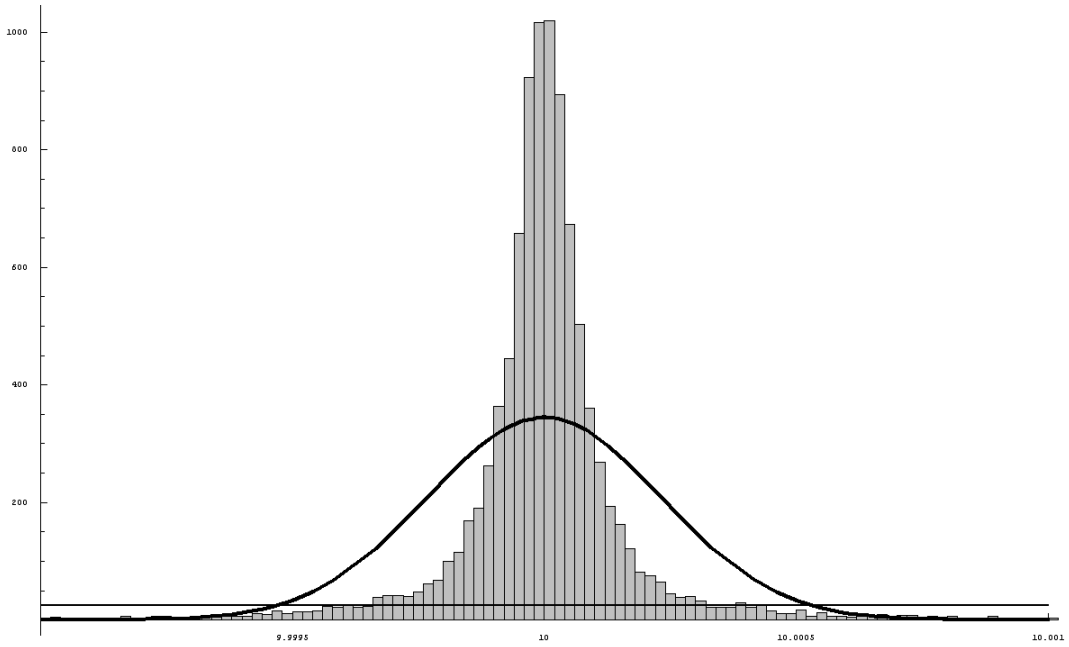
モンテカルロ法, 乱数, および疑似乱数†

杉 田 洋 / 著



† Ver.20160624 / MSJ Memoirs vol.25 (2011) の著者による改訂翻訳.

はじめに

モンテカルロ法とは，確率変数のサンプリングをコンピュータを用いて行うことによって数学的問題を (主として数理統計学における意味で) 数値的に解く手法をいう．1940 年代第二次世界大戦中に，フォン・ノイマン (von Neumann) やウラム (Ulam) らがロスアラモス研究所 (米国) の当時発明されたばかりのコンピュータによって核分裂物質中の中性子の拡散の様子をシミュレートした[注1] のが，その始まりであるらしい ([56] p.3)．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夥しい成果を挙げてきた．そしてその発展は今後も続くことであろう．

本書では，しかし，そうした華々しいモンテカルロ法の応用技術についてではなく，主として数学的基礎理論，とくに次に述べるようなサンプリングの正当性について論じる．すなわち,「コンピュータでは乱数を生成することはできない」という原理的困難のために，実際のモンテカルロ法ではコンピュータで生成する疑似乱数[注2]を乱数の代わりに用いて確率変数のサンプリングを行っているが，そこに研究者たちは重大な疑念を抱いてきた;

> 疑似乱数を用いたモンテカルロ法は決して数学的正当性を持ち得ないのではないか.

じつは，この疑念はモンテカルロ法，乱数および疑似乱数[注3]の直感的な解釈による先入観から生じた誤解に過ぎない．すなわち，乱数および疑似乱数の概念を正しく理解し，モンテカルロ法そのものの数学的定式化を適切に行うことによって，この疑念が解消される可能性は残されているし，また実際に解消できる場合もあることが示せる．本書ではそのことを詳細に解説する．同時にそのようなモンテカルロ法の数学的定式化は最も信頼できるサンプリング法の開発に結び付くことを述べる．

確率変数のサンプリングをコンピュータでどのように行うべきか，という問題はモンテカルロ法誕生以来，重大な問題であり続けてきた．1960 年代にコルモゴロフ (Kolmogorov) らは今日 “コルモゴロフ複雑度” と呼ばれる関数を用いて乱数の概念を定義した ([5, 22, 23, 24, 31])．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，暗号理論の文脈でブラム (Blum) らが疑似乱数という概念を定式化した ([2, 3, 58])．それ

[注1] 原子爆弾製造の目的のために.

[注2] 擬似乱数と書くこともある.

[注3] モンテカルロ法の文献の多くは乱数と疑似乱数を区別しないで扱っている．たとえば [21] の副題は「乱数」だが，実際の主題は疑似乱数である．本書では両者は厳格に区別される.

は，より現実的な計算可能性を論じる計算量の理論を基礎としている．だが，残念なことに，コルモゴロフらの乱数の概念にも，ブラムらの疑似乱数の概念にも，モンテカルロ法の研究者たちは注目して来なかった．前者は難解な上に現実的な解決をもたらさないこと，後者は暗号理論における疑似乱数は用途が異なるためモンテカルロ法では役立たないと考えられたこと，がその理由であった．

しかし，これはモンテカルロ法それ自体に対する従来の捉え方が不適切だったことからコルモゴロフらの乱数の概念とブラムらの疑似乱数の概念の価値の正しい評価ができなかった，ということに過ぎない．本書では，むしろコルモゴロフの乱数の概念とブラムらの疑似乱数の概念を拠り所として，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．その結果，モンテカルロ法の全貌が明確になり，未だ完全な解決ではないものの，我々の定式化の下で次のことが分かる．

> 疑似乱数を用いてもモンテカルロ法は数学的に正当化され得る可能性がある．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，それが実際に可能で，乱数と同等 (じつはそれ以上) 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，順を追って本書の内容を概観しよう．

本書では乱数にしても疑似乱数にしてももっぱら硬貨投げの確率過程をモデルとする $\{0,1\}$-列のみを扱う．第 1 章の前半では，理論上は硬貨投げの確率過程から任意の分布の確率変数 (もっと一般に任意の独立確率変数列) が構成できることを述べる．後半では，どのような確率変数がコンピュータで実現することが可能かについて考察する．

第 2 章では，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を，乱数，および疑似乱数のおおまかな定義とともに規定し，それら相互のつながりを明らかにする．諸概念の捉え方が従来のモンテカルロ法の説明 (たとえば [16, 21, 38, 41, 56]) とはまったく異なるので注意深く読んで欲しい．第 2 章の内容は本書全編の考え方の主柱をなすものであり，この章を理解するだけでも本書は十分有益だと思う．

第 3 章では，はじめに計算理論の基礎的概念である部分帰納的関数を紹介し，その基本的性質を述べる (cf. [20, 53])．次にコルモゴロフ複雑度と呼ばれる関数を導入し，それを用いて乱数を定義する (cf. [5, 22, 23, 24, 25, 28, 31, 64])．コルモゴロフ複雑度は任意の有限 $\{0,1\}$-列に対して定義はできるものの実際には計算不可能な関数であり，そのことが計算理論の重要な帰結である “停止問題の計算不可能性” と関連があることを明らかにする．第 3 章の後半では乱数とランダム性の検定に関する重要なマルティン=レーフの定理 ([31]) を証明する．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，彼はこの章で紹介するとおり，その困難な問いに見事な解答を与えているのである．最後の § 3.6 では，乱数の概念を認識することがモンテカルロ法のみならず確率論そのものの深い理解をもたらすことを述べる．

第 4 章の前半 § 4.1 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 (cf. [29, 40])．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望ましいことを明らかにする．また，この性質を持つ疑似乱数生成器の存在が未解決の **P ≠ NP** 予想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．第 4 章の後半 § 4.2 では，筆者が見出したある疑似乱数生成器 ([43]) について詳しく述べる．それは，確率論的に “計算量的に安全な疑似乱数生成器” の実現に迫るものである．§ 4.3 では § 4.2 で示した諸定理の証明を紙数を顧みずできるだけ丁寧に与えた．

第 5 章では，§ 5.2 で，モンテカルロ積分の場合に “安全な疑似乱数生成器” の働きを果たす “ランダム-ワイル-サンプリング (RWS)” について詳しく論じ，とくにその標本平均に関する中心極限定理のスケール極限が 0 に確率収束することを示す．このことは，RWS が i.i.d.-サンプリングより誤差が小さくなる確率が高いことを示していて，従ってより数値積分に有利であることが分かる (cf. [46])．さらに § 5.4 で紹介される動的ランダム-ワイル-サンプリング (DRWS，[45]) は，RWS を任意の模倣可能な確率変数の数値積分の場合に拡張したものであり，高速で最も信頼性の高いサンプリング法であることが実例によって明らかになる．

第 6 章では，§ 4.2 の疑似乱数生成器と，§ 2.5.2，§ 5.2.2 および § 5.4 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 (cf. [51])．とくに § 6.2 の汎用 C 言語ライブラリは様々な応用にそのままの形で利用できる．

