha.$$

従って，帰納法の仮定により m の場合も (89) が成り立つ．

さて，集合 $\cup_{m=1}^{\infty}\beta^{-m}A$ は $\mathbb{T}^3$ で稠密であり，一辺が $0<\rho<1$ の任意の立方体は $m = \lfloor -\log_2\rho\rfloor + 1$ として，$\beta^{-m}A$ の少なくとも一つの連結成分を含む．従って，ある $\delta>0$ が存在し，任意の立方体 S に対して

$$-1+\delta < \frac{1}{|S|}\int_S \phi_i(x,y,\alpha)dxdyd\alpha < 1-\delta. \tag{92}$$

一方，ϕ_i は $\{-1,1\}$-値可測関数だからルベーグの密度定理によれば，$S(x,y,\alpha;\rho)$ を一辺 $\rho>0$ の (x,y,α) を中心とする立方体とするとき

$$\lim_{\rho\to 0}\frac{1}{|S(x,y,\alpha;\rho)|}\int_{S(x,y,\alpha;\rho)} \phi_i(x',y',\alpha')dx'dy'd\alpha' = -1 \text{ または } 1, \qquad \text{a.e.}(x,y,\alpha)\in\mathbb{T}^3.$$

これは (92) と矛盾する．すなわち，(82)(83) を満たす可測関数 ϕ_i は 0 以外に存在しないことが分かる． □

それでは，補題 9 の証明のために T_f のエルゴード性を示そう．可測関数 $\phi:\Omega=\mathbb{T}^3\times\{-1,1\}^2\to\mathbb{C}$ が T_f-不変，すなわち

$$\phi(x,y,\alpha,\epsilon_1,\epsilon_2) = \phi(2x,2y,2\alpha,\epsilon_1 f(x,\alpha),\epsilon_2 f(y,\alpha)), \quad \mu\text{-a.e.},$$

ならば，$\phi \equiv$ 定数 μ-a.e. であることを示す．

$\psi_1(x,y,\alpha) := \sum_{\epsilon_1,\epsilon_2} \phi(x,y,\alpha,\epsilon_1,\epsilon_2)$ とすれば

$$\begin{aligned}\psi_1(x,y,\alpha) &= \sum_{\epsilon_1,\epsilon_2} \phi(2x,2y,2\alpha,\epsilon_1 f(x,\alpha),\epsilon_2 f(y,\alpha)) \\ &= \sum_{\epsilon_1,\epsilon_2} \phi(2x,2y,2\alpha,\epsilon_1,\epsilon_2) \\ &= \psi_1(2x,2y,2\alpha)\end{aligned}$$

であるから，2 進変換 β のエルゴード性より，$\psi_1 \equiv c =$ 定数，a.e. となる．ϕ の代わりに $\phi - c/4$ とすれば，$\psi_1 \equiv 0$, a.e. とできるので，そのように仮定する．

$\psi_2(x,y,\alpha,\epsilon_1) := \sum_{\epsilon_2} \phi(x,y,\alpha,\epsilon_1,\epsilon_2)$ とおけば

$$\begin{aligned}\psi_2(x,y,\alpha,\epsilon_1) &= \sum_{\epsilon_2} \phi(2x,2y,2\alpha,\epsilon_1 f(x,\alpha),\epsilon_2 f(y,\alpha)) \\ &= \sum_{\epsilon_2} \phi(2x,2y,2\alpha,\epsilon_1 f(x,\alpha),\epsilon_2) \\ &= \psi_2(2x,2y,2\alpha,\epsilon_1 f(x,\alpha))\end{aligned}$$

であるから

$$\begin{aligned}\psi_2(x,y,\alpha,-1) &= \psi_2(2x,2y,2\alpha,-f(x,\alpha)) \\ &= \psi_2(2x,2y,2\alpha,-1)\delta_1(f(x,\alpha)) + \psi_2(2x,2y,2\alpha,1)\delta_{-1}(f(x,\alpha))\end{aligned}$$

$\phi_2(2x,2y,2\alpha,-1) + \phi_2(2x,2y,2\alpha,1) = \psi_1(x,y,\alpha) \equiv 0$ だったから

$$\begin{aligned}\psi_2(x,y,\alpha,-1) &= \psi_2(2x,2y,2\alpha,-1)\,(\delta_1(f(x,\alpha)) - \delta_{-1}(f(x,\alpha))) \\ &= \psi_2(2x,2y,2\alpha,-1) f(x,\alpha).\end{aligned}$$

補題 11 より，$\psi_2(x,y,\alpha,-1) \equiv 0$, a.e. が分かる．これより，$\psi_2(x,y,\alpha,1) \equiv 0$, a.e. だから，結局，$\psi_2(x,y,\alpha,\epsilon_1) \equiv 0$, a.e.$(x,y,\alpha,\epsilon_1)$ である．従って

$$\phi(x,y,\alpha,\epsilon_1,-1) + \phi(x,y,\alpha,\epsilon_1,1) = \psi_2 \equiv 0$$

であるから

$$\phi(x,y,\alpha,\epsilon_1,\epsilon_2) = \phi(x,y,\alpha,\epsilon_1,1)\epsilon_2$$

と書ける．いま，ϵ_1 と ϵ_2 の役割を入れ替えれば同じ議論により

$$\phi(x,y,\alpha,\epsilon_1,\epsilon_2) = \phi(x,y,\alpha,1,\epsilon_2)\epsilon_1$$

この二つの式より，直ちに

$$\phi(x,y,\alpha,\epsilon_1,\epsilon_2) = \phi(x,y,\alpha,1,1)\epsilon_1\epsilon_2.$$

従って，ϕ が T_f 不変であることは

$$\phi(x,y,\alpha,1,1) = \phi(2x,2y,2\alpha,1,1) f(x,\alpha) f(y,\alpha)$$

を意味するが，補題 11 により，これを満たす ϕ は 0 しかない．これで，T_f のエルゴード性，従って $\{\zeta_m\}_m$ の既約性の証明が完了した． □

非周期性の証明

補題 9 の証明は残るところ非周期性の証明であるが，すでに既約性は示したので，$\mu(\zeta_0 = \zeta_1) > 0$ であればよい．それには

$$F' := \left\{ (x,y,\alpha) \;\middle|\; \begin{array}{ll} 0 < x < \frac{1}{2}, & 0 < x + n_{k-1}\alpha < \frac{1}{2}, \\ 0 < y < \frac{1}{2}, & 0 < y + n_{k-1}\alpha < \frac{1}{2} \end{array} \right\},$$

$$F'' := \left\{ (x,y,\alpha) \;\middle|\; \begin{array}{ll} 0 < x < \frac{1}{4}, & 0 < x + n_{k-1}\alpha < \frac{1}{4}, \\ 0 < y < \frac{1}{4}, & 0 < y + n_{k-1}\alpha < \frac{1}{4} \end{array} \right\},$$

として，$F := F' \times \{1\} \times \{1\} \subset \Omega$ とすれば，これは Ω の分割 $\{F_j\}_{j=1}^{4J}$ に属する一つの集合 F_j である．$H := F'' \times \{1\} \times \{1\}$ とおけば $H \subset F$ であり，$(x,y,\alpha) \in F''$ ならば $f(x,\alpha) = f(y,\alpha) = 1$ なので，$(x,y,\alpha,\epsilon_1,\epsilon_2) \in H$ ならば

$$\zeta_0(x,y,\alpha,\epsilon_1,\epsilon_2) = \zeta_1(x,y,\alpha,\epsilon_1,\epsilon_2) = j$$

である．$\mu(H) > 0$ だから，$\{\zeta_m\}_m$ の非周期性がいえた．

これで，補題 9 の証明，従って定理 10′(定理 10) の証明が完了した． □

3.3.5 二項間相関の指数的収束の精密評価

定理 10′(定理 10) における指数的収束の速度を表す ρ を評価しよう．現時点では二項間相関の場合にだけ次のような定理を得ている．[42]

定理 14 [1] 任意の $\rho > \rho_0 := \sqrt{\left(1 + \sqrt{17}\right)/8} = 0.80024\ldots$ について

$$\mathbf{E}\left[X_0^{(m)}(\bullet;\alpha) X_k^{(m)}(\bullet;\alpha)\right] = o(\rho^m), \quad \text{as } m \to \infty, \quad k \in \mathbb{N}, \quad \text{a.e. } \alpha.$$

実際には [1] ではもっと詳しく次の等式を導いている．

定理 15

$$\int_{\mathbb{T}^1} d\alpha \left(\mathbf{E}\left[X_0^{(m)}(\bullet;\alpha) X_k^{(m)}(\bullet;\alpha)\right]\right)^2$$
$$= \left(\frac{1}{2} + \frac{5\sqrt{17}}{102}\right)\left(\frac{1+\sqrt{17}}{8}\right)^m + \left(\frac{1}{2} - \frac{5\sqrt{17}}{102}\right)\left(\frac{1-\sqrt{17}}{8}\right)^m, \quad m \in \mathbb{N}, \quad k \in \mathbb{N}.$$

定理 15 から定理 14 が従うことは前 § の (80) から (79) が従うのと同様にして分かる．もちろん，定理 14 における定数 ρ_0 は最善であって，これより小さくすることはできない．[1] では，今の場合に補題 9 のマルコフ推移確率行列[43] $p(i,j)$ のすべての固有値を求め (その

[42] 高信が特別な 4 項間相関の場合に指数 ρ を求めている．

[43] 今の場合，それは 96×96 の正方行列である．

うちの絶対値最大の固有値が ρ_0^2 である), さらに今の問題の解を固有値 ρ_0^2 と $\left(1-\sqrt{17}\right)/8$ に対応する 2 次元固有空間上で表現できることを見出し，定理 15 に至っている．また，定理 15 は [29] でもフーリエ展開の方法で導出している．

定理 15 の証明. ここでは，漸化式の方法で証明しよう．まず，変換 $\mathbb{T}^1 \ni x \mapsto kx \in \mathbb{T}^1$ によってルベーグ測度は不変だから，$k = 1$ の場合に示せば十分である．そこで各 $m \in \mathbb{N}$ に対して

$$a_m := \int_{\mathbb{T}^1} d\alpha \left(\mathbf{E}\left[X_0^{(m)}(\bullet;\alpha)X_1^{(m)}(\bullet;\alpha)\right]\right)^2 = \int_{\mathbb{T}^1} d\alpha \left(\int_{\mathbb{T}^1} \prod_{i=1}^{m} r_i(x)r_i(x+\alpha)\,dx\right)^2$$

とおく．このとき，次の二つの式を示すことによって定理 15 を証明する．

$$a_1 = a_2 = \frac{1}{3} \tag{93}$$

$$a_{m+2} = \frac{1}{4}a_{m+1} + \frac{1}{4}a_m, \quad m \in \mathbb{N} \tag{94}$$

(93) の証明. 直接計算によって

$$a_1 = \int_{\mathbb{T}^1} d\alpha \left(\int_{\mathbb{T}^1} r_1(x)r_1(x+\alpha)dx\right)^2 = \int_{\mathbb{T}^1} (\,|2-4\alpha|-1\,)^2\,d\alpha = \frac{1}{3}.$$

次に $r_1(x)r_2(x) = r_1\left(x+\frac{1}{4}\right)$ に注意すれば

$$\begin{aligned} a_2 &= \int_{\mathbb{T}^1} d\alpha \left(\int_{\mathbb{T}^1} r_1(x)r_2(x)r_1(x+\alpha)r_2(x+\alpha)dx\right)^2 \\ &= \int_{\mathbb{T}^1} d\alpha \left(\int_{\mathbb{T}^1} r_1\left(x+\frac{1}{4}\right)r_1\left(x+\alpha+\frac{1}{4}\right)dx\right)^2 \\ &= \int_{\mathbb{T}^1} d\alpha \left(\int_{\mathbb{T}^1} r_1(x)r_1(x+\alpha)dx\right)^2 = a_1. \end{aligned}$$

(94) の証明. (44) に倣って

$$A^{(m)}(\alpha) := \int_{\mathbb{T}^1} \prod_{i=1}^{m} r_i(x)r_i(x+\alpha)\,dx$$

とおく．以下，α に関する平均 (ルベーグ積分) を $\mathbf{E}$ で表すことにする．とくに $a_m = \mathbf{E}\left[A^{(m)}(\alpha)^2\right]$ である．また

$$\begin{cases} \xi_m := \mathbf{E}\left[A(\alpha^{(m)U})^2 + A(\alpha^{(m)L})^2\right] \\ \eta_m := \mathbf{E}\left[A(\alpha^{(m)U})A(\alpha^{(m)L})\right] \end{cases}$$

とおく．それでは，漸化式 (94) を発見的に導く方法を与えよう．

補題 12

$$a_m \;=\; \frac{1}{3}\xi_m + \frac{1}{3}\eta_m \tag{95}$$

$$\begin{pmatrix}\xi_{m+1}\\ \eta_{m+1}\end{pmatrix} \;=\; \begin{pmatrix}\frac{3}{4} & \frac{1}{2}\\ -\frac{1}{4} & -\frac{1}{2}\end{pmatrix}\begin{pmatrix}\xi_m\\ \eta_m\end{pmatrix} \tag{96}$$

証明. 定理 11′ によれば

$$A^{(m)}(\alpha) \;=\; (1-2^m\langle\alpha\rangle_m)\,A(\alpha^{(m)U}) + 2^m\langle\alpha\rangle_m A(\alpha^{(m)L}).$$

ここで $A(\bullet)$ は定義 14 で与えられた関数，$\langle\alpha\rangle_m := \alpha - \lfloor\alpha\rfloor_m$ とする．上の式から

$$\begin{aligned} a_m \;&=\; \mathbf{E}\left[(1-2^m\langle\alpha\rangle_m)^2\,A(\alpha^{(m)U})^2\right] + \mathbf{E}\left[(2^m\langle\alpha\rangle_m)^2 A(\alpha^{(m)L})^2\right]\\ &\quad +2\mathbf{E}\left[(1-2^m\langle\alpha\rangle_m)\,(2^m\langle\alpha\rangle_m)A(\alpha^{(m)U})A(\alpha^{(m)L})\right]\\ &=\; \mathbf{E}\left[(1-2^m\langle\alpha\rangle_m)^2\right]\mathbf{E}\left[A(\alpha^{(m)U})^2\right] + \mathbf{E}\left[(2^m\langle\alpha\rangle_m)^2\right]\mathbf{E}\left[A(\alpha^{(m)L})^2\right]\\ &\quad +2\mathbf{E}\left[(1-2^m\langle\alpha\rangle_m)\,(2^m\langle\alpha\rangle_m)\right]\mathbf{E}\left[A(\alpha^{(m)U})A(\alpha^{(m)L})\right] \end{aligned}$$