以上，本書には筆者が重要と考えるほとんどすべての内容を盛り込んだ．しかし研究は継続中である．まだ解決を見ない問題のうち，ぜひとも解決したいいくつかの重要問題について本文中で触れた．いずれも容易でない問題だが読者にも挑戦して頂ければと願う．

本書の原型は神戸大学での集中講義 (2000 年 5 月) の講義ノートであった．当時，福山克司氏の勧めで神戸大学理学部数学教室発行のレクチャーノートとして出版することになっていた．しかし，福山氏には多数の助言まで頂きながら，筆者自身納得のいく原稿が書けず，結局，出版には至らなかった．大変迷惑を掛けたのであるが，お陰で思い存分，モンテカルロ法の問題に集中して悩むことができたし，筆者の理解も深まり本書も内容の充実したものとなった．ここに氏に改めてお詫びと感謝の意を表したいと思う．

本書の原稿を使って大阪大学大学院理学研究科大学院の講義やセミナーを行った．幾人かの熱心な学生諸君は誤りを見つけた．半田賢司氏，竹居正登氏には誤植を詳細に報告して頂いた．藤原彰夫氏には乱数に関して助言を頂いた．また英語版 [49] に関してはさらに M.Yor，H.Niederreiter 両氏にもコメントを頂いた．以上の方々にぜひ感謝したい．

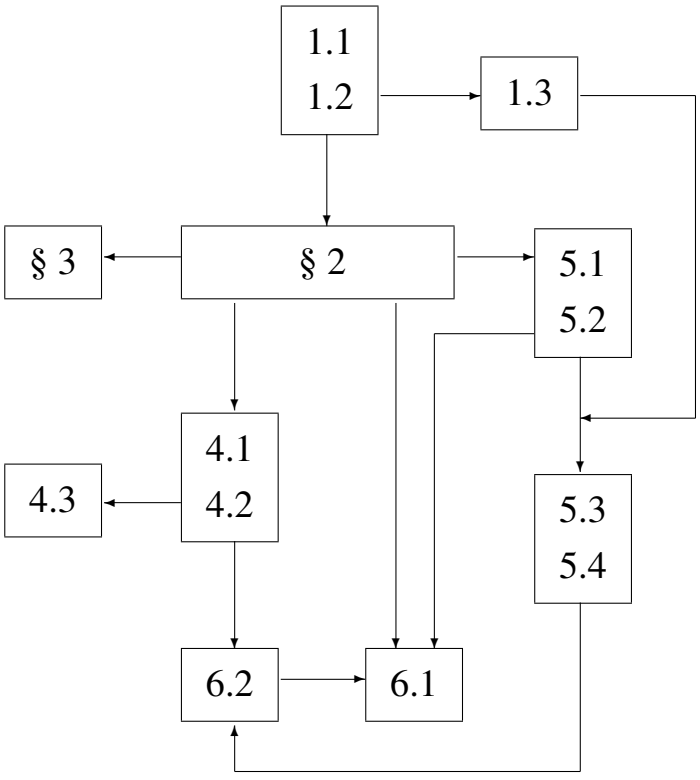
最後に，本書の内容に関しては共同研究者の高信敏氏に負うところは特別大きいことを述べなければならない．氏とともに長年に亘り研究できたことを大きな喜びに思う．

2016 年 6 月

杉田 洋
大阪大学大学院理学研究科

目 次

章・節の間の依存関係



記号

最初に全編を通じて用いられる記号をいくつか列挙しておく.

$A := B$		B のことを A と定義する ($B =: A$ も同じ)
$\forall \cdots,$		任意の $\cdots$ について,
$\exists \cdots,$		ある $\cdots$ が存在して,
$P \Longrightarrow Q$		P ならば Q (論理的包含)
$P \Longleftrightarrow Q$		P は Q と同値
s.t.		such that, 以下を満たすような
$\square$		証明終わり
$\mathbb{N}$	:=	$\{0, 1, 2, \ldots\}$, 0 と自然数全体
$\mathbb{N}^+$	:=	$\{1, 2, \ldots\}$, 自然数全体
$\mathbb{Z}$	:=	$\{\ldots, -2, -1, 0, 1, 2, \ldots\}$, 整数全体
$\mathbb{R}$	:=	実数全体
$\mathbb{C}$	:=	複素数全体
$\lfloor t \rfloor$	:=	実数 t を超えない最大の整数 (切り捨てて整数に丸める)
$\lceil t \rceil$	:=	実数 t を下まわらない最小の整数 (切り上げて整数に丸める)
$\mathbf{1}_B(x)$	:=	集合 B の定義関数 $= \begin{cases} 1 & (x \in B) \\ 0 & (x \notin B) \end{cases}$
$\#B$	:=	集合 B の元の個数
2^B	:=	集合 B の部分集合の全体
B^c	:=	集合 B の補集合
$A \setminus B$	:=	$A \cap B^c$
$\emptyset$	:=	空集合
Pr	:=	確率 (とくに確率空間を明示しない場合に用いる)
$\mathbf{E}[X]$	:=	確率変数 X の平均 (期待値)
$\mathbf{E}[X; B]$	:=	$\mathbf{E}[X\mathbf{1}_B]$
$\mathbf{V}[X]$	:=	確率変数 X の分散
$\mathcal{N}(m, \sigma^2)$	:=	平均 m, 分散 σ^2 の正規分布
$O(f(n))$	:=	ランダウ (Landau) の記号.
		$g(n) = O(f(n)) \Longleftrightarrow \exists c > 0$ s.t. $\lvert g(n) \rvert \le c f(n)$
$\mathfrak{0}$	:=	ゼロ. O(オー) と混乱を避けるために使われる字体
$a \bmod N$	:=	a を N で割った余り, $N = 1$ のときは a の小数部分

第1章　硬貨投げの確率過程

本書では乱数にしても疑似乱数にしても，すべて**硬貨投げの確率過程**[注1]をモデルとする $\{0,1\}$-列を対象としている．このことは，応用上，非常に限られた印象を与えるかもしれないが，じつはそうではなく，硬貨投げの確率過程から事実上すべての確率変数や確率過程を構成することができる．

1.1　ボレルによる硬貨投げの確率過程のモデル

硬貨の表裏をそれぞれ 1 と 0 で表せば m 回の硬貨投げは確率空間 $(\{0,1\}^m, 2^{\{0,1\}^m}, P_m)$ によって記述される．ここに P_m は $\{0,1\}^m$ 上の一様確率測度である;

$$P_m(B) := \frac{\#B}{2^m}, \quad B \subset \{0,1\}^m \ (B \in 2^{\{0,1\}^m}).$$

しかし，これでは m が変わるたびに異なる確率空間を設定する必要があり，煩わしいばかりでなく，極限定理を述べる際にも不都合である．一挙に無限回の硬貨投げ (硬貨投げの確率過程) を適当な確率空間上で作っておくのがよい．ここではボレル (E.Borel) に従って，ルベーグ確率空間上でそれを構成する．

定義 1.1

1. 区間 $[0,1)$ 上の加法として $(x+y) \bmod 1$ を考えた群を $\mathbb{T}^1$ と表し，**1 次元トーラス** (1-dimensional torus) と呼ぶ．$\mathcal{B}$ を $\mathbb{T}^1 = [0,1)$ に含まれるボレル可測集合全体からなる完全加法族 (σ-加法族)，$\mathbb{P}$ をルベーグ測度 (Lebesgue measure) とする．三つ組 $(\mathbb{T}^1, \mathcal{B}, \mathbb{P})$ を**ルベーグ確率空間** (Lebesgue probability space) とよぶ.[注2] $(\mathbb{T}^1, \mathcal{B}, \mathbb{P})$ 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．

2. $d_i(x)$ 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;

$$x = \sum_{i=1}^{\infty} d_i(x) 2^{-i}. \tag{1.1}$$

ただし，

$$d_1(x) := \mathbf{1}_{[1/2,1)}(x), \quad d_i(x) := d_1(2^{i-1}x), \quad i \in \mathbb{N}^+, \quad x \in \mathbb{T}^1.$$

[注1] 本書では，いわゆる “公平な” 硬貨投げを単に硬貨投げと呼ぶ.

[注2] $\mathcal{B}$ を $\mathbb{P}$ で完備化した完全加法族 (ルベーグ可測集合族) を考えることも多いが，数値計算の目的のためには $\mathcal{B}$ で十分なのでここではこれを採用している.

3. $m \in \mathbb{N}^+$ に対して

$$D_m := \{i2^{-m} \mid i = 0, \ldots, 2^m - 1\} \subset \mathbb{T}^1 \tag{1.2}$$

とする．集合族 $\mathcal{I}_m := \{[a, a + 2^{-m}) \mid a \in D_m\}$ 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 $\mathcal{I}_m$ の有限個の元の和集合である．D_m 上の一様確率測度を $P_{(m)}$ で表す.

4. $m \in \mathbb{N}^+$，$x \in \mathbb{T}^1$ に対して

$$\lfloor x \rfloor_m := \lfloor 2^m x \rfloor \cdot 2^{-m} \in D_m, \tag{1.3}$$

$$\lceil x \rceil_m := \lceil 2^m x \rceil \cdot 2^{-m} \in D_m, \tag{1.4}$$

さらに $\lfloor x \rfloor_\infty := x$ とする．

定理 1.2 ([4]) ルベーグ確率空間上の確率変数列 $\{d_i\}_{i=1}^\infty$ は硬貨投げの確率過程である．

証明. 任意の $n \in \mathbb{N}^+$，任意の $\epsilon_1, \ldots \epsilon_n \in \{0, 1\}$ に対して，$t := \sum_{i=1}^n 2^{-i}\epsilon_i$ と置けば

$$\left\{ x \in \mathbb{T}^1 \,\middle|\, d_i(x) = \epsilon_i,\ i = 1, \ldots, n \right\} = [t, t + 2^{-n})$$

であるから $\mathbb{P}(\, d_i = \epsilon_i,\ i = 1, \ldots, n\,) = \mathbb{P}(\,[t, t + 2^{-n})\,) = 2^{-n}$ である． □

2 進展開写像 $D_m \ni x \mapsto (d_1(x), \ldots, d_m(x)) \in \{0, 1\}^m$ は全単射であり，また，写像 $\lfloor \bullet \rfloor_m : \mathbb{T}^1 \to D_m$ (または $\lceil \bullet \rceil_m : \mathbb{T}^1 \to D_m$) は $\mathcal{B}_m$ と 2^{D_m} の間の全単射を誘導する．これらによって確率空間として次の同型がある．

$$(\{0, 1\}^m, 2^{\{0,1\}^m}, P_m) \cong (D_m, 2^{D_m}, P_{(m)}) \cong (\mathbb{T}^1, \mathcal{B}_m, \mathbb{P}).$$

1.2　硬貨投げの確率過程による確率変数の構成

定理 1.3　S を確率空間 $(\Omega, \mathcal{F}, P)$ 上で定義された任意の実数値確率変数とする．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 S と同分布なものが存在する．

証明. S の分布関数 $F(S; t) := P(S \le t)$，$t \in \mathbb{R}$，を用いて

$$f(x) := \sup\{u \in \mathbb{R} \mid F(S; u) < x\}, \quad 0 < x < 1, \tag{1.5}$$

とすれば，これがルベーグ確率空間上の確率変数として S と同分布になる．このことを示すためには

$$\{\, 0 < x < 1 \mid f(x) \le t\,\} = \{\, 0 < x < 1 \mid x \le F(S; t)\,\}, \quad t \in \mathbb{R}, \tag{1.6}$$

を示せばよい．実際，両辺の集合のルベーグ測度を見れば $\mathbb{P}(f \le t) = F(S; t)$ となるからである．そこで，(1.6) を示そう．まず，$x \le F(S; t)$ ならば $t \notin \{u \in \mathbb{R} \mid F(S; u) < x\}$

であるから，$f(x) \le t$ である．一方，$x > F(S;t)$ ならば $F(S;\bullet)$ は右連続であるから，ある $\varepsilon > 0$ が存在して $x > F(S;t+\varepsilon)$，このとき，$f(x) \ge t+\varepsilon > t$ である．以上から (1.6) が分かる． □

定理 1.3 の条件を満たす f はたくさん存在し，(1.5) 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

$$f(x) \;=\; f\left(\sum_{i=1}^{\infty} d_i(x)2^{-i}\right), \quad x \in \mathbb{T}^1,$$

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[注3]と考えることができるから，定理 1.3 によって，任意の確率変数 S に対して硬貨投げの確率過程の汎関数であるような S と同分布の確率変数が存在することが分かる．

定理 1.4 ([39])

$$\begin{aligned}
Z_1 &= \frac{1}{2}d_1 + \frac{1}{2^2}d_3 + \frac{1}{2^3}d_6 + \frac{1}{2^4}d_{10} + \cdots \\
Z_2 &= \frac{1}{2}d_2 + \frac{1}{2^2}d_5 + \frac{1}{2^3}d_9 + \cdots \\
Z_3 &= \frac{1}{2}d_4 + \frac{1}{2^2}d_8 + \cdots \\
Z_4 &= \frac{1}{2}d_7 + \cdots \\
&\vdots
\end{aligned}$$

とすれば，$\{Z_n\}_{n=1}^{\infty}$ は $\mathbb{T}^1$ で一様分布する i.i.d. 確率変数列[注4]である．

この定理の証明は易しいの省略する．

定理 1.3 と定理 1.4 より，硬貨投げの確率過程から任意の分布の独立確率変数列を構成することが可能である．たとえば，定理 1.4 の $\{Z_n\}_{n=1}^{\infty}$ から定理 1.3 を用いて標準正規分布 $N(0,1)$ に従う i.i.d. 確率変数列 $\{\xi_n\}_{n=0}^{\infty}$ を構成できる．さらにこれを用いて

$$B_t \;:=\; \frac{t}{\sqrt{\pi}}\xi_0 + \sqrt{\frac{2}{\pi}}\sum_{n=1}^{\infty}\frac{\sin nt}{n}\xi_n, \quad 0 \le t \le \pi,$$

とおけば，$\{B_t\}_{0\le t\le\pi}$ はブラウン運動である ([13] p.93 例 4)．再び定理 1.4 と同様の工夫をすれば，独立な可算個のブラウン運動さえ硬貨投げの確率過程 $\{d_i\}_{i=1}^{\infty}$ より構成できることも分かる．じつは，たとえば非可算個の i.i.d. 確率変数族を扱うという特殊な場合を除けば，事実上すべての確率論的対象は硬貨投げの確率過程 $\{d_i\}_{i=1}^{\infty}$ より構成できることが知られている．詳しくは [35] Chapter 1 を見よ．

[注3] 無限個の変数を持つ関数を汎関数という．ここでは $f(x)$ 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．

[注4] i.i.d. は独立同分布 (*i*ndependently *i*dentically *d*istributed) の意．

1.3　模倣可能な確率変数

定理 1.3 によって，任意の確率変数 S と同じ分布を持つ確率変数 f を硬貨投げの確率過程の汎関数として構成できることを見た．しかし，このことをもって任意の確率変数 S がいつもモンテカルロ法において実現可能であるわけではない．普通は S の分布関数が明示的に求まることは少なく，従って (1.5) 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のサンプルは無限 $\{0,1\}$-列が一挙にではなく第 1 項から順に供給されるので，$f(x)$ 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．

硬貨投げの確率過程の汎関数 f がモンテカルロ法で実現できるとき模倣可能 (simulatable) であるという．この節では f が模倣可能であるための条件について考えていく．[注5]

1.3.1　停止時刻に関して可測な確率変数

ルベーグ確率空間上で定義された確率変数で，ある $m \in \mathbb{N}^+$ に対して $\mathcal{B}_m$-可測であるものは明らかに模倣可能である．[注6] 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．

例 1.5 (到達時刻) ルベーグ確率空間上で定義された確率変数

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

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 は非有界であり，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．確率 1 で $\sigma(x)$ の計算は有限時間で終わる．

例 1.5 を含むような，モンテカルロ法で実際に扱い得る一般的なクラスを考えよう．

定義 1.6　関数 $\tau : \mathbb{T}^1 \to \mathbb{N}^+ \cup \{\infty\}$ は

$$\forall m \in \mathbb{N}^+, \quad \{\tau \le m\} := \{x \in \mathbb{T}^1 \,|\, \tau(x) \le m\} \in \mathcal{B}_m,$$

を満たすとき，$\{\mathcal{B}_m\}_m$**-停止時刻** (cf. [1]) という．$\{\mathcal{B}_m\}_m$-停止時刻 τ に対して，$\mathcal{B}$ の部分完全加法族 $\mathcal{B}_\tau$ を

$$\mathcal{B}_\tau := \{A \in \mathcal{B} \,|\, \forall m \in \mathbb{N}^+,\, A \cap \{\tau \le m\} \in \mathcal{B}_m\}$$

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．

[注5] § 1.3 の内容は § 5.3 まで必要としないので，初回は読み飛ばしてよい．

[注6] 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，正確には「あるチューリング機械 (Turing machine，無限のメモリを持つ仮想的なコンピュータと思えばよい．cf. [6, 20, 53]) で計算できる」という意味である．

定数時刻 $\tau(x) \equiv m \in \mathbb{N}^+$ は停止時刻であり，$\mathcal{B}_\tau = \mathcal{B}_m$ である．

関数 $f : \mathbb{T}^1 \to \mathbb{R} \cup \{\pm\infty\}$ が $\mathcal{B}_m$-可測であるための必要十分条件は $f(x) = f(\lfloor x \rfloor_m)$, $x \in \mathbb{T}^1$，が成り立つことである．このことの一般化として次が成り立つ．

補題 1.7 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

$$f(x) \;=\; f\left(\lfloor x \rfloor_{\tau(x)}\right), \qquad x \in \mathbb{T}^1, \tag{1.7}$$