ここでは $\langle\alpha\rangle_m$ と $\alpha^{(m)L}$ または $\alpha^{(m)U}$ の独立性を用いた．$2^m\langle\alpha\rangle_m = \langle 2^m\alpha\rangle$ であるから，その分布は α 自身の分布，すなわち一様分布，に等しいから

$$\begin{aligned} a_m \;&=\; \mathbf{E}\left[(1-\alpha)^2\right]\mathbf{E}\left[A(\alpha^{(m)U})^2\right] + \mathbf{E}\left[\alpha^2\right]\mathbf{E}\left[A(\alpha^{(m)L})^2\right]\\ &\quad +2\mathbf{E}\left[(1-\alpha)\alpha\right]\mathbf{E}\left[A(\alpha^{(m)U})A(\alpha^{(m)L})\right]\\ &=\; \frac{1}{3}\mathbf{E}\left[A(\alpha^{(m)U})^2\right] + \frac{1}{3}\mathbf{E}\left[A(\alpha^{(m)L})^2\right] + \frac{1}{3}\mathbf{E}\left[A(\alpha^{(m)U})A(\alpha^{(m)L})\right] \end{aligned}$$

となって (95) が示された．次に補題 3 によって

$$\begin{aligned} &\mathbf{E}\left[A(\alpha^{(m+1)U})^2 + A(\alpha^{(m+1)L})^2\right]\\ &=\; \mathbf{E}\left[\frac{1}{4}\left(A(\mathcal{U}\alpha^{(m+1)U}) + A(\mathcal{L}\alpha^{(m+1)U})\right)^2 + \frac{1}{4}\left(A(\mathcal{U}\alpha^{(m+1)L}) + A(\mathcal{L}\alpha^{(m+1)L})\right)^2\right]\\ &=\; \mathbf{E}\left[\frac{1}{4}\left(A(\alpha^{(m)U}) + A(\alpha^{(m)L})\right)^2 + A(\alpha^{(m)L})^2\,;\, d_{m+1}(\alpha) = 0\right]\\ &\quad + \mathbf{E}\left[A(\alpha^{(m)U})^2 + \frac{1}{4}\left(A(\alpha^{(m)U}) + A(\alpha^{(m)L})\right)^2\,;\, d_{m+1}(\alpha) = 1\right]\\ &=\; \frac{1}{2}\mathbf{E}\left[\frac{1}{4}\left(A(\alpha^{(m)U}) + A(\alpha^{(m)L})\right)^2 + A(\alpha^{(m)L})^2\right]\\ &\quad + \frac{1}{2}\mathbf{E}\left[A(\alpha^{(m)U})^2 + \frac{1}{4}\left(A(\alpha^{(m)U}) + A(\alpha^{(m)L})\right)^2\right]\\ &=\; \frac{3}{4}\mathbf{E}\left[A(\alpha^{(m)U})^2 + A(\alpha^{(m)L})^2\right] + \frac{1}{2}\mathbf{E}\left[A(\alpha^{(m)U})A(\alpha^{(m)L})\right] \end{aligned}$$

また同様に

$$\begin{aligned}
&\mathbf{E}\left[A(\alpha^{(m+1)U})A(\alpha^{(m+1)L})\right] \\
&= \mathbf{E}\left[\left(-\frac{1}{2}A(\alpha^{(m)U}) - \frac{1}{2}A(\alpha^{(m)L})\right)A(\alpha^{(m)L})\,;\, d_{m+1}(\alpha) = 0\right] \\
&\quad + \mathbf{E}\left[A(\alpha^{(m)U})\left(-\frac{1}{2}A(\alpha^{(m)U}) - \frac{1}{2}A(\alpha^{(m)L})\right)\,;\, d_{m+1}(\alpha) = 1\right] \\
&= -\frac{1}{4}\mathbf{E}\left[\left(A(\alpha^{(m)U}) + A(\alpha^{(m)L})\right)A(\alpha^{(m)L})\right] \\
&\quad - \frac{1}{4}\mathbf{E}\left[A(\alpha^{(m)U})\left(A(\alpha^{(m)U}) + A(\alpha^{(m)L})\right)\right] \\
&= -\frac{1}{4}\mathbf{E}\left[A(\alpha^{(m)U})^2 + A(\alpha^{(m)L})^2\right] - \frac{1}{2}\mathbf{E}\left[A(\alpha^{(m)U})A(\alpha^{(m)L})\right]
\end{aligned}$$

であるから (96) が示された. □

それでは (94) の証明を再開しよう．まず，(96) より

$$\begin{pmatrix}\xi_{m+2}\\ \eta_{m+2}\end{pmatrix} = \begin{pmatrix} \frac{3}{4} & \frac{1}{2} \\ -\frac{1}{4} & -\frac{1}{2}\end{pmatrix}^2\begin{pmatrix}\xi_m\\ \eta_m\end{pmatrix} = \begin{pmatrix} \frac{7}{16} & \frac{1}{8} \\ -\frac{1}{16} & \frac{1}{8}\end{pmatrix}\begin{pmatrix}\xi_m\\ \eta_m\end{pmatrix}$$

であるから

$$\begin{aligned}
a_{m+2} &= \frac{1}{3}\xi_{m+2} + \frac{1}{3}\eta_{m+2} \\
&= \frac{1}{3}\left(\frac{7}{16}\xi_m + \frac{1}{8}\eta_m\right) + \frac{1}{3}\left(-\frac{1}{16}\xi_m + \frac{1}{8}\eta_m\right) \\
&= \frac{1}{8}\xi_m + \frac{1}{12}\eta_m. \qquad (97)
\end{aligned}$$

また同様に

$$a_{m+1} = \frac{1}{3}\xi_{m+1} + \frac{1}{3}\eta_{m+1} = \frac{1}{6}\xi_m \qquad (98)$$

そこで

$$a_{m+2} = c_1 a_{m+1} + c_2 a_m, \quad m \in \mathbb{N},$$

となる定数 c_1, c_2 を求めるには (95)(97)(98) より

$$\begin{aligned}
\frac{1}{8}\xi_m + \frac{1}{12}\eta_m &= c_1\frac{1}{6}\xi_m + c_2\left(\frac{1}{3}\xi_m + \frac{1}{3}\eta_m\right) \\
&= \left(\frac{1}{6}c_1 + \frac{1}{3}c_2\right)\xi_m + \frac{1}{3}c_2\eta_m, \quad m \in \mathbb{N},
\end{aligned}$$

を解けばよい．両辺の係数比較をして

$$c_1 = c_2 = \frac{1}{4}$$

であることが分かる．これで (94) の証明，従って定理 15 の証明が完了した. □

4　モンテカルロ積分

数値解析学では，一般に，被積分関数のクラスごとに適切な数値積分法を選んで積分の近似計算を行う．被積分関数が滑らかであればあるほど，サンプル点の少ない能率のよい数値積分法が存在する．複雑な関数になればなるほど，能率のよい数値積分法がなくなって，最後の砦がモンテカルロ積分である．モンテカルロ積分は精度はよくないけれども，非常に複雑な関数でも積分できるという利点を持つ．

§ 4.1 と § 4.2 で扱う被積分関数のクラスは $\mathbb{T}^1$ で定義された $\mathcal{B}_m$-可測関数である．2 進数展開写像を通じて，これは m 回の硬貨投げの関数の平均を数値的に求めることを考えることと同等である．§ 4.1 では興味のある $2^m \gg 1$ の場合には，L^2-ロバスト性と呼ばれる基準の下で，i.i.d.-サンプリングあるいはペアごとに独立な確率変数によるサンプリングがほぼ最適であることを示す．そして § 4.2 では，$\mathcal{B}_m$-可測関数のための能率的なペアごとに独立なサンプルによるサンプリング法を紹介する．例 9 では § 1.2.2 の例題に具体的な数値解の例を示す．

§ 4.3 ではコンピュータで扱うことのできる確率変数の最も広いクラスを設定する．そして次の § 4.4 でそのクラスに属する確率変数のためのペアごとに独立なサンプルによるサンプリング法を紹介する．

4.1　L^2-ロバスト性

§ 1.5 では m 回の硬貨投げの関数として得られる確率変数の平均を大数の法則を利用して推定する方法 — i.i.d.-サンプリングとペアごとに独立な同分布確率変数列を用いたサンプリング — について述べた．ここでは，一般の確率変数列によるサンプリングの中で，これらのサンプリング法が持つある特徴について述べる．

§ 1.5 で扱われた確率変数は $\{0,1\}^m$ 上の関数であったが，これは 2 進小数展開によって自然に D_m 上の関数，さらには $\mathbb{T}^1$ 上の $\mathcal{B}_m$-可測関数と思うことができる．以下では，被積分関数は主として $\mathbb{T}^1$ 上の関数とする．

定理 16 (サンプリングに関する基本的な不等式，[20]) $\{\psi_l\}_{l=1}^{2^m-1}$ を $L^2(\mathcal{B}_m)$ の正規直交系で，各 l に対して $\int_{\mathbb{T}^1} \psi_l(x)dx = 0$ を満たすとする．このとき，任意の確率変数列 $\{X_n\}_{n=1}^{2^m} \subset \mathbb{T}^1$ に対して，不等式

$$\sum_{l=1}^{2^m-1} \mathbf{E}\left[\left|\frac{1}{N}\sum_{n=1}^{N}\psi_l(X_n)\right|^2\right] \geq \frac{2^m}{N} - 1, \qquad 1 \leq N \leq 2^m, \tag{99}$$

が成り立つ．

証明. まず，$\psi_l \in L^2(\mathcal{B}_m)$ なので，$\{X_n\}_{n=1}^N \subset D_m$ としてよい．さらに，定理の特別な場合として，各 X_n が決定論的 (detreministic) であってもよい．また逆に，任意の決定論的な数列 $\{x_n\}_{n=1}^N$ について定理が成り立てば，任意の確率過程 $\{X_n\}_{n=1}^N$ に対して成り立つ．そこで，$\{x_n\}_{n=1}^N \subset D_m$ について定理の主張を示すことにする．

まず,

$$g(y) := \frac{2^m}{N}\sum_{n=1}^{N}\mathbf{1}_{[x_n,x_n+2^{-m})}(y) \tag{100}$$

とすると任意の $f \in L^2(\mathcal{B}_m)$ に対して,

$$\frac{1}{N}\sum_{n=1}^{N} f(x_n) = \langle f, g\rangle_{L^2(\mathcal{B}_m)} := \int_{\mathbb{T}^1} f(x)g(x)dx.$$

が成り立つ．関数系 $\{1, \psi_1, \dots \psi_{2^m-1}\}$ は $L^2(\mathcal{B}_m)$ の正規直交基底なので，パーセヴァル (Parseval) の等式 (ピュタゴラス (Pythagoras) の定理) から

$$\|g\|^2_{L^2(\mathcal{B}_m)} = \langle g, 1\rangle^2_{L^2(\mathcal{B}_m)} + \sum_{l=1}^{2^m-1}\langle g, \psi_l\rangle^2_{L^2(\mathcal{B}_m)}. \tag{101}$$

$\langle g, 1\rangle_{L^2(\mathcal{B}_m)} = 1$ および不等式

$$\begin{aligned}
\|g\|^2_{L^2(\mathcal{B}_m)} &= \frac{2^{2m}}{N^2}\sum_{n=1}^{N}\sum_{n'=1}^{N}\int_{\mathbb{T}^1}\mathbf{1}_{[x_n,x_n+2^{-m})}\mathbf{1}_{[x_{n'},x_{n'}+2^{-m})}dx \\
&\geq \frac{2^{2m}}{N^2}\sum_{n=n'=1}^{N}\int_{\mathbb{T}^1}\mathbf{1}_{[x_n,x_n+2^{-m})}\mathbf{1}_{[x_{n'},x_{n'}+2^{-m})}dx \\
&= \frac{2^{2m}}{N^2}\sum_{n=n'=1}^{N}\frac{1}{2^m} = \frac{2^m}{N}
\end{aligned} \tag{102}$$

を (101) に代入すれば,

$$\frac{2^m}{N} \leq 1 + \sum_{l=1}^{2^m-1}\left|\frac{1}{N}\sum_{n=1}^{N}\psi_l(x_n)\right|^2.$$

これより (99) が従う． □

系 2 ([20]) $2^m \geq N > 1$ とする．任意の確率変数列 $\{X_n\}_{n=1}^{\infty} \subset \mathbb{T}^1$ に対して，定数関数でないある $f \in L^2(\mathcal{B}_m)$ が存在して次の不等式を満たす.

$$\mathbf{E}\left[\left|\frac{1}{N}\sum_{n=1}^{N} f(X_n) - \int_{\mathbb{T}^1} f(x)dx\right|^2\right] \geq \left(\frac{1}{N} - 2^{-m}\right)\mathbf{Var}(f). \tag{103}$$

定理 16 より，少なくとも一つの ψ_l は不等式 (103) を満たすから系 2 が従う.

系 2 によれば，$2^m \gg N \gg 1$ のとき，サンプリングによる数値積分の収束の速さは，どのような点列 (確率変数列) を取ろうとも，最善でも i.i.d.-サンプリング程度であるような被積分関数が存在することが分かる.

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 16 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 (99) 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．

これに対して，low risk なサンプリング法，すなわち，どのような被積分関数に対しても，常に安定した積分の近似値を与える数値積分法をロバスト (robust，堅固) である，ということにしよう．次に，ロバスト性の一つの定量的な定義を与える．

定義 18　確率変数列 $\{X_n\}_{n=1}^{2^m}$ を用いたサンプリングによる数値積分法が **L^2-ロバスト** (詳しくは $L^2(\mathcal{B}_m)$-ロバスト) であるとは，任意の $L^2(\mathcal{B}_m)$-関数 f に対して，

$$\mathbf{E}\left[\left|\frac{1}{N}\sum_{n=1}^{N} f(X_n) - \int_{\mathbb{T}^1} f(x)dx\right|^2\right] \le \frac{\mathbf{Var}(f)}{N}, \qquad 1 \le N \le 2^m, \tag{104}$$

を満たすことをいう．

決定論的なサンプリングはこの意味でロバストではないことに注意せよ (決定論的数列 $\{x_n\}_{n=1}^{N}$ を用いて (100) で定義された関数 $g(y)$ を数値積分する場合を考えよ．)

i.i.d.-サンプリングの場合は (104) が等式で成り立つから，もちろん，L^2-ロバストである．定理 16 と系 2 によれば，$2^m \gg N \gg 1$ のとき，不等式 (104) はこれ以上ほとんど改良されないことが分かる．一般に，L^2-ロバストなサンプリング法は i.i.d.-サンプリングと同程度に low risk だが，やはりそれと同程度の return しか得られない．このように L^2-ロバスト性という観点からは i.i.d.-サンプリングはほとんど最適なサンプリング法なのである．

さらに，ペアごとに独立な確率変数列を用いたサンプリングの場合も (104) が等式として成り立つから，L^2-ロバストである．その上，§ 1.5.2 で述べたように，このサンプリング法はモンテカルロ積分のための安全な疑似乱数生成器としても機能する．

4.2　ランダム-ワイル-サンプリング

4.2.1　定義とペアごとの独立性の定理

§ 1.5.2 定理 1 の証明で構成した疑似乱数生成器 $g : \{0,1\}^{2m} \to \{0,1\}^{Nm}$，$N \le 2^m$，は N 個の $\{0,1\}^m$ 上一様分布するペアごとに独立な確率変数列を生成するが，そのような疑似乱数生成器の中で種のビット数が $2m$ と最も小さい．しかし，GF(2^m) 上の演算が複雑なため実用的ではない．この § では，種のビット数は最小ではないが，ずっと容易にかつ高速にペアごとに独立なサンプルを生成する方法 — ランダム-ワイル-サンプリング —を紹介する．

定義 19 (ランダム-ワイル-サンプリング，cf. [20, 26])　$j \in \mathbb{N}$ とし

$$Z_n(x,\alpha) := \lfloor x + n\alpha \,(\mathrm{mod}\, 1) \rfloor_m \in D_m, \quad (x,\alpha) \in D_{m+j} \times D_{m+j}, \quad n = 1, 2, 3, \ldots, 2^{j+1},$$

と定義する．$\{Z_n\}_{n=1}^{2^{j+1}}$ をサンプリング点として用いる数値積分法を**ランダム-ワイル-サンプリング** (random Weyl sampling，以下 RWS と略す) と呼ぶ．

定理 17 (cf. [20, 26][44]) $D_{m+j}\times D_{m+j}$ 上の一様確率測度 $P_{(m+j,m+j)}$ の下で $\{Z_n\}_{n=1}^{2^{j+1}}$ はペアごとに独立であり，各 Z_n は D_m で一様分布する．

証明. 周期 1 の周期関数 $F,G:\mathbb{R}\to\mathbb{R}$ がそれぞれ $F(x)=F(\lfloor x\rfloor_m)$，$G(x)=G(\lfloor x\rfloor_m)$，$x\in[0,1)$，を満たすとき，$1\le n<n'\le 2^{j+1}$ に対して

$$\mathbf{E}[F(Z_{n'})G(Z_n)]=\int_{\mathbb{T}^1}F(t)dt\int_{\mathbb{T}^1}G(s)ds \tag{105}$$

となることを示せばよい．ここに $\mathbf{E}$ は $P_{(m+j,m+j)}$ による平均値を表す．

定義により

$$\begin{aligned}\mathbf{E}[F(Z_{n'})G(Z_n)] &= \frac{1}{2^{2m+2j}}\sum_{q=1}^{2^{m+j}}\sum_{p=1}^{2^{m+j}}F\left(\frac{p}{2^{m+j}}+\frac{n'q}{2^{m+j}}\right)G\left(\frac{p}{2^{m+j}}+\frac{nq}{2^{m+j}}\right)\\ &= \frac{1}{2^{2m+2j}}\sum_{q=1}^{2^{m+j}}\sum_{p=1}^{2^{m+j}}F\left(\frac{p}{2^{m+j}}+\frac{(n'-n)q}{2^{m+j}}\right)G\left(\frac{p}{2^{m+j}}\right).\end{aligned} \tag{106}$$

ここで，$0<n'-n=2^il\le 2^{j+1}-1$，ただし $0\le i\le j$ かつ l は奇数，としよう．すると

$$\frac{1}{2^{m+j}}\sum_{q=1}^{2^{m+j}}F\left(\frac{p}{2^{m+j}}+\frac{(n'-n)q}{2^{m+j}}\right)=\frac{1}{2^{m+j}}\sum_{q=1}^{2^{m+j}}F\left(\frac{p}{2^{m+j}}+\frac{lq}{2^{m+j-i}}\right). \tag{107}$$

各 $r=1,2,3,\dots,2^{m+j-i}$ に対して $lq_r\equiv r(\mathrm{mod}\,2^{m+j-i})$ なる q_r を一つ定めれば l は奇数だから

$$\begin{aligned}&\#\{1\le q\le 2^{m+j}\,|\,lq\equiv r(\mathrm{mod}\,2^{m+j-i})\}\\ &\quad=\#\{1\le q\le 2^{m+j}\,|\,lq\equiv lq_r(\mathrm{mod}\,2^{m+j-i})\}\\ &\quad=\#\{1\le q\le 2^{m+j}\,|\,l(q-q_r)\equiv 0(\mathrm{mod}\,2^{m+j-i})\}\\ &\quad=\#\{1\le q\le 2^{m+j}\,|\,q\equiv q_r(\mathrm{mod}\,2^{m+j-i})\}\\ &\quad=2^i.\end{aligned}$$

このことから

$$\begin{aligned}\frac{1}{2^{m+j}}\sum_{q=1}^{2^{m+j}}F\left(\frac{p}{2^{m+j}}+\frac{lq}{2^{m+j-i}}\right) &= \frac{1}{2^{m+j-i}}\sum_{r=1}^{2^{m+j-i}}F\left(\frac{p}{2^{m+j}}+\frac{r}{2^{m+j-i}}\right)\\ &= \frac{1}{2^{m+j-i}}\sum_{r=1}^{2^{m+j-i}}F\left(\frac{r}{2^{m+j-i}}\right)\\ &= \int_{\mathbb{T}^1}F(t)dt.\end{aligned} \tag{108}$$

[44]定理 17 は [20, 26] で紹介されているランダム-ワイル-サンプリングより，わずかばかり改良されている．

従って (106)(107)(108) から

$$\begin{aligned}\mathbf{E}[F(Z_{n'})G(Z_n)] &= \frac{1}{2^{m+j}}\sum_{p=1}^{2^{m+j}}\left(\frac{1}{2^{m+j}}\sum_{q=1}^{2^{m+j}}F\left(\frac{p}{2^{m+j}}+\frac{(n'-n)q}{2^{m+j}}\right)\right)G\left(\frac{p}{2^{m+j}}\right)\\ &= \frac{1}{2^{m+j}}\sum_{p=1}^{2^{m+j}}\left(\frac{1}{2^{m+j}}\sum_{q=1}^{2^{m+j}}F\left(\frac{p}{2^{m+j}}+\frac{lq}{2^{m+j-i}}\right)\right)G\left(\frac{p}{2^{m+j}}\right)\\ &= \int_{\mathbb{T}^1}F(t)dt\cdot\frac{1}{2^{m+j}}\sum_{p=1}^{2^{m+j}}G\left(\frac{p}{2^{m+j}}\right) = \int_{\mathbb{T}^1}F(t)dt\int_{\mathbb{T}^1}G(s)ds.\end{aligned}$$

以上で (105) が示された. □

2 進展開写像によって $\{0,1\}^m \cong D_m$ と同一視する．定理 17 の $\{Z_n\}_{n=1}^{2^{j+1}}$ を用いて疑似乱数生成器 $\tilde{g}:\{0,1\}^{2m+2j}\to\{0,1\}^{Nm}$，ただし $N\le 2^{j+1}$，を定理 1 の証明のように

$$\begin{aligned}\tilde{g}(\omega') &:= (Z_1(\omega'),Z_2(\omega'),\dots,Z_N(\omega'))\in D_m^N\cong\{0,1\}^{Nm},\\ \omega' &= (x,\alpha)\in D_{m+j}\times D_{m+j}\cong\{0,1\}^{2m+2j}\end{aligned}$$

と定義しよう．この疑似乱数生成器を用いて § 1.2.2 の例題を具体的に解くことができる．

たとえば例 5 (§ 1.4.1) の類似として，$m=2^7$，$j=18$，$N=2^{19}$ として，疑似乱数生成器 $\tilde{g}:\{0,1\}^{292}\to\{0,1\}^{2^{26}}$ を用いるとき,[45] リスク評価は

$$P_{292}\left(\left|\frac{S_{2^{19}}(\tilde{g}(\omega'))}{2^{19}}-p\right|\ge 2^{-8}\right)\le\frac{1}{32} \tag{109}$$

のように得られる (cf.(3)). この疑似乱数生成器 $\tilde{g}$ の方が定理 1 の疑似乱数生成器 g より少し長い種が必要だから，その意味では能率が悪いが，その代わりアルゴリズムが簡単で疑似乱数の生成がずっと早くできる.

例 9　疑似乱数生成器 $\tilde{g}:\{0,1\}^{292}\to\{0,1\}^{2^{26}}$ を用いて具体的に $S_{2^{19}}(\tilde{g}(\omega'))$ を求めた例を挙げよう．筆者が § 1.6.1 にある方法で硬貨を実際に 292 回投げて選んだ $\omega'=(x,\alpha)\in D_{146}\times D_{146}\cong\{0,1\}^{292}$ は次の通りである (2 進数表示).

$$\begin{aligned}x &= \texttt{0.1110110101 1011101101 0100000011 0110101001 0101000100}\\ &\quad\texttt{0101111101 1010000000 1010100011 0100011001 1101111101}\\ &\quad\texttt{1101010011 1111001000 1010010000 1101011101 001100}\\ \alpha &= \texttt{0.1100000111 0111000100 0001101011 1001000001 0010001000}\\ &\quad\texttt{1010101101 1110101110 0010010011 1000000011 0101000110}\\ &\quad\texttt{0101110010 0101111110 0110110000 0101100001 101110}\end{aligned}$$

このときコンピュータよって $S_{2^{19}}(\tilde{g}(\omega'))=204,650$ と計算された (実装については § 5.1 を見よ). 従って

$$\frac{S_{2^{19}}(\tilde{g}(\omega'))}{2^{19}} = \frac{204650}{524288} = 0.390339$$

であるから，これが求める確率 p の近似値であると考えられる.

[45] 今の場合 $2m+2j=292$.

注意 11　RWS の場合，アリスがどんな $\omega' = (x, \alpha) \in \{0,1\}^{2m+2j}$ を選ぶべきでないか，について少しだけ助言をすることができる．それは，とくに α を簡単な数にしないことである．極端な場合 $\alpha = (0, 0, \ldots, 0) \in \{0,1\}^{m+j}$ と選ぶと RWS はほぼ完全に失敗することがすぐ分かる (cf. § 1.6.1).

4.2.2　$m \gg 1$ の場合の RWS

RWS を確率変数 $\{0,1\}^m \to \mathbb{R}$ の数値積分に適用するためには，アリスは種 $\omega' \in \{0,1\}^{2m+2j}$ を選ばなければならない．ここでもし，m が巨大だと，実際には ω' をアリスの意思で選ぶことさえ困難になってくる．そのような場合は，ω' をさらに別の補助的な疑似乱数生成器 $g' : \{0,1\}^n \to \{0,1\}^{2m+2j}$ で選ぶ破目になる (cf. 注意 1).

この補助的な疑似乱数生成器 g' を選ぶ際,

(1) i.i.d.-サンプリングに比べれば，使用する疑似乱数のビット数を大幅に削減できるため，RWS は i.i.d.-サンプリングより，疑似乱数生成器 g' の質に対して非常に鈍感である．つまり，質の低い疑似乱数生成器を使用してもあまり誤差が大きくなることは稀である．

(2) 理想的には g' として計算量的に安全なものを選びたい．一般に高品質の疑似乱数生成器はサンプルの生成速度が遅いが，RWS では使用する疑似乱数が少ないため，そうした疑似乱数生成器も使用可能である．

とくに (2) は，数値積分において疑似乱数の生成速度がまったく重要な要因ではないことを示した点で重要である．

4.2.3　ある極限定理

RWS はペアごとに独立な確率変数による数値積分法なので，大数の法則が成り立つ．ここでは，さらに RWS による標本平均が中心極限定理に基づいたスケーリングの下で 0 に確率収束することを示す.

この目的のために，ここでは D_m ではなく，($m \to \infty$ として) $\mathbb{T}^1$ 上の RWS を考える．まず，ペアごとの独立性について見ておく．次の定理 18 は定理 1 の連続版ということがいえるかも知れない.

定理 18 ([9, 26]) $(\mathbb{T}^2, \mathcal{B}^2, \mathbb{P}^2)$ 上の $\mathbb{T}^1$-値確率変数列 $\{x + n\alpha\}_{n\in\mathbb{Z}}$ はペアごとに独立，すなわち $n \neq n'$ ならば $(x + n\alpha)$ と $(x + n'\alpha)$ は独立，であり，各 $(x + n\alpha)$ は $\mathbb{T}^1$ 上に一様分布する.