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，$\{\tau = m \text{ かつ } f \le t\} \in \mathcal{B}_m$ である．このことから $f(x)\mathbf{1}_{\{\tau=m\}}(x)$ は $\mathcal{B}_m$-可測であることが分かる．従って

$$\begin{aligned}
f(x) &= \sum_{m\in\mathbb{N}^+} f(x)\mathbf{1}_{\{\tau=m\}}(x) + f(x)\mathbf{1}_{\{\tau=\infty\}}(x) \\
&= \sum_{m\in\mathbb{N}^+} f\left(\lfloor x \rfloor_m\right)\mathbf{1}_{\{\tau=m\}}(x) + f\left(\lfloor x \rfloor_\infty\right)\mathbf{1}_{\{\tau=\infty\}}(x) \\
&= \sum_{m\in\mathbb{N}^+} f\left(\lfloor x \rfloor_{\tau(x)}\right)\mathbf{1}_{\{\tau=m\}}(x) + f\left(\lfloor x \rfloor_{\tau(x)}\right)\mathbf{1}_{\{\tau=\infty\}}(x) \\
&= f\left(\lfloor x \rfloor_{\tau(x)}\right)\sum_{m\in\mathbb{N}^+} \mathbf{1}_{\{\tau=m\}}(x) + f\left(\lfloor x \rfloor_{\tau(x)}\right)\mathbf{1}_{\{\tau=\infty\}}(x) \;=\; f\left(\lfloor x \rfloor_{\tau(x)}\right).
\end{aligned}$$

十分性 : もし f が (1.7) を満たせば，各 $m \in \mathbb{N}^+$，$t \in \mathbb{R}$ に対して

$$\{f \le t\} \cap \{\tau \le m\} \;=\; \{f\left(\lfloor \bullet \rfloor_{\tau(\bullet)}\right) \le t\} \cap \{\tau \le m\} \;=\; \{f\left(\lfloor \bullet \rfloor_m\right) \le t\} \cap \{\tau \le m\} \;\in\; \mathcal{B}_m$$

だから，f は τ-可側である． □

例 1.5 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．

確率 1 で有限なある $\{\mathcal{B}_m\}_m$-停止時刻 τ に対して τ-可測であるような f は模倣可能である．実際，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 $\tau(t) = m$ 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 $f(x)$ を出力し終了する．

逆に $f(x)$ が模倣可能ならば，それは確率 1 で有限回の硬貨投げから計算されなければならない．すなわちルベーグ測度に関しほとんどすべての $x \in \mathbb{T}^1$ に対して，ある $m \in \mathbb{N}^+$ が存在して $f(x) = f(\lfloor x \rfloor_m)$ でなければならない．ここで m は x に依存してよいので，これを $\tau(x)$ と書けば，確率変数 $\tau(x) \in \mathbb{N}^+ \cup \{\infty\}$ は $\mathbb{P}(\tau < \infty) = 1$ を満たす．しかし τ が $\{\mathcal{B}_m\}_m$-停止時刻でないときは，$f(x)$ を計算するために $d_i(x)$ たちを永遠に供給し続けなければならない事態が生じるから，τ が $\{\mathcal{B}_m\}_m$-停止時刻であることは f をモンテカルロ法で実現するためには不可欠であることが分かる．

だから，f が模倣可能であるとは「ある $\{\mathcal{B}_m\}_m$-停止時刻 τ が存在して f が τ-可測であること」と定義するのがよいだろう．

例 1.8 (最終脱出時刻)

$$\tau'(x) := \sup\left(\left\{ n \in \mathbb{N}^+ \;\middle|\; \frac{d_1(x) + d_2(x) + \cdots + d_n(x)}{n} - \frac{1}{3} < 0 \right\} \cup \{1\}\right), \quad x \in \mathbb{T}^1,$$

は，大数の強法則により確率 1 で有限だが $\{\mathcal{B}_m\}_m$-停止時刻ではない．実際，$\tau'(x)$ の値は有限個の $d_i(x)$ たちの値からは決定できない．$\tau'(x)$ は模倣可能でない．

1.3.2　$\mathbb{T}^1$-値一様分布独立確率変数列がランダム源の場合

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 $\{Z_l\}_{l=1}^{\infty}$ 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 1.9 [注7] f は $\{Z_l\}_{l=1}^{\infty}$ の汎関数であって，f を計算するためには確率 1 で，有限個の $Z_1, \ldots, Z_T$ しか必要でない，と仮定する．ここで，Z_l たちの個数 T は確率変数であるが，各 $l \in \mathbb{N}^+$ に対して，事象 $\{T \le l\}$ が起こるかどうかは $Z_1, \ldots, Z_l$ の値から判断され，$Z_{l'}$, $l' \ge l+1$，に関して何の知識も必要ない．

例 1.10　f 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 T が定数の場合は，f は仮定 1.9 を満たす．

次の例のように，T 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 1.11 (フォン・ノイマンの棄却法 [33]) $\mathbb{T}^1$-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 f 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 $[a,b] \times [0,M]$ で一様分布する確率変数とするとき，

$$\Pr(\,\xi \in [c,d] \,|\, p(\xi) \ge \eta\,) \;=\; \int_c^d p(x)dx, \quad a \le c < d \le b,$$

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．

[注7] 正確な定式化は § 5.4.5 で与えられる．

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，$f := \xi$ 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 f の分布は望みのものになっている．このとき f は仮定 1.9 を満たしている．

例 1.12 (例 1.5 のいい換え) $\{Y_n\}_{n=1}^{\infty}$ を

$$Y_n := \begin{cases} 0 & (Z_n \in [0, 1/2)) \\ 1 & (Z_n \in [1/2, 1)) \end{cases} \qquad n = 1, 2, \ldots,$$

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

$$f := \inf\{\, n \in \mathbb{N}^+ \mid Y_1 + Y_2 + \cdots + Y_n = 5 \,\}$$

とすれば f は仮定 1.9 を満たす．f 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．

実際には実数計算が有限精度，たとえば 2^{-K}，で行われるので，$\{Z_l\}_{l=1}^{\infty}$ の代わりに D_K-値一様分布に従う i.i.d. $\{Z_l^{(K)}\}_{l=1}^{\infty}$ を $(\mathbb{T}^1, \mathcal{B}, \mathbb{P})$ 上で

$$Z_n^{(K)} := \sum_{i=1}^{K} 2^{-i} d_{(n-1)K+i}, \qquad n \in \mathbb{N}^+, \tag{1.8}$$

すなわち

$$\begin{aligned} Z_1^{(K)} &= \frac{1}{2}d_1 + \frac{1}{2^2}d_2 + \cdots + \frac{1}{2^K}d_K \\ Z_2^{(K)} &= \frac{1}{2}d_{K+1} + \frac{1}{2^2}d_{K+2} + \cdots + \frac{1}{2^K}d_{2K} \\ Z_3^{(K)} &= \frac{1}{2}d_{2K+1} + \frac{1}{2^2}d_{2K+2} + \cdots + \frac{1}{2^K}d_{3K} \\ &\vdots \end{aligned}$$

のように実現して用いる．そして，f が仮定 1.9 を満たすとき，それを $\{Z_l^{(K)}\}_{l=1}^{\infty}$ の汎関数と見なして

$$\tau(x) := \inf\{\, lK \in K\,\mathbb{N}^+ \mid f(x) \text{ は } Z_1^{(K)}(x), \ldots, Z_l^{(K)}(x) \text{ から計算される} \,\}$$

とおけば，τ は $\{\mathcal{B}_m\}_m$-停止時刻であり，f は τ-可測となる．

第2章　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困難である乱数の問題，およびその解決を目的とする疑似乱数生成器について基本事項を述べる．この章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章で紹介しそれぞれ掘り下げて解説する．

2.1　概要

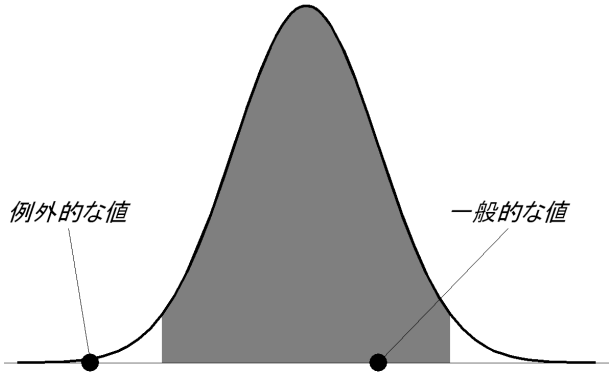
最初に第 2 章全体の概要を述べる．

モンテカルロ法 (Monte-Carlo method) とは，確率変数のサンプリングをコンピュータを用いて行うことによって数学的問題を (主として数理統計学における意味で) 数値的に解く手法をいう．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 $(\Omega, \mathcal{F}, P)$ — 第 2 章では簡単のため，有限回の硬貨投げの確率空間

$$(\{0,1\}^L,\ 2^{\{0,1\}^L},\ P_L), \quad L \in \mathbb{N}^+,$$

に限る — 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** (sampling，標本抽出) と呼ぶ．

図 2.1: S の分布と一般的な値 (概念図)



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 (§ 2.2.1)．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**自分の意思で** ω を選ぶ．彼女のリスクは

$$A := \left\{ \omega \in \{0,1\}^L \,\middle|\, S(\omega) \text{ は } S \text{ の例外的な値} \right\}$$

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．[注1] 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，$L \gg 1$ (§ 2.2.2 の例題では $L = 10^8$) 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，$\omega \in \{0,1\}^L$ を指定するには，$L = 10^8$ ともなればそれは約 12MB のデータにもなるので,[注2]情報量の大きさからいってコンピュータを用いるよりない．しかしこれだけの情報量のデータ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，従って特殊なものに限られてしまう (§ 2.3)．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，$\{0,1\}^L$ の大部分を占める「アリスの意志では選ぶことのできないような ω」— そのような ω を**乱数**と呼ぶ — によるサンプリングが必要であると考えるのは自然である．[注3]