証明. 任意の $f, h \in L^2(\mathbb{T}^1, \mathcal{B}, \mathbb{P})$ に対して

$$\begin{aligned}\int_{\mathbb{T}^1} d\alpha \int_{\mathbb{T}^1} dx\, f(x + n\alpha)h(x + n'\alpha) &= \int_{\mathbb{T}^1} d\alpha \int_{\mathbb{T}^1} dx\, f(x)h(x + (n' - n)\alpha) \\ &= \int_{\mathbb{T}^1} dx\, f(x) \int_{\mathbb{T}^1} d\alpha\, h(x + (n' - n)\alpha)\end{aligned}$$

$$
\begin{aligned}
&= \int_{\mathbb{T}^1} dx\, f(x) \int_{\mathbb{T}^1} d\alpha\, h((n'-n)\alpha) \\
&= \int_{\mathbb{T}^1} dx\, f(x) \int_{\mathbb{T}^1} d\alpha\, h(\alpha).
\end{aligned}
$$

□

よく知られているようにワイル変換はルベーグ確率空間 $(\mathbb{T}^1, \mathcal{B}, \mathbb{P})$ 上でエルゴード的であり，従って $f \in L^1(\mathbb{T}^1, \mathcal{B}, \mathbb{P})$ に対して大数の法則が成り立つ．とくに f が滑らかな関数のときは大数の法則の収束が早い．実際，$\exp(2k\pi\sqrt{-1}\,x)$，$0 \neq k \in \mathbb{Z}$，の場合，

$$
\frac{1}{N}\sum_{n=1}^{N} e^{2\sqrt{-1}\pi k(x+n\alpha)} = \frac{1}{N} \times \frac{1 - e^{2\sqrt{-1}\pi Nk\alpha}}{1 - e^{2\sqrt{-1}\pi k\alpha}} \times e^{2\sqrt{-1}\pi k(x+\alpha)} = O\left(\frac{1}{N}\right), \quad N \to \infty.
$$

$\int_{\mathbb{T}^1} \exp(2k\pi\sqrt{-1}\,x)dx = 0$ であるから，大数の法則の収束速度が $O(N^{-1})$ であることを示している．一般の関数のときはフーリエ (Fourier) 級数で近似すればよい．このとき，滑らかな関数ほどフーリエ係数が速く 0 に収束するので，その場合の大数の法則も，ほとんど $O(N^{-1})$ に近い速さで収束するのである．

RWS は $\alpha \in \mathbb{T}^1$ を $x \in \mathbb{T}^1$ とともにランダムに選ぶ．選ばれる α は確率 1 で無理数であり，従って，上の段落で述べたことが確率 1 で成り立っている．このことから想像されることは，RWS に関する大数の法則は i.i.d.-サンプリングの場合より収束が早いであろう，ということである．実際，$1 \leq p < 2$ なる p に対しては，RWS の p 次平均誤差に関して次の極限定理がある．

定理 19 ([7, 26]) 2 乗可積分関数 $f : \mathbb{T}^1 \to \mathbb{R}$ および $1 \leq p < 2$ に対して

$$
\lim_{N\to\infty} \iint_{\mathbb{T}^1\times\mathbb{T}^1} \left| \frac{1}{\sqrt{N}} \sum_{n=1}^{N} \left(f(x+n\alpha) - \int_{\mathbb{T}^1} f(y)dy \right) \right|^p d\alpha\, dx = 0.
$$

従って，任意の $\rho > 0$ について

$$
\lim_{N\to\infty} \mathbb{P}^2 \left\{ (x,\alpha) \in \mathbb{T}^2 \;\middle|\; \left| \frac{1}{\sqrt{N}} \sum_{n=1}^{N} \left(f(x+n\alpha) - \int_{\mathbb{T}^1} f(y)dy \right) \right| > \rho \right\} = 0. \tag{110}
$$

すなわち，標本平均の中心極限定理のスケーリングによる極限分布は退化する．

証明. 簡単のため $\int_{\mathbb{T}^1} dx\, f(x) = 0$ を仮定する．$M \in \mathbb{N}$ に対して関数 $F_M : \mathbf{T}^k \to \mathbb{R}$ を次のように定義する:

$$
f_M(t) := \sum_{|l|\leq M} \widehat{f}(l) e^{2\sqrt{-1}\pi lt},
$$

ただし $\widehat{f}(l)$ は F のフーリエ (Fourier) 係数，すなわち

$$
\widehat{f}(l) = \int_{\mathbb{T}^1} dt\, f(t) e^{-2\sqrt{-1}\pi lt}.
$$

$\int_{\mathbb{T}^1} dt f(t) = 0$ から $\widehat{f}(0) = 0$ が従うことに注意せよ．任意の $1 < p < 2$ を固定する．三角不等式，ヘルダー (Hölder) の不等式，および定理 18 によって

$$\begin{aligned}
\left\| \frac{1}{\sqrt{N}} \sum_{n=1}^{N} f(x+n\alpha) \right\|_p &:= \left(\int\int_{\mathbb{T}^1 \times \mathbb{T}^1} d\alpha dx \left| \frac{1}{\sqrt{N}} \sum_{n=1}^{N} f(x+n\alpha) \right|^p \right)^{\frac{1}{p}} \\
&\leq \left\| \frac{1}{\sqrt{N}} \sum_{n=1}^{N} f_M(x+n\alpha) \right\|_p + \left\| \frac{1}{\sqrt{N}} \sum_{n=1}^{N} (f - f_M)(x+n\alpha) \right\|_p \\
&\leq \left\| \frac{1}{\sqrt{N}} \sum_{n=1}^{N} f_M(x+n\alpha) \right\|_p + \left\| \frac{1}{\sqrt{N}} \sum_{n=1}^{N} (f - f_M)(x+n\alpha) \right\|_2 \\
&= \left\| \frac{1}{\sqrt{N}} \sum_{n=1}^{N} f_M(x+n\alpha) \right\|_p + \sqrt{\mathbf{Var}(f - f_M)}. \qquad (111)
\end{aligned}$$

(111) の最後の辺の第一項を詳しく計算しよう．f_M の定義によって

$$\frac{1}{\sqrt{N}} \sum_{n=1}^{N} f_M(x+n\alpha) = \sum_{0<|l|\leq M} \left(\widehat{f}(l) e^{2\sqrt{-1}\pi l x} \times \frac{1}{\sqrt{N}} \sum_{n=1}^{N} e^{2\sqrt{-1}\pi n l \alpha} \right)$$

だから，$L^p(\mathbb{T}^2, d\alpha dx)$-ノルムをとれば

$$\begin{aligned}
\left\| \frac{1}{\sqrt{N}} \sum_{n=1}^{N} f_M(x+n\alpha) \right\|_p &\leq \sum_{0<|l|\leq M} \left| \widehat{f}(l) \right| \left(\int_{\mathbb{T}^1} d\alpha \left| \frac{1}{\sqrt{N}} \sum_{n=1}^{N} e^{2\sqrt{-1}\pi n l \alpha} \right|^p \right)^{1/p} \\
&= \sum_{0<|l|\leq M} \left| \widehat{f}(l) \right| \left(\int_{\mathbb{T}^1} d\alpha \left| \frac{1}{\sqrt{N}} \sum_{n=1}^{N} e^{2\sqrt{-1}\pi n \alpha} \right|^p \right)^{1/p},
\end{aligned}$$

ここで変換 $\mathbb{T}^1 \ni \alpha \mapsto l\alpha \in \mathbb{T}^1$ がルベーグ測度を保存することを用いた．そして

$$\begin{aligned}
\int_{\mathbb{T}^1} d\alpha \left| \frac{1}{\sqrt{N}} \sum_{n=1}^{N} e^{2\sqrt{-1}\pi n \alpha} \right|^p &= \int_0^{\frac{1}{2}} d\alpha \left| \frac{1}{\sqrt{N}} \sum_{n=1}^{N} e^{2\sqrt{-1}\pi n \alpha} \right|^p + \int_{\frac{1}{2}}^{1} d\alpha \left| \frac{1}{\sqrt{N}} \sum_{n=1}^{N} e^{2\sqrt{-1}\pi n \alpha} \right|^p \\
&= 2 \int_0^{\frac{1}{2}} d\alpha \left| \frac{1}{\sqrt{N}} \sum_{n=1}^{N} e^{2\sqrt{-1}\pi n \alpha} \right|^p \\
&= 2 \int_0^{\frac{1}{2}} d\alpha \left| \frac{1}{\sqrt{N}} \frac{\sin \pi N \alpha}{\sin \pi \alpha} \right|^p \\
&= 2 \int_0^{\frac{N}{2}} \frac{dt}{N} \left| \frac{1}{\sqrt{N}} \frac{\sin \pi t}{\sin \pi \frac{t}{N}} \right|^p \quad (\text{変数変換 } N\alpha = t) \\
&= 2 \left(\frac{1}{N} \right)^{\frac{p}{2}+1} \int_0^{\frac{N}{2}} dt \left| \frac{\pi \frac{t}{N}}{\sin \pi \frac{t}{N}} \right|^p \left| \frac{\sin \pi t}{\pi t} \right|^p N^p \\
&= 2 \left(\frac{1}{N} \right)^{1-\frac{p}{2}} \int_0^{\frac{N}{2}} dt \left| \frac{\pi \frac{t}{N}}{\sin \pi \frac{t}{N}} \right|^p \left| \frac{\sin \pi t}{\pi t} \right|^p \\
&< \left(\frac{1}{N} \right)^{1-\frac{p}{2}} 2 \left(\frac{\pi}{2} \right)^p \int_0^{\infty} dt \left| \frac{\sin \pi t}{\pi t} \right|^p,
\end{aligned}$$

ここで $0 < y < \pi/2$ ならば $y/\sin y < \pi/2$ であることを用いた．これより

$$\left\| \frac{1}{\sqrt{N}} \sum_{n=1}^{N} f_M(x+n\alpha) \right\|_p \leq \sum_{0<|l|\leq M} \left|\widehat{f}(l)\right| \left(\int_{\mathbb{T}^1} d\alpha \left| \frac{1}{\sqrt{N}} \sum_{n=1}^{N} e^{2\sqrt{-1}\pi n\alpha} \right|^p \right)^{1/p} \underset{N\to\infty}{\longrightarrow} 0.$$

従って結局

$$\overline{\lim_{N\to\infty}} \left\| \frac{1}{\sqrt{N}} \sum_{n=1}^{N} f(x+n\alpha) \right\|_p \leq \sqrt{\mathbf{Var}(f - f_M)} \underset{M\to\infty}{\longrightarrow} 0.$$

□

数値積分の観点からは極限 (110) は中心極限定よりもずっと望ましい．(110) によれば，サンプル数 N が大きければ大きいほど，RWS の誤差が $\rho/\sqrt{N} > 0$ を超えるペア (x, α) は，ますます少なくなる．これは大変望ましい性質といえる．

用心深い人のために少し付け加えよう．じつは，分散は

$$\int_{\mathbb{T}^2} \left| \frac{1}{N} \sum_{n=1}^{N} f(x+n\alpha) - \int_{\mathbb{T}^1} f(y)dy \right|^2 dxd\alpha = \frac{\mathbf{Var}(f)}{N} \tag{112}$$

を満たすので，もし，運悪く (110) の左辺の事象が起こってしまうと，サンプリングの誤差は非常に大きくなることが考えられる．有限精度 2^{-m} の RWS の場合で考えよう．もし，$\alpha = 0$ と選んでしまったら，すべての n で $X_n(\omega) = x$ になってしまい，とんでもなく悪いサンプリング点を生成してしまう (cf. 注意 11)．その確率は 2^{-m} であり，i.i.d.-サンプリングの場合の同様の事象の起こる確率よりずっと大きい．いい換えれば，前者のサンプリングでとんでもなく大きな誤差を生ずる確率が後者よりずっと大きいのである．一方で，(112) を満たすから，このような事象が起こらないとき，前者のサンプリングの誤差は後者より小さくなければならない．

もっとも，このような「とんでもなく大きな誤差を生ずる確率」は，m がそこそこ大きければ，大変小さいので実用上はまったく心配することはなかろう．よって結論として，RWS は i.i.d.-サンプリングより数値積分に適していると考えられるのである．

注意 12　非常に複雑な被積分関数のときは，RWS の生成するサンプルがほとんど独立のようになってしまって，RWS は i.i.d.-サンプリングと収束速度がほとんど変わらなくなることがある．たとえば，§ 3.2.3 で見た従属性消滅定理 (定理 13) はその例を与えている．

4.3　模倣可能な確率変数

4.3.1　確率変数のルベーグ確率空間上での実現

ある確率変数 W をモンテカルロ法で扱うには，W と同じ分布を持つ確率変数 f を硬貨投げの確率過程を基にして構成しなければならない．まず，次の定理から始めよう．

定理 20　W を確率空間 $(\Omega, \mathcal{F}, P)$ 上で定義された確率変数とする．このとき，ルベーグ確率空間上で定義された確率変数 f で W と同分布なものが存在する．

証明. W の分布関数 $F_W(x) := P(W \le x)$, $x \in \mathbb{R}$, を利用して

$$f(x) := \inf\{t \in \mathbb{R} \mid F_W(t) \ge x\}, \quad 0 \le x < 1, \tag{113}$$

とすれば，これがルベーグ確率空間上の確率変数として W と同分布になる．実際，$a \in \mathbb{R}$ に対して

$$\begin{aligned} \mathbb{P}(f(x) \le a) &= \mathbb{P}\left(\inf F_W^{-1}([x,1)) \le a\right) \\ &= \mathbb{P}\left(x \le F_W(a)\right) \\ &= F_W(a). \end{aligned}$$

□

定理 20 の条件を満たす f は一意的に定まるものではない．(113) の f は一つの例に過ぎない．

ルベーグ確率空間上の確率変数 f は

$$f(x) = f\left(\sum_{i=1}^{\infty} d_i(x) 2^{-i}\right), \quad x \in \mathbb{T},$$

によっていつでも硬貨投げの確率過程 $\{d_i\}_{i=1}^{\infty}$ の汎関数[46]と考えることができるから，定理 20 によって，任意の確率変数 W に対して硬貨投げの確率過程の汎関数であるような W と同分布の確率変数が存在する．

しかし，このことをもって任意の確率変数 W がいつもモンテカルロ法で扱えるわけではない．普通は W の分布関数が明示的に求まることは少なく，従って (113) を実際に計算できることは稀である．また，モンテカルロ法の実施において，硬貨投げの確率過程は無限 $\{0,1\}$-列が一挙にして与えられるのではなく，第一項から順に生成されていくものなので，$f(x)$ の値は確率 1 で有限回の硬貨投げの結果から計算されなければならない．

一般に，確率変数 W と同分布の (あるいは計算の丸め誤差を多少認めた上でほぼ同分布の) 確率変数をモンテカルロ法で扱い得る形で構成することを**模倣** (simulation) という．W が模倣可能であるための条件を求めていこう．

4.3.2 停止時刻に関して可測な確率変数

ある $m \in \mathbb{N}$ に対して $\mathcal{B}_m$-可測であるような確率変数は明らかに模倣可能である．[47] 一方，どのような m に対しても $\mathcal{B}_m$-可測とならないような $\mathcal{B}$-可測関数 (無限精度関数と呼ぼう) でも，場合によっては実際的な計算の上で扱うことが可能である．次の例を見よ．

[46] 無限個の変数を持つ関数を汎関数という．ここでは $W(x)$ が 無限個の値 $d_i(x)$, $i = 1, 2, \ldots$ の関数と見なせる，ということ．

[47] もちろん，m が天文学的に巨大だと現実のコンピュータでは扱うことができなくなる．ここでいう模倣可能性は，あるチューリング機械で計算できる，という意味である．

例 10 (到達時刻) ルベーグ確率空間上で定義された確率変数

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

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 は非有界な無限精度関数である．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．

例 10 を含むような，モンテカルロ法で実際に扱い得る無限精度関数の一般的なクラスを考えよう．

定義 20　関数 $\tau : \mathbb{T}^1 \to \mathbb{N} \cup \{\infty\}$ は

$$\{\tau \leq m\} := \{x \in \mathbb{T}^1 \,|\, \tau(x) \leq m\} \in \mathcal{B}_m, \quad \forall m \in \mathbb{N}.$$