しかし一方，現実の問題で扱う 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 — つまり乱数でなくても — 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 $\{0,1\}$-列に引き伸ば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」という別の新しい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 g に依存するが，もし $P_n(g(\omega') \in A) \ll 1$ であるような g を見つけることができたなら，膨大な時間と労力をかけずに実際に高い確率でアリスは S の一般的な値を手に入れることができる．

[注1] $a \ll b$ 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．

[注2] 400 字詰原稿用紙で約 15,000 枚分の情報量．

[注3] そのため，物理乱数を用いる立場もあるが，本書では以下に述べるように疑似乱数生成器による解決について論じる．

この場合，長大な乱数は必要でない．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 (§ 2.4.2)．モンテカルロ法におけるサンプリングの問題はこのような安全な疑似乱数生成器を見出すことで解決される．

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 (§ 2.4.3，§ 4.1)．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 i.i.d. 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る．そして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 (§ 2.5.2)．

2.2 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解を推定することが実際的である．モンテカルロ法はまさにそのような場合の一つである．

2.2.1 プレーヤーの目的

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— 典型的な，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 (コンピュータを必要としない) モンテカルロ法の例を示す．

例 2.1 r を未知の整数とする．100 枚のカードがあり，そのうち 99 枚には r が，残りの 1 枚には $r+1$ が，書いてある．アリスがその中から 1 枚だけを選び，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 1/100 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，$r+1$ 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 1/100 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 2.1 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.[注4]

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 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 2.2 $\Pr(X < c) = 1/10000$ 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 $\{X_k\}_{k=1}^{40000}$ を X の独立なコピーとし，$S := \min_{1\le k\le 40000} X_k$ とする．このとき

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

となる.[注5] だから，S は高い確率 0.98 で c 未満の値を取る.

2.2.2 一つの例題

例 2.1 を実行するのには，100 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．第 2 章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 100 回行うとき表が続けて 6 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 100 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 6 回以上出る」という事象が起った回数を 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．より具体的には

例 2.3 $N := 10^6$ とする．$S_{10^6}/10^6$ の平均と分散は

$$\mathbf{E}\left[\frac{S_{10^6}}{10^6}\right] = p, \quad \mathbf{V}\left[\frac{S_{10^6}}{10^6}\right] = \frac{p(1-p)}{10^6} \le \frac{1}{4\cdot 10^6}$$

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

$$\Pr\left(\left|\frac{S_{10^6}}{10^6} - p\right| \ge \frac{1}{200}\right) \le \frac{1}{4\cdot 10^6}\cdot 200^2 = \frac{1}{100}, \tag{2.1}$$

が成り立つ．いい換えれば，$S_{10^6}/10^6$ 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.

[注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....

[注5] $x \approx y$ は x と y の値がほぼ等しいの意.

例 2.3 では (2.1) 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 $100\times10^6 = 10^8 = 1$ 億回投げて S_{10^6} 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 2.3 の S_{10^6} を硬貨投げの確率空間 $(\Omega := \{0,1\}^{10^8}, 2^\Omega, P_{10^8})$ 上で次のように実現しよう: まず，関数 $X : \{0,1\}^{100} \to \{0,1\}$ を

$$X(\xi_1,\dots,\xi_{100}) := \max_{1\le l\le 100-5} \prod_{i=l}^{l+5} \xi_i, \qquad (\xi_1,\dots,\xi_{100}) \in \{0,1\}^{100},$$

と定義する．これは $(\xi_1,\dots,\xi_{100})$ の中で 1 が 6 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 $X_k : \{0,1\}^{10^8} \to \{0,1\}$，$k = 1,2,\dots,10^6$，を

$$X_k(\omega) := X(\omega_{100(k-1)+1},\dots,\omega_{100k}), \quad \omega = (\omega_1,\dots,\omega_{10^8}) \in \{0,1\}^{10^8},$$

と定義し，さらに $S_{10^6} : \{0,1\}^{10^8} \to \mathbb{Z}$ を

$$S_{10^6}(\omega) := \sum_{k=1}^{10^6} X_k(\omega), \quad \omega \in \{0,1\}^{10^8},$$

と定義する．S_{10^6} の例外値を与える ω の集合 A_0 を

$$A_0 := \left\{ \omega \in \{0,1\}^{10^8} \;\middle|\; \left| \frac{S_{10^6}(\omega)}{10^6} - p \right| \ge \frac{1}{200} \right\} \tag{2.2}$$

とすれば，(2.1) より $P_{10^8}(A_0) \le 1/100$ となる．そこで例 2.3 は次のような賭けとして捉えることができる．

例 2.4 一つの $\omega \in \{0,1\}^{10^8}$ 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 1/100 以下である．

2.3 乱数の問題

例 2.1 と例 2.4 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 2.4 の賭けを実行するために一つの $\omega \in \{0,1\}^{10^8}$ を選ぼうとしているとしよう．しかし，じつは $\{0,1\}^{10^8}$ 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，たとえリスク評価が $P_{10^8}(A_0) \ll 1$ 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 $\omega \in \{0,1\}^{10^8}$ は $10^8 = 1$ 億ビット，すなわち約 12MB のデータであり，アリスがこれを選ぶにはコンピュータを用いるよりないが，それとて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 $\{0,1\}^{10^8}$ 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．[注6] このとき，アリスが自分の意志で選ぶことのできる $\omega \in \{0,1\}^{10^8}$ の個数はせいぜい 2^{1001} 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．) $\{0,1\}^{10^8}$ の元の個数は 2^{10^8} もあるのだから，アリスの選ぶことのできる $\omega \in \{0,1\}^{10^8}$ がいかに僅かであるかが分かるだろう．たとえ，アリスが 10^8-10 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^{10^8-9} であり，これでも $\{0,1\}^{10^8}$ 全体の個数 2^{10^8} の $1/512$ に過ぎない．いい換えると $\{0,1\}^{10^8}$ 全体の少なくとも $511/512$ は 10^8-9 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．一方で言うまでもなく，任意の $\{0,1\}^{10^8}$ の元は「それ自身を入力する」という方法で 10^8 ビットの入力で指定することができる．

短い入力で指定できる $\{0,1\}$-列は，その列に何らかの規則性があるからそうできる，と考えられる．逆に規則性がないと指定するために長い入力が必要になるはずである．そこで，それ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 $\{0,1\}$-列を**乱数**と呼ぶ．[注7]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．

例2.4の賭けのリスク評価 $P_{10^8}(A_0) \leq 1/100$ は，どの $\omega \in \{0,1\}^{10^8}$ も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．だから，$P_{10^8}(A_0) \leq 1/100$ に実質的な意味を持たせるためには，ω は主として圧倒的多数を占める乱数たちから選ばれるべきであろう．これがモンテカルロ法に乱数が必要だといわれる理由である．しかし，乱数は，それほど多いにもかかわらず，指定することがあまりに大変なために，事実上，アリスはそのうち一つさえ選ぶことができない．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困難なのである．

2.4 疑似乱数生成器

疑似乱数生成器は乱数を用いずに確率変数 S の一般的な値を高い確率でサンプリングするための道具である．その持つべき性質，すなわち安全性，について述べる．

2.4.1 定義と役割

例2.4の賭けを実行するとき，$S_{10^6}(\omega)$ を計算するためにアリスはとにかく一つの $\omega \in \{0,1\}^{10^8}$ を選ばなければならない．それには何か道具が必要である．ここでは最もよく用いられる道具，すなわち疑似乱数生成器を用いる場合を考えよう．

[注6] じつは，そうした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．

[注7] 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．

定義 2.5 $n < L$ 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)[注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．それが疑似乱数生成器である．疑似乱数生成器は**初期化**[注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 $g(\omega') \in \{0,1\}^L$ 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 2.6 例 2.4 において，アリスがたとえばある疑似乱数生成器 $g : \{0,1\}^{238} \to \{0,1\}^{10^8}$ を使うとしよう.[注10] アリスは g の種 $\omega' \in \{0,1\}^{238}$ 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 238 ビット (アルファベットおよそ 30 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 $S_{10^6}(g(\omega'))$ 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 $\omega \in \{0,1\}^{10^8}$ 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る．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 2.1 で 100 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．

2.4.2 安全性

例 2.6 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 $\omega' \in \{0,1\}^{238}$ を自由に選ぶことができる．今の場合，リスクは確率

$$P_{238}\left(\left|\frac{S_{10^6}(g(\omega'))}{10^6} - p\right| \geq \frac{1}{200}\right) \tag{2.3}$$

であり，次にこれを計算する必要がある．もちろん，確率 (2.3) 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 $S_{10^6}(g(\omega'))$ 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 (少し曖昧な) 定義を設けよう; 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

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

[注8] 初期値ともいう.
[注9] ランダマイズ (randomize) ともいう.
[注10] 238 という半端な数の由来は例 2.9 で明らかになる.