を満たすとき，$\{\mathcal{B}_m\}_m$-**停止時刻** (cf. [2]) 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．$\mathcal{B}_\tau$-可測関数を以下簡単のため τ-**可測**な関数という．また，$L^p(\mathbb{T}^1, \mathcal{B}_\tau, \mathbb{P})$ を単に $L^p(\mathcal{B}_\tau)$ と書く．

補題 13　関数 $f : \mathbb{T}^1 \to \mathbb{R} \cup \{\pm\infty\}$ が τ-可測な関数であるための必要十分条件は

$$f(x) = f\left(\lfloor x \rfloor_{\tau(x)}\right), \qquad x \in \mathbb{T}^1, \tag{114}$$

となることである．

証明. 必要性:もし f が τ-可測ならば，各 $m \in \mathbb{N}$, $t \in \mathbb{R}$ に対して，$\{\tau \leq m \text{ かつ } f \leq t\} \in \mathcal{B}_m$ である．このことから $\tau(x) \leq m$ ならば $f(x) = f\left(\lfloor x \rfloor_m\right)$ が分かる．従って

$$\begin{aligned}
f(x) &= \sum_{m=1}^{\infty} f(x) \mathbf{1}_{\{\tau(\bullet)=m\}}(x) + f(x) \mathbf{1}_{\{\tau(\bullet)=\infty\}}(x) \\
&= \sum_{m=1}^{\infty} f\left(\lfloor x \rfloor_m\right) \mathbf{1}_{\{\tau(\bullet)=m\}}(x) + f\left(\lfloor x \rfloor_\infty\right) \mathbf{1}_{\{\tau(\bullet)=\infty\}}(x) \\
&= \sum_{m=1}^{\infty} f\left(\lfloor x \rfloor_{\tau(x)}\right) \mathbf{1}_{\{\tau(\bullet)=m\}}(x) + f\left(\lfloor x \rfloor_{\tau(x)}\right) \mathbf{1}_{\{\tau(\bullet)=\infty\}}(x) \\
&= f\left(\lfloor x \rfloor_{\tau(x)}\right) \sum_{m=1}^{\infty} \mathbf{1}_{\{\tau(\bullet)=m\}}(x) + f\left(\lfloor x \rfloor_{\tau(x)}\right) \mathbf{1}_{\{\tau(\bullet)=\infty\}}(x) = f\left(\lfloor x \rfloor_{\tau(x)}\right).
\end{aligned}$$

十分性：もし f が (114) を満たせば，各 $m \in \mathbb{N}$, $t \in \mathbb{R}$ に対して

$$\{f \leq t\} \cap \{\tau \leq m\} = \{f\left(\lfloor \bullet \rfloor_{\tau(\bullet)}\right) \leq t\} \cap \{\tau \leq m\} = \{f\left(\lfloor \bullet \rfloor_m\right) \leq t\} \cap \{\tau \leq m\} \in \mathcal{B}_m$$

だから，f は τ-可側である． □

例 10 の確率変数 σ は $\{\mathcal{B}_m\}_m$-停止時刻であり，もちろん，σ は σ-可測である．

定理 21　τ を確率 1 で有限な $\{\mathcal{B}_m\}_m$-停止時刻とするとき，任意の τ-可測な f はモンテカルロ法で扱うことができる．[48]

証明.　硬貨投げの確率過程 $\{d_i(x)\}_{i=1}^{\infty}$ は $d_1(x), d_2(x), \ldots$ の順で供給されるとする．f が $\{\mathcal{B}_m\}_m$-停止時刻 τ に関して可測のとき，次のようなアルゴリズムを考える:

1. $m = 1$ とする．
2. $t := \sum_{i=1}^{m} 2^{-i} d_i(x)\,(= \lfloor x \rfloor_m)$ とする．
3. もし $\tau(t) \le m$ ならば，$f(t)$ を出力し，終了する．
4. もし $\tau(t) > m$ ならば $m := m + 1$ として 2. に戻る．

τ が確率 1 で有限なので上のアルゴリズムは必ず有限時間内に終了し，そのときの出力は $f(x)$ である．　□

さて，モンテカルロ法ではランダム源を $\mathbb{T}^1$-値一様分布に従う i.i.d. 確率変数列 $\{Z_l\}_{l\in\mathbb{N}}$ として議論することが多い．その文脈で停止時間に関する可測性は次のような仮定として述べることができる．

仮定 1　W を計算するためには確率 1 で，有限個の $Z_1, \ldots, Z_L$ しか必要でない，と仮定する．ここで，Z_l たちの個数 L は確率変数であるが，各 $l \in \mathbb{N}$ に対して，$Z_1, \ldots, Z_l$ が与えられたとき，次の Z_{l+1} が W を計算するために必要かどうかが，$Z_{l'}$, $l' \ge l+1$，に関して何の知識がなくても，判断できなければならない．

仮定 1 が成り立たなければ，W を計算するために Z_l たちを永遠に生成し続ける事態が生じるから，この仮定は，確率変数をモンテカルロ法で扱うためには不可欠であることが分かる．

実際には実数計算が有限精度，たとえば 2^{-K}，で行われることを考慮に入れて，$\{Z_l\}_{l\in\mathbb{N}}$ の代わりに D_K-値一様分布に従う i.i.d. $\{Z_l^{(K)}\}_{l\in\mathbb{N}}$ を $(\mathbb{T}^1, \mathcal{B}, \mathbb{P})$ 上で

$$Z_n^{(K)} := \sum_{i=1}^{K} 2^{-i} d_{(n-1)K+i}, \qquad n \in \mathbb{N}, \tag{115}$$

すなわち

$$\begin{aligned}
Z_1^{(K)} &= \frac{1}{2}d_1 + \frac{1}{2^2}d_2 + \cdots + \frac{1}{2^K}d_K \\
Z_2^{(K)} &= \frac{1}{2}d_{K+1} + \frac{1}{2^2}d_{K+2} + \cdots + \frac{1}{2^K}d_{2K} \\
Z_3^{(K)} &= \frac{1}{2}d_{2K+1} + \frac{1}{2^2}d_{2K+2} + \cdots + \frac{1}{2^K}d_{3K} \\
&\vdots
\end{aligned}$$

[48] 詳しくは，硬貨投げの確率過程を入力とし f を計算するチューリング機械が存在する，ということ．

のように実現する．そして，W が仮定 1 を満たすとき

$$\tau(x) := \inf\{\, lK \,|\, W(x) \text{ は } Z_1^{(K)}(x), \ldots, Z_l^{(K)}(x) \text{ から計算される} \,\}$$

とおけば，τ は $\{\mathcal{B}_m\}_m$-停止時刻であり，W は τ-可測となる．[49]

例 11　W が常に一定の個数の Z_l たちから計算される場合，すなわち L が定数の場合は，W は有限精度であり明らかに仮定 1 を満たす．

次の例のように，停止時刻 τ は実際の数値計算においては，多くの場合，あからさまに意識する必要がない．

例 12 (フォン・ノイマンの棄却法 [16])　$[0,1)$-値一様分布に従う独立な確率変数列 $\{Z_l\}_{l=1}^{\infty}$ が与えられたとき，$[a,b]$ 上で有界可測なある確率密度関数 $p(x)$ の分布に従う確率変数 W を生成したいとせよ．$M > 0$ を p の上界の一つとする．(ξ,η) を $[a,b]\times[0,M]$ で一様に分布する確率変数とするとき，

$$\Pr(\xi \in [x, x+dx) \,|\, p(\xi) \geq \eta) \;=\; p(x)dx, \quad \text{a.e.-}x,$$

であることに注意する．そこで，次のようなアルゴリズムを考える．

1. $l := 1$ とする．
2. $(\xi,\eta) := ((b-a)Z_{2l-1} + a, MZ_{2l})$ とする．
3. もし $p(\xi) \geq \eta$ ならば，$W := \xi$ として，これを出力する．
4. もし $p(\xi) < \eta$ ならば $l := l+1$ として 2. に戻る．

このアルゴリズムで得られる確率変数 W の分布は望みのものになっている．このとき W は仮定 1 を満たしている．

例 13 (例 10 の言い換え)　$\{Y_n\}_{n=1}^{\infty}$ を

$$Y_n := \begin{cases} 0, & \text{if } Z_n \in [0,1/2) \\ 1, & \text{if } Z_n \in [1/2,1) \end{cases} \qquad n = 1,2,\ldots,$$

とする．$\{Y_n\}_{n=1}^{\infty}$ は硬貨投げの確率過程である．このとき

$$W := \inf\{\, n \geq 1 \,|\, Y_1 + Y_2 + \cdots + Y_n = 5\}$$

とすれば W は仮定 1 を満たす．W 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．

[49]従って定理 21 の逆も成り立つ．

4.3.3 停止時刻に関して可測な関数の i.i.d.-サンプリング

停止時刻に関して可測な f に対して i.i.d.-サンプリングによって数値積分を実行することができる.

定理 22 τ を $\mathbb{P}(\tau < \infty) = 1$ なる $\{\mathcal{B}_m\}_m$-停止時刻とし，f は τ-可測な関数とする.

$$y_n(x) := 2^{\sum_{i=1}^{n-1}\tau(y_i(x))}x, \quad x \in \mathbb{T}^1, \quad n \in \mathbb{N},$$

とすれば，$(\mathbb{T}^1, \mathcal{B}, \mathbb{P})$ 上の確率変数列 $\{f(y_n)\}_{n\in\mathbb{N}}$ は i.i.d. であり，その共通の分布は f 自身の分布に等しい.

証明. $y_1(x) = x$ だから，明らかに $f(y_1)$ と f の分布は等しい. (114) より，$f(y_n) = f(\lfloor y_n \rfloor_{\tau(y_n)})$ だから，$\{\lfloor y_n \rfloor_{\tau(y_n)}\}_{n\in\mathbb{N}}$ が i.i.d. であることを示せばよい.[50]

任意の $n \in \mathbb{N}$，任意の $a_1, \ldots, a_n \in D = \cup_{m\in\mathbb{N}} D_m$ に対して

$$\begin{aligned}
&\mathbb{P}\left(\lfloor y_i \rfloor_{\tau(y_i)} = a_i,\ 1 \le i \le n\right) \\
&= \sum_{m_1,\ldots,m_{n-1}\in\mathbb{N}} \mathbb{P}\left(\lfloor y_i \rfloor_{m_i} = a_i, \tau(y_i) = m_i,\ 1 \le i \le n-1,\ \lfloor y_n \rfloor_{\tau(y_n)} = a_n,\right) \\
&= \sum_{m_1,\ldots,m_{n-1}\in\mathbb{N}} \mathbb{P}\left(\lfloor y_i \rfloor_{m_i} = a_i, \tau(y_i) = m_i,\ 1 \le i \le n-1,\ \left\lfloor 2^{\sum_{i=1}^{n-1} m_i} x \right\rfloor_{\tau\left(2^{\sum_{i=1}^{n-1} m_i} x\right)} = a_n,\right) \\
&= \sum_{m_1,\ldots,m_{n-1}\in\mathbb{N}} \mathbb{P}\left(\lfloor y_i \rfloor_{m_i} = a_i, \tau(y_i) = m_i,\ 1 \le i \le n-1\right) \mathbb{P}\left(\left\lfloor 2^{\sum_{i=1}^{n-1} m_i} x \right\rfloor_{\tau\left(2^{\sum_{i=1}^{n-1} m_i} x\right)} = a_n\right) \\
&= \sum_{m_1,\ldots,m_{n-1}\in\mathbb{N}} \mathbb{P}\left(\lfloor y_i \rfloor_{m_i} = a_i, \tau(y_i) = m_i,\ 1 \le i \le n-1\right) \mathbb{P}\left(\lfloor x \rfloor_{\tau(x)} = a_n\right) \\
&= \mathbb{P}\left(\lfloor y_i \rfloor_{\tau(y_i)} = a_i,\ 1 \le i \le n-1\right) \mathbb{P}\left(\lfloor x \rfloor_{\tau(x)} = a_n\right).
\end{aligned}$$

この計算を繰り返せば,

$$\mathbb{P}\left(\lfloor y_i \rfloor_{\tau(y_i)} = a_i,\ 1 \le i \le n\right) = \prod_{i=1}^{n} \mathbb{P}\left(\lfloor x \rfloor_{\tau(x)} = a_i\right).$$

□

τ を $\{\mathcal{B}_m\}_m$-停止時刻とし，f を τ-可測な関数とするとき定理 22 から

$$\frac{1}{N}\sum_{i=1}^{N} f(y_i(x)) \tag{116}$$

によって f の数値積分を行うことができる.

[50] 確率過程論に詳しい読者には，以下の証明の要点は,「i.i.d. は強マルコフ性を持つ」という事実に基づくことが分かるだろう.

注意 13　(116) を計算するための計算時間は，必要とされるランダムビットの個数にほぼ比例すると考えられる．従ってその平均は $N\mathbf{E}[\tau]$ に比例する．その際，もし $\mathbf{E}[\tau^2] = \infty$ であった場合，実際の計算時間は，その平均から大きく離れる確率があまり小さくならないので，とんでもなく大きくなる可能性がある．そこで，実用面からは $\mathbf{E}[\tau^2] < \infty$ であるのが望ましい．たとえば，もし τ がフォン・ノイマンの棄却法 (例 12) に付随した停止時間であるときには τ の分布は幾何分布であるので $\mathbf{E}[\tau^2] < \infty$ である．

停止時刻に関して可測な確率変数の i.i.d.-サンプリングは，一見，複雑に思えるかもしれないが，実際は簡単なアルゴリズムで計算される．数値積分を行いたい確率変数 W は仮定 1 を満たすとする．ただし，仮定 1 で登場する i.i.d. 確率変数列 $\{Z_1, Z_2, \ldots\}$ は以下では，計算機の精度 (2^{-K} とする)，$Z_l^{(K)} \in D_K$ と考える．次に，仮想的関数

- function Random$_m$: D_m-valued;

が，D_m-値一様分布に従い，以前に生成されたあらゆる確率変数と独立な確率変数を返す，と仮定する．(実際には，この関数の代わりに疑似乱数生成器を用いる．)

そして数値積分のために生成する W のサンプルの総数を N とする．

i.i.d.-サンプリングのアルゴリズム

- メイン・ルーチン

```
function Mean_of_W : Real;
begin
  S := 0.0;
  For i := 1 to N do
    begin
      Z :=Random_K;
      W を計算しようと試みる;
      while ( W を計算するためにさらに別の Z が必要 ) do
        begin
          Z :=Random_K;
          W を計算しようと試みる;
        end
      S := S + W ;
    end;
  result:= S/N;
end;
```

関数 Mean_of_W は変数 result に代入された値，すなわち，S/N を返す．ここで W のサンプルの生成に必要な Z_l はすべて関数 Random$_K$ により生成される．

4.4 動的ランダム-ワイル-サンプリング

ここでは，停止時刻に関して可測な関数に適用可能なペアごとに独立なサンプリング法を提案する．我々はそれを動的ランダム・ワイル・サンプリングと呼ぶ.