が成り立つことをいう．

例 2.6 において，もし $g : \{0,1\}^{238} \to \{0,1\}^{10^8}$ が (2.2) 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 $\omega' \in \{0,1\}^{238}$ 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，長大な乱数は必要でない．いい換えると，S の一般的な値をサンプリングしたい際に g を用いてもリスクが大きくならないという意味で，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．モンテカルロ法におけるサンプリングの問題 — すなわち乱数の問題 — はこのような安全な疑似乱数生成器を見出すことで解決される．

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，$P_L(\omega \in A_g) \le 2^{n-L}$ 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．

2.4.3　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」のクラスであると考えられる．

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 § 2.3 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．$L = 10^8$ の場合だと，大部分の $A \subset \{0,1\}^{10^8}$ のプログラムの長さはほぼ 2^{10^8} 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．ただし，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 (§ 4.1).[注11]

例 2.6 の場合に g がもし計算量的に安全であれば，$S_{10^6}(\omega)$ と $S_{10^6}(g(\omega'))$ の分布は十分近い．実際，S_{10^6} 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

[注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より実際的な意味がある．

対して $A(c_1, c_2) := \{\omega | c_1 \le S_{10^6}(\omega) \le c_2\}$ とすれば，$\omega \in A(c_1, c_2)$ は計算量的に判定可能である．従って p' の値が何であっても

$$P_{10^8}\left(\left|\frac{S_{10^6}(\omega)}{10^6} - p'\right| \ge \frac{1}{200}\right) \approx P_{238}\left(\left|\frac{S_{10^6}(g(\omega'))}{10^6} - p'\right| \ge \frac{1}{200}\right)$$

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 (§ 4.1.3)．また，この概念は正確に述べればある種の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．

2.5 モンテカルロ積分

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

$$\mathbf{E}[X] = \frac{1}{2^m}\sum_{\xi\in\{0,1\}^m} X(\xi)$$

を求める問題を考える．ここで m が小さいときは上式の右辺を直接計算すればよいが，m が大きいとき (たとえば $m = 100$)，計算量が莫大になりそれは事実上不可能になる．**モンテカルロ積分** (Monte-Carlo integration) とは，そのような場合に，大数の法則を用いて確率変数の平均を推定する手法をいう (例 2.3)．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していて，それらはほとんどの場合，モンテカルロ積分である．

2.5.1 i.i.d.-サンプリング

例 2.3 を一般的な設定の下で述べると以下のようになる．X の独立な N 個のコピー $\{X_k\}_{k=1}^N$ 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

$$X_k(\omega) := X(\omega_k), \quad \omega_k \in \{0,1\}^m, \quad \omega = (\omega_1, \ldots, \omega_N) \in \{0,1\}^{Nm}, \tag{2.4}$$

$$S_N(\omega) := \sum_{k=1}^N X_k(\omega). \tag{2.5}$$

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 $\mathbf{V}[S_N/N] = \mathbf{V}[X]/N$ を満たす．S_N/N をサンプリングすることによって X の平均を推定する方法を **i.i.d.-サンプリング**[注12] と呼ぶ．このときのリスクをチェビシェフの不等式

$$P_{Nm}\left(\left|\frac{S_N(\omega)}{N} - \mathbf{E}[X]\right| \ge \delta\right) \le \frac{\mathbf{V}[X]}{N\delta^2}, \quad \delta > 0 \tag{2.6}$$

[注12] $\{X_k\}_{k=1}^N$ が i.i.d. 確率変数列なのでそう呼ぶ．

で評価しよう.[注13] このことは

$$A_1 := \left\{ \omega \in \{0,1\}^{Nm} \;\middle|\; \left| \frac{S_N(\omega)}{N} - \mathbf{E}[X] \right| \geq \delta \right\} \tag{2.7}$$

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.

2.5.2　ランダム-ワイル-サンプリング

モンテカルロ積分の場合，サンプリングの対象となる確率変数は S_N であり，確率変数の中でも特殊な形をしている．その事実を利用すれば A_1 に対し安全な疑似乱数生成器を具体的に構成することができる.

定義 2.7 (cf. [44, 52])　$j \in \mathbb{N}^+$ とし

$$\begin{aligned} Z_k(\omega') &:= \lfloor x + k\alpha \rfloor_m \in D_m \cong \{0,1\}^m, \quad k = 1, 2, 3, \ldots, 2^{j+1}, \\ \omega' &:= (x, \alpha) \in D_{m+j} \times D_{m+j} \cong \{0,1\}^{2m+2j}, \end{aligned}$$

を定義する.[注14] これを用いて疑似乱数生成器 $g : \{0,1\}^{2m+2j} \to \{0,1\}^{Nm}$，ただし $N \leq 2^{j+1}$，を

$$g(\omega') := (Z_1(\omega'), Z_2(\omega'), \ldots, Z_N(\omega')) \in D_m^N \cong \{0,1\}^{Nm}, \tag{2.8}$$

と定義する．この疑似乱数生成器を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.[注15]

定理 2.8[注16] (2.8) の疑似乱数生成器 $g : \{0,1\}^{2m+2j} \to \{0,1\}^{Nm}$ は (2.5) の S_N に対して

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

を満たす．ただし ω'，ω はそれぞれ P_{2m+2j}，P_{Nm} に従うものとする．これより (2.6) と同様のチェビシェフの不等式

$$P_{2m+2j}(g(\omega') \in A_1) = P_{2m+2j}\left(\left| \frac{S_N(g(\omega'))}{N} - \mathbf{E}[X] \right| \geq \delta \right) \leq \frac{\mathbf{V}[X]}{N\delta^2}$$

が成り立つ．この意味で g は (2.7) の A_1 に対して安全な疑似乱数生成器である.

[注13] $\mathbf{E}[X]$ が未知であるのと同様に $\mathbf{V}[X]$ 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 $\mathbf{V}[X]$ の上からの評価が得られれば (たとえば 例 2.3 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.

[注14] 記号 $\lfloor \cdot \rfloor_m$ および同一視 $\{0,1\}^m \cong D_m$ などについては定義 1.1 を見よ.

[注15] [44, 52] 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.

[注16] この定理は後述の定理 5.5 の離散化で得られたものである.

証明. 第 1 段 (cf. [44, 52]) $D_{m+j} \times D_{m+j}$ 上の一様 (直積) 確率測度 $P_{(m+j)} \otimes P_{(m+j)}$ の下で $\{Z_k\}_{k=1}^{2^{j+1}}$ はペアごとに独立であり，各 Z_k は D_m で一様分布することを示そう．そのためには $F, G : \mathbb{T}^1 \to \mathbb{R}$ が $\mathcal{B}_m$-可測のとき，$1 \le k < k' \le 2^{j+1}$ に対して

$$\mathbf{E}[F(Z_{k'})G(Z_k)] = \int_0^1 F(t)dt \int_0^1 G(s)ds \tag{2.9}$$

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

平均の定義と F, G が $\mathcal{B}_m$-可測であることより

$$\begin{aligned}
\mathbf{E}[F(Z_{k'})G(Z_k)] &= \frac{1}{2^{2m+2j}} \sum_{q=1}^{2^{m+j}} \sum_{p=1}^{2^{m+j}} F\left(\frac{p}{2^{m+j}} + \frac{k'q}{2^{m+j}}\right) G\left(\frac{p}{2^{m+j}} + \frac{kq}{2^{m+j}}\right) \\
&= \frac{1}{2^{2m+2j}} \sum_{q=1}^{2^{m+j}} \sum_{p=1+kq}^{2^{m+j}+kq} F\left(\frac{p}{2^{m+j}} + \frac{(k'-k)q}{2^{m+j}}\right) G\left(\frac{p}{2^{m+j}}\right) \\
&= \frac{1}{2^{2m+2j}} \sum_{q=1}^{2^{m+j}} \sum_{p=1}^{2^{m+j}} F\left(\frac{p}{2^{m+j}} + \frac{(k'-k)q}{2^{m+j}}\right) G\left(\frac{p}{2^{m+j}}\right).
\end{aligned} \tag{2.10}$$

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

$$\frac{1}{2^{m+j}} \sum_{q=1}^{2^{m+j}} F\left(\frac{p}{2^{m+j}} + \frac{(k'-k)q}{2^{m+j}}\right) = \frac{1}{2^{m+j}} \sum_{q=1}^{2^{m+j}} F\left(\frac{p}{2^{m+j}} + \frac{lq}{2^{m+j-i}}\right). \tag{2.11}$$

各 $r = 1, 2, 3, \ldots, 2^{m+j-i}$ 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

$$\begin{aligned}
\frac{1}{2^{m+j}} \sum_{q=1}^{2^{m+j}} F\left(\frac{p}{2^{m+j}} + \frac{lq}{2^{m+j-i}}\right) &= \frac{1}{2^{m+j-i}} \sum_{r=1}^{2^{m+j-i}} F\left(\frac{p}{2^{m+j}} + \frac{r}{2^{m+j-i}}\right) \\
&= \frac{1}{2^{m+j-i}} \sum_{r=1}^{2^{m+j-i}} F\left(\frac{r}{2^{m+j-i}}\right) \\
&= \int_0^1 F(t)dt.
\end{aligned} \tag{2.12}$$

従って (2.10)(2.11)(2.12) から

$$\begin{aligned}\mathbf{E}[F(Z_{k'})G(Z_k)] &= \frac{1}{2^{m+j}}\sum_{p=1}^{2^{m+j}}\left(\frac{1}{2^{m+j}}\sum_{q=1}^{2^{m+j}}F\left(\frac{p}{2^{m+j}}+\frac{(k'-k)q}{2^{m+j}}\right)\right)G\left(\frac{p}{2^{m+j}}\right)\\ &= \frac{1}{2^{m+j}}\sum_{p=1}^{2^{m+j}}\left(\frac{1}{2^{m+j}}\sum_{q=1}^{2^{m+j}}F\left(\frac{p}{2^{m+j}}+\frac{lq}{2^{m+j-i}}\right)\right)G\left(\frac{p}{2^{m+j}}\right)\\ &= \int_0^1 F(t)dt\cdot\frac{1}{2^{m+j}}\sum_{p=1}^{2^{m+j}}G\left(\frac{p}{2^{m+j}}\right) = \int_0^1 F(t)dt\int_0^1 G(s)ds.\end{aligned}$$

以上で (2.9) が示された.

第 2 段　まず，各 $Z_k(\omega')$ は $\{0,1\}^m$ 上で一様分布するから

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

であることが分かる．次にペアごとの独立性によって

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

以上から，g は要請された性質を持つことが分かる．　□

例 2.9　RWS を用いて § 2.2.2 の例題を解くことができる．例 2.6 (§ 2.4.1) において，$m = 100$，$N = 10^6$，$j = 19$，とした定義 2.7 の疑似乱数生成器 $g : \{0,1\}^{238} \to \{0,1\}^{10^8}$ を用いるとき,[注17] リスク評価は

$$P_{238}\left(\left|\frac{S_{10^6}(g(\omega'))}{10^6}-p\right|\ge\frac{1}{200}\right)\le\frac{1}{100} \tag{2.13}$$

のように得られる (cf. (2.3))．アリスは自分の意思でどの種 $\omega' \in \{0,1\}^{238}$ でも容易に選ぶことができるから，この場合は長大な乱数は必要ない.

具体的に $S_{10^6}(g(\omega'))$ を求めた例を挙げよう．(アリスの代わりに) 筆者が選んだ種 $\omega' = (x,\alpha) \in D_{119}\times D_{119} \cong \{0,1\}^{238}$ は次の通りである (2 進数表示) ;

[注17] ここでは $2^{j+1} = 2^{20} > 10^6 = N$ であり $2m + 2j = 238$．現実のモンテカルロ法ではサンプルサイズ N があらかじめ定まっていることは少なく，数値実験を行いながら適切な N を見つけていく場合が多い．そのような場合に備えて実際には j を少々大きめに与えておくとよい.

```
x  =  0.1110110101 1011101101 0100000011 0110101001 0101000100
      0101111101 1010000000 1010100011 0100011001 1101111101
      1101010011 111100100
α  =  0.1100000111 0111000100 0001101011 1001000001 0010001000
      1010101101 1110101110 0010010011 1000000011 0101000110
      0101110010 010111111
```

このときコンピュータよって $S_{10^6}(g(\omega')) = 546,177$ と計算された (C 言語による実装については § 6.1.1 を見よ). 従って，この場合，

$$\frac{S_{10^6}(g(\omega'))}{10^6} \;=\; 0.546177$$

が求める確率 p の推定値である．

注意 2.10 ここではリスク評価をチェビシェフの不等式 (2.1)(2.13) によって行ったが，実際には中心極限定理に訴えてもっと精密にすることができる．このことについては § 5.2 で詳しく述べる．

注意 2.11 RWS の場合，アリスがどんな種 $\omega' = (x, \alpha) \in \{0,1\}^{2m+2j}$ 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 はほぼ間違いなく失敗することがすぐ分かる．

注意 2.12 もっと大規模なモンテカルロ積分で $2m+2j \gg 1$ の場合だと，再び乱数の問題によって，アリスは RWS の種 $\omega' \in \{0,1\}^{2m+2j}$ 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 $g' : \{0,1\}^n \to \{0,1\}^{2m+2j}$, $n \ll 2m+2j$, を用いて $\omega' \in \{0,1\}^{2m+2j}$ を選ぶ破目になる．この場合，合成された疑似乱数生成器 $g \circ g' : \{0,1\}^n \to \{0,1\}^{2m+2j} \to \{0,1\}^{Nm}$ が A_1 に対して安全になるかどうかは分からない．詳しくは § 5.2.2 を見よ．

例 2.13 例 2.2 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 $S' := \sum_{k=1}^{40000} \mathbf{1}_{\{X'_k < c\}}$ とすれば

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

だからチェビシェフの不等式より

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

これより $\min_{1\leq k\leq 40000} X'_k$ 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．

2.6 数理統計学の視点から

2.6.1 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** (random sampling)[注18]を行うことを重要と考えるからである．実際，RWS の場合は，注意 2.11 で述べたことを逆手にとって，悪い種 ($\alpha = (0,0,\ldots,0) \in \{0,1\}^{m+j}$) を選んで賭けにわざと負けることができる．すなわち，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 2.9 の ω' を選ぶときは，硬貨を 238 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 238 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，無作為なサンプリングが実行される．じつは，例 2.9 の ω' は，実際，そのようにして得られた種である．

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19]

2.6.2 疑似乱数の検定

[16] をはじめ，夥しい数の疑似乱数の検定の多くは次の手順で行われている．

(1) 検定項目 (連の検定，ポーカー検定など) を決める．

(2) 無作為に選ばれた複数の種 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．

[注18] ここでいう random sampling はプレーヤの意思に依らないサンプリングを意味する．i.i.d.-サンプリングや RWS のように，確率変数のサンプリングを行うという意味で random sampling という用語を使う場合があるので注意されたい．(本書では，モンテカルロ法を賭けだと規定しているので，確率変数のサンプリングはプレーヤの意思で行うことを基本としている．)

[注19] この解決方法はじつは古くからある乱数表の使い方 — 乱数表のページや数を拾い始める箇所の決め方は無作為に，あとはルールに従って決定論的に数を拾っていくというやり方 ([16] p.7) — によく似ている．

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るといえるだろう．

第3章　乱数

与えられた有限 {0, 1}-列 x を出力するコンピュータプログラムを考える．もし，x が短いプログラムで出力されるなら x は規則的，逆に長いプログラムでなければ出力されないなら不規則的であると解釈されよう．そこで非常に長いプログラムでなければ出力されないような {0, 1}-列を乱数と定義すればよい．このアイデアを正確にかつ普遍的に定義するために部分帰納的関数という概念を用いる ([5, 22, 23, 24, 25, 31], cf. [28, 64])．この章では，二つの基本定理 (定理 3.3 と定理 3.6) の詳しい証明は専門書 ([20, 53] など) に譲るものの，それらをもとに他の定理を証明する．とくに，与えられた {0, 1}-列が乱数であるかどうかを判定することが不可能であることを示した定理 3.15 と，乱数であることと万能検定と呼ばれるある普遍的な検定に採択されることの同値性を述べたマルティン=レーフの定理 (定理 3.20) が重要である．

乱数の概念は現実的な問題解決には直結しないかも知れないが，それを認識することはモンテカルロ法のみならず確率論そのものの深い理解に有益である (§ 3.6).

3.1　部分帰納的関数

現代のコンピュータで扱われるデータの種類は多彩である．入力としては，キーボード，マウス，スキャナー，ビデオカメラなどからのデータ，出力としては文書，画像，音声，映像，電子機器の制御データ．．．．しかしそれらはすべて突き詰めれば 2 進データ，すなわち有限 {0, 1}-列に過ぎない．[注1]有限 {0, 1}-列は 2 進数表示を通じて自然数と対応させることができる (§ 3.1.3) から，コンピュータで扱われるデータは入力も出力も本質的にすべて自然数とみなすことができる．すなわちコンピュータのあらゆる動作は数学的には $f : \mathbb{N} \to \mathbb{N}$ なる関数として捉えることができるわけである．

コンピュータの動作はインストールするプログラム (ソフトウェア) で決定される．そしてすべての入出力データと同様にプログラムもまた有限 {0, 1}-列，従って自然数とみなすことができる．従ってプログラム全体は可算集合である．$f : \mathbb{N} \to \mathbb{N}$ なる関数の全体が非可算集合であるので，そのうち可算個だけがコンピュータの動作によって実現可能な関数なのである．

コンピュータの動作 $f : \mathbb{N} \to \mathbb{N}$ を数学的に表現するために，部分帰納的関数という概念が用いられる．それにより，コンピュータのすべての動作 (無限ループに陥って停止しなくなる場合も含む) が表現できる．ここでは後の準備として，部分帰納的関数に関連するいくつかの概念や定理を紹介する．

[注1] ここでは，入力または出力が永遠に続くようなデータ処理については考えない．

3.1.1　原始帰納的関数と部分帰納的関数

定義 3.1　(**原始帰納的関数**，primitive recursive function，[20, 53])

1. (基本関数)

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

   は原始帰納的関数.

2. (合成)
   $g : \mathbb{N}^m \to \mathbb{N}$，$g_j : \mathbb{N}^n \to \mathbb{N}$，$j = 1,\ldots,m$，が原始帰納的関数ならば,

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

   は原始帰納的関数.

3. (帰納法)
   $g : \mathbb{N}^n \to \mathbb{N}$，$h : \mathbb{N}^{n+2} \to \mathbb{N}$ が原始帰納的関数ならば，$f : \mathbb{N}^{n+1} \to \mathbb{N}$,

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

   は原始帰納的関数.