動的ランダム・ワイル・サンプリングのアルゴリズムは大変簡単なので，i.i.d.-サンプリングと同じくらい手軽な感覚でプログラムを作ることができるし (§ 5, [25])，ペアごとに独立なサンプル列の生成は十分に速い．動的ランダム・ワイル・サンプリングは i.i.d.-サンプリングが適用可能な場合ならばいつも適用可能だし，しかも i.i.d.-サンプリングよりずっと精度および信頼性が高い (§ 4.4.4).

ただし，動的ランダム・ワイル・サンプリングは一つのサンプルを生成する分のビット列を常にメモリ上に記憶しておく必要があるので，i.i.d-サンプリングよりメモリの消費量が大きいことに注意が必要である (§ 5.5.3).

4.4.1 定義と定理

τ を $\mathbb{P}(\tau < \infty) = 1$ なる $\{\mathcal{B}_m\}_m$-停止時刻とする．$j \in \mathbb{N}$ として

$$(x_l, \alpha_l) \in D_{K+j} \times D_{K+j} \subset \mathbb{T}^1 \times \mathbb{T}^1, \quad l \in \mathbb{N}, \tag{117}$$

を一様分布に従う独立確率変数列とする．確率変数 $\mathbf{x}_n$ を

$$\mathbf{x}_n := \sum_{l=1}^{\infty} 2^{-(l-1)K} \lfloor x_l + \nu_{n,l}\, \alpha_l \rfloor_K \in \mathbb{T}^1, \quad n = 1, \ldots, 2^{j+1}, \tag{118}$$

と定める．ただし，$\nu_{n,l}$ は次で定義される確率変数である:

$$\nu_{n,l} := \begin{cases} n & (l = 1) \\ \#\{\, 1 \le u \le n \mid \tau(\mathbf{x}_u) > (l-1)K \,\} & (l > 1) \end{cases} \tag{119}$$

τ が $\{\mathcal{B}_m\}_m$-停止時間であることから，$\nu_{n,l}$ および $\mathbf{x}_n$ は確かに定義される．このとき，次の定理が成り立つ.

定理 23 ([21]) f が τ-可測ならば，確率変数列 $\{f(\mathbf{x}_n)\}_{n=1}^{2^{j+1}}$ は同分布かつペアごとに独立である．その共通の分布は f を $(\mathbb{T}^1, \mathcal{B}, \mathbb{P})$ 上の確率変数とした考えたときの分布と同じである.

$\{\mathbf{x}_n\}_{n=1}^{2^{j+1}}$ はすべて一様分布するが，ペアごとに独立ではないことに注意せよ (補題 14)．定理 23 は，それらが τ-可測関数 f に代入されるとペアごとに独立になることを主張している.

定義 21 ([21]) $\mathbf{E}[\tau] < \infty$ と $f \in L^1(\mathcal{B}_\tau)$ を仮定する．このとき，$\{\mathbf{x}_n\}_{n=1}^{2^{j+1}}$ によるサンプリング法，すなわち，$\mathbf{E}[f]$ の近似を

$$\frac{1}{N} \sum_{n=1}^{N} f(\mathbf{x}_n), \quad 1 \le N \le 2^{j+1},$$

によって行う方法を**動的ランダム・ワイル・サンプリング** (dynamic random Weyl sampling, 以下 DRWS と略す) という.

“動的” という語はサンプル点 $\{\mathbf{x}_n\}_{n=1}^{2^{j+1}}$ が被積分関数，詳しくは停止時間 τ によって変化することを示唆している．

系 3　$f \in L^2(\mathcal{B}_\tau)$ ならば，DRWS は i.i.d.-サンプリングと同じ平均 2 乗誤差を持つ．すなわち，

$$\mathbf{E}\left[\left|\frac{1}{N'}\sum_{n=0}^{N'-1} f(\mathbf{x}_n) - \mathbf{E}[f]\right|^2\right] = \frac{\mathbf{Var}[f]}{N'}, \quad 1 \le N' \le 2^{j+1}. \tag{120}$$

注意 14　注意 13 で述べたように，$\mathbf{E}[\tau^2] < \infty$ であることが望ましい．

4.4.2　定理 23 の証明

以下では $1 \le n < n' \le 2^{j+1}$ を固定して考える．$m(n,n') := \max\{l\,;\, \nu_{n,l} < \nu_{n',l}\}$ とおく．このとき $i = 1,\dots,l$ に対して $\nu_{n,l} < \nu_{n',l}$ より $\nu_{n,i} < \nu_{n',i}$ が従うから

$$m(n,n') = \max\{l\,;\, \nu_{n,i} < \nu_{n',i},\ i = 1,\dots,l\}. \tag{121}$$

そこで

$$\widetilde{\mathbf{x}}_{n'} := \sum_{l=1}^{\infty} 2^{-(l-1)K}\lfloor x_l + \widetilde{\nu}_{n',l}\,\alpha_l\rfloor_K, \quad \widetilde{\nu}_{n',l} := \begin{cases} \nu_{n',l} & (l \le m(n,n')) \\ n' & (l > m(n,n')) \end{cases} \tag{122}$$

とおく．

補題 14　(i) 各 $\mathbf{x}_n$ は $\mathbb{T}^1 = [0,1)$ で一様分布する．
(ii) $\mathbf{x}_n$ と $\widetilde{\mathbf{x}}_{n'}$ は独立である．

証明. (i) と (ii) を示すには，任意の $M \in \mathbb{N}$ に対して，確率変数たち

$$\lfloor x_l + \nu_{n,l}\,\alpha_l\rfloor_K, \quad \lfloor x_l + \widetilde{\nu}_{n',l}\,\alpha_l\rfloor_K, \qquad l = 1,\dots,M, \tag{123}$$

がすべて D_K で一様分布し，そして独立であることを示せばよい．

もし $l \ge 2$ ならば $\nu_{n,l}$ も $\widetilde{\nu}_{n',l}$ も $(x_1,\alpha_1),\dots,(x_{l-1},\alpha_{l-1})$ には依存するが，(x_l,α_l) とは独立であることに注意せよ．また，常に $\nu_{n,l} < \widetilde{\nu}_{n',l}$ であることにも注意せよ．

P をすべての (x_l,α_l) を支配する確率測度とする．任意の $s_1,t_1,\dots,s_M,t_M \in D_K$ に対して

$$\begin{aligned}
&P(\lfloor x_l + \nu_{n,l}\,\alpha_l\rfloor_K < s_l,\ \lfloor x_l + \widetilde{\nu}_{n',l}\,\alpha_l\rfloor_K < t_l,\ l = 1,\dots,M) \\
&= \sum_{\substack{p_l < p_l';\\ l=1,\dots,M}} P\left(\begin{array}{ll} \lfloor x_l + p_l\alpha_l\rfloor_K < s_l, & \nu_{n,l} = p_l, \\ \lfloor x_l + p_l'\alpha_l\rfloor_K < t_l, & \widetilde{\nu}_{n',l} = p_l', \end{array}\ l = 1,\dots,M\right) \\
&= \sum_{\substack{p_l < p_l';\\ l=1,\dots,M}} P\left(\begin{array}{l} \lfloor x_M + p_M\alpha_M\rfloor_K < s_M \\ \lfloor x_M + p_M'\alpha_M\rfloor_K < t_M \end{array}\right) \\
&\qquad \times P\left(\begin{array}{ll} \lfloor x_l + p_l\alpha_l\rfloor_K < s_l, & \nu_{n,l} = p_l, \\ \lfloor x_l + p_l'\alpha_l\rfloor_K < t_l, & \widetilde{\nu}_{n',l} = p_l', \end{array}\ l = 1,\dots,M-1,\ \begin{array}{l} \nu_{n,M} = p_M \\ \widetilde{\nu}_{n',M} = p_M' \end{array}\right).
\end{aligned}$$

$p_M \neq p'_M$ だから定理 17 により，事象 $\{\lfloor x_M + p_M\alpha_M\rfloor_K < s_M\}$ と $\{\lfloor x_M + p'_M\alpha_M\rfloor_K < t_M\}$ は独立である．よって

$$
\begin{aligned}
&= \sum_{\substack{p_l<p'_l;\\ l=1,\dots,M}} P\left(\lfloor x_M + p_M\alpha_M\rfloor_K < s_M\right) P\left(\lfloor x_M + p'_M\alpha_M\rfloor_K < t_M\right) \\
&\quad \times P\left(\begin{array}{ll} \lfloor x_l + p_l\alpha_l\rfloor_K < s_l, & \nu_{n,l} = p_l, \\ \lfloor x_l + p'_l\alpha_l\rfloor_K < t_l, & \widetilde{\nu}_{n',l} = p'_l, \end{array} \ l = 1,\dots,M-1, \ \begin{array}{l} \nu_{n,M} = p_M \\ \widetilde{\nu}_{n',M} = p'_M \end{array} \right) \\
&= s_M t_M \sum_{\substack{p_l<p'_l;\\ l=1,\dots,M-1}} P\left(\begin{array}{ll} \lfloor x_l + p_l\alpha_l\rfloor_K < s_l, & \nu_{n,l} = p_l, \\ \lfloor x_l + p'_l\alpha_l\rfloor_K < t_l, & \widetilde{\nu}_{n',l} = p'_l, \end{array} \ l = 1,\dots,M-1 \right) \\
&= s_M t_M P(\lfloor x_l + \nu_{n,l}\,\alpha_l\rfloor_K < s_l,\, \lfloor x_l + \widetilde{\nu}_{n',l}\,\alpha_l\rfloor_K < t_l,\, l = 1,\dots,M-1).
\end{aligned}
$$

この操作を続ければ，最終的に

$$
P(\lfloor x_l + \nu_{n,l}\,\alpha_l\rfloor_K < s_l,\, \lfloor x_l + \widetilde{\nu}_{n',l}\,\alpha_l\rfloor_K < t_l,\, l = 1,\dots,M) \;=\; \prod_{i=1}^{M} s_i t_i,
$$

となって補題の主張を得る． □

定理 23 の証明. 初めに補題 14(i) により，$f(\mathbf{x}_n)$ と f は同分布である．次に，(118) で n を n' で置き換えたものと (122) によって

$$
\lfloor \mathbf{x}_{n'}\rfloor_{m(n,n')K} \;=\; \lfloor \widetilde{\mathbf{x}}_{n'}\rfloor_{m(n,n')K}. \tag{124}
$$

$s := \lceil \tau(\mathbf{x}_{n'})/K\rceil$ とせよ．このとき，$\tau(\mathbf{x}_{n'}) > (s-1)K$ となるから $\nu_{n,s} < \nu_{n',s}$ である．従って，(121) より

$$
s \le m(n,n'). \tag{125}
$$

(124) と (125) から

$$
\lfloor \mathbf{x}_{n'}\rfloor_{sK} = \lfloor \widetilde{\mathbf{x}}_{n'}\rfloor_{sK} \tag{126}
$$

が従う．一方，$\tau(\mathbf{x}_{n'}) \le sK$ であることと τ は $\{\mathcal{B}_m\}_m$-停止時刻であることから，$\tau(\mathbf{x}_{n'})$ の値は $\lfloor \mathbf{x}_{n'}\rfloor_{sK}$ によって決まる．$\tau(\mathbf{x}_{n'}) \;=\; \tau(\lfloor \mathbf{x}_{n'}\rfloor_{sK})$. すると (126) より

$$
\tau(\widetilde{\mathbf{x}}_{n'}) = \tau(\mathbf{x}_{n'}) \;\le\; sK \tag{127}
$$

でなければならない．よって，(126) と (127) より

$$
\lfloor \mathbf{x}_{n'}\rfloor_{\tau(\mathbf{x}_{n'})} \;=\; \lfloor \widetilde{\mathbf{x}}_{n'}\rfloor_{\tau(\widetilde{\mathbf{x}}_{n'})}. \tag{128}
$$

f は τ-可測だから，(114) と (128) より，$f(\mathbf{x}_{n'}) \;=\; f(\widetilde{\mathbf{x}}_{n'})$ である．最後に補題 14(ii) から，$f(\mathbf{x}_n)$ と $f(\mathbf{x}_{n'})$ が独立であることが従う．これで証明が完了した． □

4.4.3 アルゴリズム

DRWS をどのように実装するかについて述べよう．§ 4.3.3 での設定を踏襲する．さらに，ここでは次を仮定する．

$$2^{j+1} \geq N. \tag{129}$$

DWRS のアルゴリズム

- 大域変数

  l : integer;
  $\{x_l, \alpha_l\}_l$: array (variable length) of $(D_{K+j})^2$-valued vectors;

- 関数

```
function First_RWS : D_K-valued;
begin
   l := 0;
   result:=Next_RWS;
end;
```

```
function Next_RWS : D_K-valued;
begin
   l := l + 1;
   if (x_l, α_l) がまだ生成されたいなかったら
      then
         begin
            x_l :=Random_{K+j};
            α_l :=Random_{K+j};
         end;
   x_l := x_l + α_l;
   result:= ⌊x_l⌋_K;
end;
```

- メイン・ルーチン

```
function Mean_of_W : Real;
begin
   S := 0.0;
   For i := 1 to N do
      begin
         Z :=First_RWS;
         W を計算しようと試みる;
         while ( W を計算するためにさらに別の Z が必要 ) do
            begin
               Z :=Next_RWS;
               W を計算しようと試みる;
            end
         S := S + W;
```

```
        end;
    result:= S/N;
end;
```

DRWS のメイン・ルーチンは i.i.d.-サンプリングのそれ (§ 4.3.3) に大変よく似ている．そのため，ユーザはまるで i.i.d.-サンプリングをしているかのように DRWS を実行することができる．違いはたった次の点だけである: DRWS では関数 Random_K を直接呼んで Z_l を生成するのはなく，二つの関数 First_RWS の Next_RWS のいずれかを呼んで生成する．前者は，最初の Z_1 を生成するときにだけ呼ばれ，後者は必要ならばその後の $Z_2, Z_3, \ldots$ を生成するのに呼ばれる．

ランダムな関数 Random_{K+j} は Next_RWS にだけ，それも，x_l と α_l がまだ生成されていない場合にのみ，それらを生成するために，呼ばれる．このように，DRWS には i.i.d.-サンプリングよりもずっと小さいランダムネスしか必要でない．

DRWS で使用されているランダムな関数が Random_{K+j} であることに注意せよ．ここでは，計算精度 2^{-K} のワイル変換を N 回だけ正確に実行するために，余分な j ビットが必要なのである．

注意 15　DRWS はモンテカルロ積分であり，これを “賭け” と考える立場からは，種 $\{(x_l, \alpha_l)\}_l$ は，プレーヤー，アリスが自分の意思で選ぶべきである．もちろん，それは上のアルゴリズムにおいて Random_{K+j} が呼び出されるたびに，アリスが $K+j$ ビットの “種” をコンピュータに入力とすることで可能ではある．しかし，実際にそのようにすると大変面倒なので，実装では補助的な疑似乱数生成器を用いて Random_{K+j} の代用とするのがよいだろう.[51] その補助的な疑似乱数生成器としてどのようなものが相応しいかは，§ 4.2.2 でのべたことと同じことが DRWS についてもいえる．

注意 16　確率変数 W が非常に多くの Z_l たちを必要とする確率が無視できないとき，沢山の (x_l, α_l) を記憶しておくために，DRWS は計算機のメモリを使い果たすかも知れない．具体的な実装におけるメモリの問題の解決については § 5.5.3 を見よ．

4.4.4　性能比較

例 13 の確率変数 W の平均を DRWS を用いて具体的に求めてみた．DRWS によって W のペアごとに独立なコピーを $10^3, 10^4, \ldots, 10^8$ 個生成してその標本平均をそれぞれ計算する．具体的には § 5.4.2 のサンプルプログラムで `#define SAMPLE_SIZE` を 10^3 から 10^8 まで変えて計算した．例 13 の確率変数 W の平均は 10 である．表 2 は，計算結果との誤差をサンプル数ごとに記録したものである．最後の行はサンプル数 10^8 の場合の計算時間である．比較のために，i.i.d.-サンプリング (ランダム源として，C の標準関数 `rand()`, MT(メルセンヌ・ツイスター，ここでは [8] に掲載のソースコードを使用), およびワイル変換による疑似乱数生成器 (§ 5.2 の `m90randombit()` を用いたもの)) についても同様の計算を行った.[52]

[51] § 5 における実装では補助的な疑似乱数生成器として，ワイル変換による疑似乱数生成器を用いている．

[52] 筆者の研究室のノートパソコンで計測した．コンパイラは BORLAND C++ COMPILER 5.5, COMMAND LINE TOOLS でとくに何のオプションも設定していない．

表 2: 性能比較

	rand-i.i.d.	MT-i.i.d.	m90-i.i.d.	DRWS
サンプル数 10^3 の誤差	0.15200000	-0.12500000	0.18600000	0.01700000
10^4	-0.05570000	0.02960000	0.03980000	-0.00030000
10^5	0.00650000	-0.01372000	-0.00170000	0.00076000
10^6	0.00470000	-0.00061300	-0.00382000	0.00007300
10^7	-0.00170760	0.00125260	0.00076940	0.00000560
10^8	-0.00095602	-0.00003483	0.00026567	-0.00000030
最終計算結果	9.99904398	9.99996517	10.00026567	9.99999970
計算時間 (秒)	13	27	87	35

誤差を比較すると DRWS の優位が見える．サンプルサイズを 10^7 としても DRWS の誤差は 0.00000560 に留まり，MT による i.i.d.-サンプリングのサンプル数 10^8 の場合より小さい．なお．サンプルサイズを 10^7 とした場合の DRWS の計算時間は 3 秒であった．このように DRWS の計算誤差が非常に小さい原因として，定理 19 のような極限定理が DRWS にも成り立つからではないか，と筆者は予想している．(特別な場合，たとえば付随する停止時刻 τ が定数の場合などには，それは正しいことが分かる ([26]).)

5 実装

5.1 例 9 の実装例

例 9 での計算例は，次の C プログラムによって得られた．このプリグラムは $S_{2^{19}}(\tilde{g}(\omega'))$ の値 204,650 を出力する．

```
/*============================================================*/
/*  file name: rws_example.c                                  */
/*============================================================*/
#include <stdio.h>

#define SAMPLE_SIZE  524288
#define M            128
#define M_PLUS_J     146    /*  J = 18  */

/* seed */
char xch[M_PLUS_J] =
   "1110110101" "1011101101" "0100000011" "0110101001" "0101000100"
   "0101111101" "1010000000" "1010100011" "0100011001" "1101111101"
   "1101010011" "1111001000" "1010010000" "1101011101" "001100";
char ach[M_PLUS_J] =
   "1100000111" "0111000100" "0001101011" "1001000001" "0010001000"
   "1010101101" "1110101110" "0010010011" "1000000011" "0101000110"
   "0101110010" "0101111110" "0110110000" "0101100001" "101110";

int  x[M_PLUS_J], a[M_PLUS_J];

void longadd(void)   /* x = x + a ( long digit addition ) */
{
 int i, s, carry = 0;
 for ( i = M_PLUS_J-1; i >= 0; i-- ){
   s = x[i] + a[i] + carry;
   if ( s >= 2 ) {carry = 1; s = s - 2; } else carry = 0;
   x[i] =  s;
 }
}

int maxLength(void)  /* count the longest run of 1's */
{
 int len = 0, count = 0, i;
 for ( i = 0; i <= M-1; i++ ){
  if ( x[i] == 0 ){ if ( len < count ) len = count; count = 0;}
  else count++ ;  /*  if x[i]==1 */
 }
 if ( len < count ) len = count;
 return len;
}
```

```
int main()
{
 int n, count = 0;
 for( n = 0; n <= M_PLUS_J-1; n++ ){
  if( xch[n] == '1' ) x[n] = 1; else x[n] = 0;
  if( ach[n] == '1' ) a[n] = 1; else a[n] = 0;
 }
 for ( n = 1 ; n <= SAMPLE_SIZE ; ++n ){
  longadd();
  if ( maxLength() >= 7 ) count++;
 }
 printf ( "%6d\n", count);
 return 0;
}
/*================ End of rws_sample.c ====================*/
```

5.2 モンテカルロ法のための汎用 C 言語ライブラリ

ワイル変換による疑似乱数生成器と動的ランダム・ワイル・サンプリング (DRWS) の C 言語ライブラリを紹介する．なお，[25] には最新版 (他言語仕様を含む) を公開しているので参照されたい．

このライブラリにおける実装の形態は以下のとおり：

- m90random
  $\alpha = (\sqrt{5} - 1)/2$，$m = 90$，$j = 60$ として § 3.2.1 の (22)(23)(24) による離散化の方法を用いてワイル変換による疑似乱数生成器を実装している．[53]

- DRWS
  $K = j = 31$ として § 4.4.1 の (117) (118)(119) によって DRWS のサンプル点列を生成する．ランダム源として m90random を用いている．

ソースコードは，random_sampler.c (ライブラリ本体) と random_sampler.h (ヘッダ) からなる．

5.2.1 random_sampler.c

```
/*=========================================================*/
/*  file name: random_sampler.c                            */
/*=========================================================*/
#include <stdlib.h>

#define  LIMIT_30  0x3fffffff
```

[53] m90 なる接頭辞はもちろんパラメータ m を 90 に設定していることによる．

```
#define  LIMIT_31  0x7fffffff
#define  CARRY_31  0x40000000
#define  CARRY_32  0x80000000

static unsigned long  omega[5]; /* for m90random */

struct data_pair_s {            /* for DRWS */
   unsigned long x1;
   unsigned long x2;
   unsigned long a1;
   unsigned long a2;
   struct data_pair_s *next;
};
typedef struct data_pair_s  data_pair_t;

static long location;
static long locmax;
static long locmaxmax=-1;
static data_pair_t  random_list;
static data_pair_t  *current_ptr;

/*=========================================================*/
/*   Functions for pseudo-random generation "m90random"     */


/* Initialization */
void  m90setseeds( unsigned long s0, unsigned long s1,
                   unsigned long s2, unsigned long s3,
                   unsigned long s4 )
{
   omega[0] = s0 & LIMIT_30; omega[1] = s1 & LIMIT_30;
   omega[2] = s2 & LIMIT_30; omega[3] = s3 & LIMIT_30;
   omega[4] = s4 & LIMIT_30;
}

/* Returns the current seeds */
void  m90getseeds( unsigned long *sp0, unsigned long *sp1,
                   unsigned long *sp2, unsigned long *sp3,
                   unsigned long *sp4 )
{
   *sp0 = omega[0]; *sp1 = omega[1]; *sp2 = omega[2];
   *sp3 = omega[3]; *sp4 = omega[4];
}

/* Generates a random bit */
char  m90randombit(void)
{
   static unsigned long alpha[5] = { /* Data of (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_31 ){ omega[j] &= LIMIT_30; omega[--j]++; }
      else --j;
   }
   omega[0] += alpha[0]; omega[0] &= LIMIT_30;
   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 >> 2 );
   data_byte ^= ( data_byte >> 1 );
   return( 1 & data_byte );
}

/* Generates a 31 bit random integer */
unsigned long m90random31(void)
{
   int j;
   unsigned long b=0;
   for ( j=0; j<30; j++ ) { b |= m90randombit(); b <<= 1; }
   b |= m90randombit();
   return b;
}

/* Generates a 31 bit random real in [0,1) */
double m90randomu(void)
{
   return (double)m90random31()/CARRY_32;
}

/*============================================================*/
/*   Functions for dynamic random Weyl sampling "DRWS"      */

/* Initialization */
void init_drws(void)
{
   locmax = -1; location = -1; random_list.next = 0;
}

/* Finalization */
```

```
void end_drws(void)
{
   data_pair_t *previous_ptr;
   if (random_list.next != 0){
      current_ptr = random_list.next;
      previous_ptr = &random_list;
      while (current_ptr -> next !=0){
         previous_ptr = current_ptr;
         current_ptr = current_ptr -> next;
      }
      free(current_ptr);
      previous_ptr -> next = 0;
      end_drws();
   }
}

/* Returns the locmax */
long get_locmax(void)
{
   return locmax;
}

/* Sets the locmaxmax */
void set_locmaxmax(long n)
{
   locmaxmax = n;
}

/* Generates the next random Weyl sample (31 bit integer) */
unsigned long next_rws31(void)
{
   data_pair_t *p;

   location++;
   if ((locmaxmax > 0)&&(location > locmaxmax )) return m90random31();

   if ( location > locmax ){
      p = (data_pair_t *) malloc(sizeof(data_pair_t));
      if ( p == 0 ) return CARRY_32;

      current_ptr -> next = p;
      p -> x1 = m90random31(); p -> x2 = m90random31();
      p -> a1 = m90random31(); p -> a2 = m90random31();
      p -> next = 0;
      locmax++;
   }
   current_ptr = current_ptr -> next;
   current_ptr -> x2 += current_ptr -> a2;
   current_ptr -> x1 += current_ptr -> a1;
```

```
   if ( current_ptr -> x2 & CARRY_32 ) {
      current_ptr -> x2 &= LIMIT_31;
      current_ptr -> x1 ++;
   }
   return ( current_ptr -> x1 &= LIMIT_31 );
}

/* Generates the first random Weyl sample (31 bit integer) */
unsigned long first_rws31(void)
{
   location = -1;
   current_ptr = &random_list;
   return next_rws31();
}

/* Generates the next random Weyl sample in [0,1) */
double next_rwsu(void)
{
   unsigned long rws31 = next_rws31();
   if ( rws31 == CARRY_32 ) return -1.0;
   else return (double)rws31/(double)CARRY_32;
}

/* Generates the first random Weyl sample in [0,1) */
double first_rwsu(void)
{
   location = -1;
   current_ptr = &random_list;
   return next_rwsu();
}
/*================ End of random_sampler.c ====================*/
```

5.2.2 random_sampler.h

```
/*============================================================*/
/*  file name: random_sampler.h                               */
/*  (header for random_sampler.c)                             */
/*============================================================*/

/* Constant */

#define RANDMAX   0x80000000

/* Functions for pseudo-random generation "m90random"  */

extern  void           m90setseeds(unsigned long, unsigned long,
                                   unsigned long, unsigned long,
                                   unsigned long);
extern  void           m90getseeds(unsigned long *, unsigned long *,
```

```
                                    unsigned long *, unsigned long *,
                                    unsigned long *);
extern  char           m90randombit(void);
extern  unsigned long  m90random31(void);
extern  double         m90randomu(void);

/* Functions for dynamic random Weyl sampling "DRWS"  */

extern  void           init_drws(void);
extern  void           end_drws(void);
extern  long           get_locmax(void);
extern  void           set_locmaxmax(long);
extern  unsigned long  first_rws31(void);
extern  unsigned long  next_rws31(void);
extern  double         first_rwsu(void);
extern  double         next_rwsu(void);

/*================ End of random_sampler.h ====================*/
```

5.3 ライブラリの定数と関数の仕様

このライブラリで定義されている定数と関数の仕様について述べる．

- 定数
  - RANDMAX
    値は 0x40000000= $2^{31} = 2,147,483,648$.
- 疑似乱数生成器 m90ramdom
  - void m90setseeds(unsigned long, unsigned long, ...);
    5 個の unsigned long 型整数を疑似乱数の “種” として与えることによって初期化する．この種は (22) の $\vec{x}$ に相当している．モンテカルロ計算の結果はすべて種を根源事象とする確率変数として捉えられるので，初期化は必ず行わなければならない．
  - void m90getseeds(unsigend long *, unsigned long *, ...);
    疑似乱数生成器の現在の状態を 5 個の unsigned long 型整数変数に読み込む．これら変数の値を m90setseeds の引数として渡すと，疑似乱数生成器をその状態に戻すことになり，計算の続きを行うことができる．
  - cahr m90randombit();
    ランダムな 1bit の整数 (0 または 1) を返す．これは (21) の $Y_n^{(m)}$ に相当する．
  - unsigend long m90random31();
    ランダムな 31bit の整数 ($0 \sim 2^{31} - 1 = 2,147,483,647$ = 0x3fffffff) を返す．内部では m90randombit() を 31 回呼び出している．

– `double m90randomu();`
ランダムな 31bit 精度の $[0,1]$-値実数を返す．内部では `m90random31()` の値を `RANDMAX` で割り算する．

- DRWS
以下の解説では，被積分関数である確率変数は § 4.3.2 の仮定 1 を満たしているとし，§ 4.4.3 で用いた記号も踏襲する．

  – `void init_drws();`
  DRWS の初期化．最初に必ず一度呼び出す．

  – `void end_drws();`
  DRWS が使用したメモリーを解放する．

  – `unsigned long first_rws31();`
  符号なし 31bit 整数を返す．一つのサンプルを生成するときのランダム源として，最初にこの関数を呼び出す．すなわち，Z_1 を生成するときに使う．

  – `unsigned long next_rws31();`
  符号なし 31bit 整数を返す．一つのサンプルを生成するときのランダム源として，2 番目以降にこの関数を呼び出す．すなわち，Z_l，$l \geq 2$，を生成するときに使う．l が `set_locmaxmax` で設定した上限を超えた場合，DRWS から i.i.d.-サンプリングにスイッチ，すなわち，`next_rws31()` は `m90random31()` を呼び出して，その値を返す．なお，メモリー領域を使い果たした場合，`next_rws31()` は `RANDMAX` を返してプログラマに異常事態を知らせる．

  – `double first_rwsu();`
  31bit 精度の $[0,1]$-値実数を返す．一つのサンプルを生成するときのランダム源として，最初にこの関数を呼び出す．すなわち，Z_1 を生成するときに使う．

  – `double next_rwsu();`
  31bit 精度の $[0,1]$-値実数を返す．一つのサンプルを生成するときのランダム源として，2 番目以降にこの関数を呼び出す．すなわち，Z_l，$l \geq 2$，を生成するときに使う．l が `set_locmaxmax` で設定した上限を超えた場合，DRWS から i.i.d.-サンプリングにスイッチ，すなわち，`next_rws31()` は `m90random31()` を呼び出して，その値を `RANDMAX` で割った値を返す．なお，メモリー領域を使い果たした場合，`next_rwsu()` は -1.0 を返してプログラマに異常事態を知らせる．

  – `long get_locmax();`
  現在の最大位置 (現在まで生成した W に用いた $Z_1, \ldots, Z_L$ のうち，最大の L の値) を返す．

  – `void set_locmaxmax(long);`
  最大位置 (L) の上限を設定する．-1 を代入すると上限を設定しない．デフォルトでは上限を設定しない．

5.4 サンプルプログラム

5.4.1 m90.c

このサンプルプログラムでは長さ 30 のランダムな $\{0,1\}$-列

$$110110011011010001001111110011$$

を出力する.

```
/*================================================================*/
/*  m90.c : A sample program for  Pseudo-random number generator  */
/*================================================================*/
#include <stdio.h>
#include "random_sampler.h"

int  main()
{
   int j;
   m90setseeds(0,0,0,0,0);
   for (j=1; j<=30; j++) printf("%d",m90randombit());
   printf("\n");
   return 0;
}
```

5.4.2 drws.c

このサンプルプログラムは § 4.3.2 の例 13 の確率変数 W の平均を求める. すなわち, 硬貨を投げ続けて初めて表の出た回数が 5 になるような最初の時刻の平均値を求める.

```
01:/*================================================================*/
02:/*    drws.c : A sample program for DRWS                          */
03:/*================================================================*/
04:#include <stdio.h>
05:#include "random_sampler.h"
07:
08:#define SAMPLE_SIZE  1000000
09:
10:int main()
11:{
12:   unsigned long halfmax = RANDMAX >> 1;
13:   long i;
14:   int number_of_heads, w;
15:   double sum_of_w;
16:
17:   m90setseeds(0,53,0,0,0);
18:   init_drws();
```

```
19:
20:    sum_of_w=0.0;
21:    for ( i=1; i <= SAMPLE_SIZE; i++ ){
22:       number_of_heads=0;
23:       w=1;
24:       if ( first_rws31() >= halfmax ) number_of_heads++;
25:       while ( number_of_heads < 5 ){
26:          w++;
27:          if ( next_rws31() >= halfmax ) number_of_heads++;
28:       }
29:       sum_of_w += w;
30:    }
31:    printf("Mean of hitting time = %f\n", sum_of_w/SAMPLE_SIZE);
32:    printf("locmax = %d\n", get_locmax());
33:    end_drws();
34:    return 0;
35:}
```

左端に記した行番号に沿って説明する．

05: ライブラリ random_sampler.h を読み込む．

12: 定数 RANDMAX は random_sampler.h で定義されている．同ライブラリで定義されている 31bit 整数を返す関数

m90random31(), fisrt_rws31(), next_rws31()

の値の最大値 +1 の値が RANDMAX であり，その値は 0x40000000 である．従って，この行で定義された halfmax は，その値の半分 0x20000000 である．

17: 疑似乱数生成器の初期化を行う．引数は固定せずにユーザが入力できる形にしておかなければいけないが，ここでは説明を簡単にするため，固定している．

18: DRWS を初期化する．

21: for 文の内容は SAMPLE_NUM = 1,000,000 回繰り返される．

24: first_rws31() によって最初の Z_1 に相当するランダムな整数が生成される．これが halfmax 以上になる確率は 1/2 であるから，そのとき，変数 head の値が一つ増える．

27: next_rws31() によって 2 番目以降の $Z_2, Z_3, \ldots$ に相当するランダムな整数が生成される．これが halfmax 以上になる確率はやはり 1/2 だから，そのとき，変数 head の値が一つ増える．

28: 25 行目の while 文のループの終わり．変数 head の値が 5 になったとき，このループの外に出る．その間，next_rws31() が呼ばれる回数は状況によって変化する．

29: この行の式の右辺の n の値が，目的の確率変数の値である．

31: 21 行目の for ループを抜けて，実験が終了．求める確率変数の平均は

```
n/SAMPLE_NUM
```

である．サンプルプログラムは，10.000073 という値を出力する．

32: get_locmax は，この数値実験全体を通して必要とされた $Z_1, Z_2, \ldots$ の最大個数 L を返す．サンプルプログラムは，37 という値を出力する．

33: 最後に end_drws() は DRWS のために確保されたメモリ領域を開放する．

DRWS を利用するときの要点は，各サンプルを生成するために必要な $Z_1, Z_2, \ldots$ の生成を，最初の Z_1 だけ関数 first_rws を呼び出し，(必要なら) 後の $Z_2, \ldots$ は関数 next_rws を呼び出す，ということ．それ以外は i.i.d.-サンプリングの場合とまったく同じである．だから，i.i.d.-サンプリングのプログラムがあれば，疑似乱数を呼び出すところで first_rws と next_rws の区別をしてやればよい．

注意 17 first_rws31() は next_rws31() より，first_rwsu() は next_rwsu() より，それぞれ必ず先に呼び出さなければならない．たとえば，関数

```
my_function(longint x, longint y)
```

に対して

```
my_function(first_rws31(), next_rws31());
```

という式は避けるべきである．なぜなら，my_function(x,y) において，必ずしも x の方が y より先に評価されるかどうかは不明だから．これは次のように書き直せ．

```
longint x,y;
x = first_rws31();
y = next_rws31();
my_function(x,y);
```

DRWS はきわめて高速にサンプルを生成する．実際，関数 next_rws31() において，疑似乱数生成器 m90randombit31() を呼び出すことは，全計算過程において非常に少なく，ほとんどの場合，少数の加算命令が実行されるだけなので，サンプル生成が疑似乱数生成器の速さとほぼ無関係に高速に行われるのである．

5.5 制限事項

5.5.1 m90random: サンプル数の上限

m90randombit() の生成できるランダムビットの総数は 2^{60} 個，m90random31() または m90randomu() の場合は 2^{55} 個である．

5.5.2 DRWS: サンプル数の上限

本ライブラリの生成できる DRWS のサンプルの総数は $2^{32} = 4,294,967,296$ 個である．この意味は，たとえば前節のサンプルプログラムで与えた SAMPLE_NUM の値は最大 $4,294,967,296$ まで許されるということである．

5.5.3 DRWS: メモリの使用量の管理

random_sampler は一組の (x_l, α_l) を生成するために 160 bit = 20 byte のメモリ領域を使用する．[54] たとえば前節のサンプルプログラムでは locmax = 37 を最後に出力するが，これはこのプログラムが $37 \times 20 = 740$ byte のメモリ領域を使ったことを意味する．最近の計算機は十分なメモリを有しているから，普通は問題にならない．しかし，大規模な計算，つまり W が非常に多くの Z_l たちを必要とする確率が無視できないとき，DRWS はメモリ領域を使い果たすかも知れない．そのため，現在，DRWS がどのくらいのメモリを使用中であるかを get_locmax を呼び出して常にチェックすることを心がけるのがよい．さらに実際的な解決法は，set_locmaxmax で上限を設定しておくことである．そうすれば，l がその上限を超えたとき，i.i.d.-サンプリングにスイッチする．

[54] unsigned long 型が 32 bit = 4byte の場合．

おわりに

晩年のコルモゴロフはパーキンソン氏病という難病を患いながらも,「ランダムとは何か」という問題に没頭したという ([30] p.115). § 1.3 で述べたとおり，乱数の問題はモンテカルロ法の本質的な困難である．しかしながら，モンテカルロ法を扱った論文や単行本に，コルモゴロフの乱数が正面から扱われている例を筆者は知らない．クヌースの本 [13] の pp.165-166 に紹介はあるが内容についてはまともな記述がない．数学辞典第三版には「420 乱数」という項目で内容について少し扱われてくらいである．しかし，最近出版された数学辞典第四版 [18] にはコルモゴロフの乱数の記述はなくなってしまった．もちろん，マルティン=レーフの論文 [15] は見当たらない．残念な事態である．

暗号理論における疑似乱数の考え方はインターネットサーフィン中に知った．目から鱗が落ちる思いだった．現代の計算機科学の重要な課題はすべて確率が絡んでるといって過言でない．困難な問題に対して小さな確率的リスクを覚悟した上で実行可能な労力の範囲で解決を図るのが，現代計算機科学の研究の主流だという．このような意識でモンテカルロ法を眺めるとき，それはすぐれて計算機科学の問題であるように見えてくる．

計算機科学の分野で確率論が使われているといっても，代数的あるいは組み合わせ論的な事実を確率論的に解釈して応用する，という方針が多用されている．たとえば，§ 1.5.2 定理 1 では有限体上の連立一次方程式の解の存在と一意性を確率論の独立性と解釈するという具合だ．しかし，数値解析の分野では代数的アイデアはあまりうまく働かない．実際，定理 1 の証明で示した疑似乱数生成器が実用化された，という話は聞かない．計算機科学においても理論上の道具にすぎないようだ．それはたとえば GF(2^{128}) の四則計算の実体は F_2-係数の 127 次多項式の四則計算であり，とても大変だからである．素体 F_p だと四則計算は楽だから，$2^{128} < p$ なる素数をとって F_p-値のペアごとに独立で一様分布する確率変数列を構成するのがよいかもしれない ([10]). しかし，それとてそのような素数 p を探すのは容易ではない.[55] それならいっそ，解析学的に容易に得られる § 4.2.3 定理 18 を離散化した RWS が実際の計算には数段便利である．もはや素数に縛られることはない．

この例でも分かるように，代数的に構成された工夫は数学的には単純で美しく最適解であることが多いが，一方，パラメータを少し変化させるとたちまち成り立たなくなる．p が素数であっても $p+1$ は素数でないから，同じ工夫は $p+1$ では成り立たないのである．現在流行の楕円曲線を用いた暗号理論でも，楕円曲線を定義するパラメータの特殊な選び方が重要な問題になっている．単に大きな整数をとればよい，という類のものではない．

解析学では，数の個性 (素数であることなど) に注目しないので最適解を得られないかも知れないが，パラメータの変動に伴い単調に変化する現象に注目する．たとえば § 3.2.2 定理 10 は $m \to \infty$ に伴ってワイル変換による疑似乱数生成器がある次ビット予測が指数関数的に困難になることを示している． 筆者は，計算機科学の諸問題に，解析学，とくに確率論の手法が生かされる場がたくさんあるのではないか，と期待している．

[55] *Mathematica* での計算によれば $2^{128} < p$ なる最小の素数 p は $2^{128} + 51$ である．

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

[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*, vol. **13**, (1984) 850–864. A preliminary version appears in *Proceedings of the IEEE Foundations of Comput. Sci.* (1982), 112–117.

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

[6] S. Heinrich, E. Novak, and H. Pfeiffer, How many random bits do we need for Monte Carlo integration? *Monte Carlo and quasi-Monte Carlo methods 2002*, Springer, Berlin (2004), 27–49.

[7] S. Janson, Some pairwise independent sequences for which the central limit theorem fails, *Stochastics* **23-4** (1988), 439–448.

[8] 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.

[9] A. Joffe, On a sequence of almost deterministic pairwise independent random variables, *Proc. Amer. Math. Soc.* **29** (1971), 381–382.

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

[11] 笠井琢美，計算量の理論，近代科学社，(1987).

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

[13] 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)

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

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

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

[17] D.R. Stinson, *Cryptography (Theory and practice)*, CRC Press, (1995).

[18] 数学辞典 (第 4 版)，484 (XX-9) “乱数とモンテカルロ法”，岩波書店，(2007), 1589–1591.

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

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

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

[22] 杉田洋，複雑な関数の数値積分とランダムサンプリング,「数学」岩波書店 **56-1**, (2004) 1–17.

[23] H. Sugita, An analytic approach to secure pseudo-random generation, *Proceedings of 2003 Ritsumeikan Symposium on Stochastic Processes and its Applications to Mathematical Finance*, World Scientific, (2004) 355–368.

[24] H. Sugita, Security of Pseudo-random Generator and Monte-Carlo Method, *Monte Carlo Methods and Appl.*, **10-3**, VSP(2004), 609–615.

[25] 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://homepage.mac.com/hiroshi_sugita/mathematics.html`.

[26] 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.

[27] 高橋正子，計算論 (計算可能性とラムダ計算)，近代科学社，(1991).

[28] 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.

[29] 高信敏，private communication, (1998).

[30] 高橋陽一郎+志賀浩二，確率論をめぐって，日本評論社，(1992).

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

[32] 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).

[33] K. Yasutomi, A limit theorem for sequences generated by Weyl transformation: Disappearance of dependence, *Probab. Theory Relat. Fields* **124** (2002), no. 2, 178–188.

[34] K. Yasutomi, A direct proof of dependence vanishing theorem for sequences generated by Weyl transformation, *J. Math. Kyoto Univ.* **43** (2003), no. 3, 599–607.

[35] K. Yasutomi, A dependence vanishing theorem for sequences generated by Weyl transformation, *J. Math. Kyoto Univ.* **44** (2004), no. 2, 365–380.

* 文献は網羅的なものではない．とくに計算機科学関係の文献はもっと充実させる必要がある．

索引

sugita@math.sci.osaka-u.ac.jp
http://homepage.mac.com/hiroshi_sugita/mathematics.html