4. (直積)
   $g_j : \mathbb{N}^{n_j} \to \mathbb{N}$，$j = 1,\ldots,m$，が原始帰納的関数のとき,

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

   は原始帰納的関数.

5. 基本関数から，合成，帰納法，および直積の操作を有限回適用して得られる関数 ($\mathbb{N}^m \to \mathbb{N}^n$)，およびそれらだけが原始帰納的関数.

定義 3.2　(**部分帰納的関数**，partial recursive function，[20, 53])

1. (μ-作用素)
   $p : \mathbb{N}^{n+1} \to \mathbb{N}$ が部分関数[注2]のとき，$\mu_y(p(\bullet,\cdots,\bullet,y)) : \mathbb{N}^n \to \mathbb{N}$ を

$$\mu_y(p(x_1,\ldots,x_n,y)) := \begin{cases} \min A_p(x_1,\ldots,x_n) & (A_p(x_1,\ldots,x_n) \neq \emptyset), \\ \text{定義しない} & (A_p(x_1,\ldots,x_n) = \emptyset) \end{cases}$$

   とする．ただし，$A_p(x_1,\ldots,x_n)$ は次の集合である;

$$A_p(x_1,\ldots,x_n) := \left\{ y \in \mathbb{N} \;\middle|\; \begin{array}{l} p(x_1,\ldots,x_n,y) = 0 \text{ かつ } 0 \le z \le y \text{ について} \\ p(x_1,\ldots,x_n,z) \text{ が定義されている} \end{array} \right\}.$$

[注2] 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．$\mathbb{N}^n$ 全体で定義されている場合は全域関数という.

2. 基本関数から，合成，帰納法，直積，およびμ-作用素を有限回適用して得られる関数 ($\mathbb{N}^m \to \mathbb{N}^n$)，およびそれらだけが部分帰納的関数.

原始帰納的関数は部分帰納的関数である．μ-作用素を用いない (従って原始帰納的関数である) か，たとえμ-作用素を使っていても，条件 $\{y \in \mathbb{N} \mid p(x_1, \ldots, x_n, y) = 0\} \neq \emptyset$ が満たされるときに限ってμ-作用素 $\mu_y(p(x_1, \ldots, x_n, y))$ が用いられているとき，部分帰納的関数は全域で定義される．これを**全域帰納的関数** (total recursive function) という.

実際のコンピュータの任意の動作 $f : \mathbb{N} \to \mathbb{N}$ は部分帰納的関数として捉えることができる．また，任意の帰納的関数は原理的には[注3]コンピュータで実現することができる.

3.1.2 クリーネの標準形

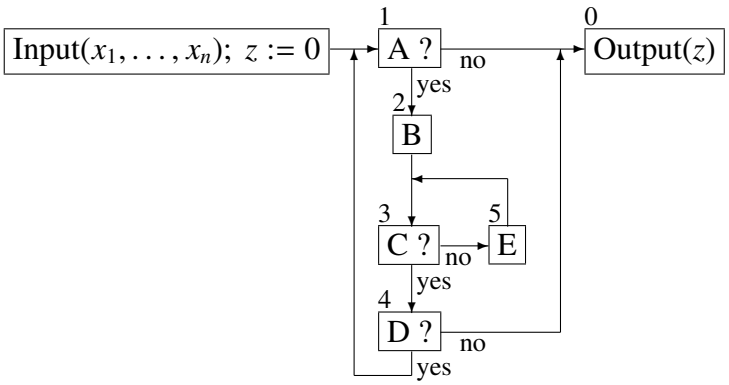
定理 3.3 (クリーネ (Kleene)[注4] の標準形) 任意の部分帰納的関数 $f : \mathbb{N}^n \to \mathbb{N}$ に対して，ある二つの原始帰納的関数 $g, p : \mathbb{N}^{n+1} \to \mathbb{N}$ が存在して

$$f(x_1, x_2, \ldots, x_n) = g(x_1, x_2, \ldots, x_n, \mu_y(p(x_1, x_2, \ldots, x_n, y))), \quad (x_1, x_2, \ldots, x_n) \in \mathbb{N}^n, \tag{3.1}$$

が成り立つ.

定理 3.3 の詳しい証明は [6, 53] などを参照されたい．ここでは，証明のアイデアを一つの例によって説明するに留める．アイデアの核心は f を実現するプログラムがループ (部分帰納的関数でいえばμ-作用素に相当) を複数持っていたとしても，それをただ一つのループでまとめてしまう方法を見つけることである.

図 3.1: 流れ図 (A)



[注3] 記憶容量と計算時間に糸目を付けなければ，の意．抽象的なコンピュータのモデルであるチューリング機械で考えるのが計算機科学においては標準的である.

[注4] このカタカナ表記には問題があるらしいが，日本ではこのように呼ばれている.

図 3.1(流れ図 (A)[注5]) は f を計算するプログラムの流れ図であるとしよう．A ? C ? D ? は分岐の条件を表し，B E はループを含まない計算処理 (数学的には原始帰納的関数の計算) のあと出力用変数 z の値を設定していることを表す．このプログラムは，主ループ A ?→B→C ?→D ? →A ? のほかに入れ子になったループ C ?→E→C ? があり，また D ? から主ループを抜け出る道もある．

このように複数のループの存在するプログラムでも，新たな変数 u を導入することによって，一つのループにまとめることができる．その準備のために，各処理 A ? C ? D ? B E と出力部分に u が参照する番号 (0 ~ 5) を対応させておく．それを流れ図 (A) の各処理を表す箱の左上に記した．

図 3.2: 流れ図 (B)

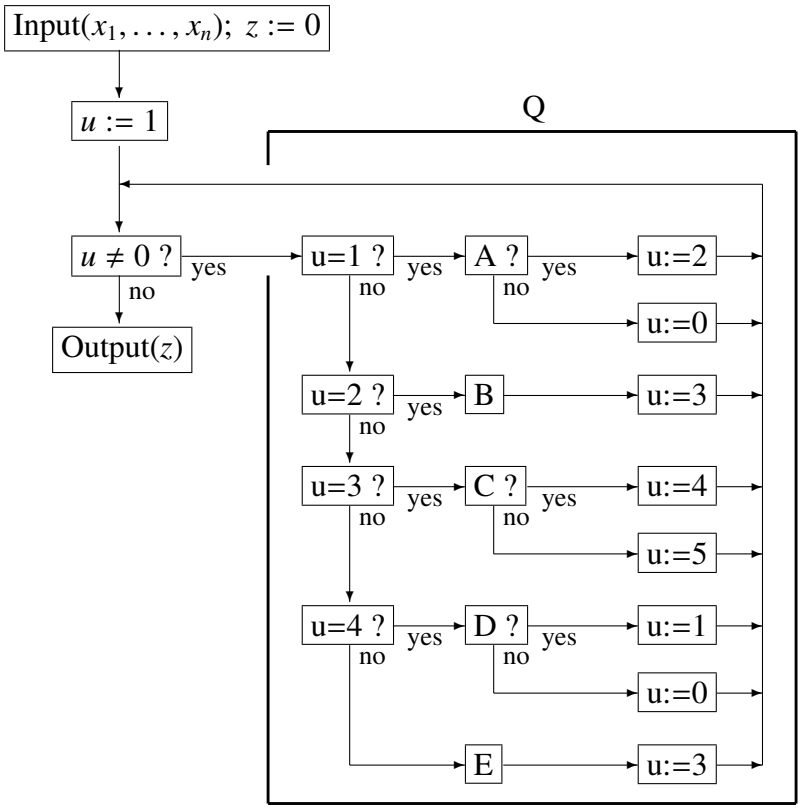


図 3.2(流れ図 (B)) が u を用いてループを一つにまとめたものである．流れ図 (A) が関数 f を計算するプログラムならば流れ図 (B) もそうであることを確認せよ．

次に，流れ図 (B) がさらに (3.1) という式で表わされることを示そう．流れ図 (B) において太い線で囲った部分の計算処理をまとめて Q と呼ぶことにする．入力 $(x_1, \ldots, x_n)$ の下で y 回だけ Q が実行されたときの出力用変数 z の値を $g(x_1, \ldots, x_n, y)$ と定義する．また，入力 $(x_1, \ldots, x_n)$ の下で y 回だけ Q が実行されたときの u の値を $p(x_1, \ldots, x_n, y)$ と定義する．このとき (3.1) が成り立つことが分かる．

[注5] 流れ図 (A)(B) は教科書 [53] p.12 図 3，p.13 図 4 をそれぞれ少し修正したものである．

3.1.3 標準的順序

有限 $\{0,1\}$-列は 2 進表示を通じて自然数と見なすことができる．このことを正確に表現しておこう．

$\{0,1\}^* := \bigcup_{n\in\mathbb{N}}\{0,1\}^n$ とする．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 $\{0,1\}$-列も仮想的に考えてこれを**空語** (empty word) と呼ぶ．$\{0,1\}^*$ に**標準的順序** (canonical order) を入れる; $x, y \in \{0,1\}^*$ において，x の方が y より長い $\{0,1\}$-列のとき，$x > y$ 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．以下では，この順序によって $\{0,1\}^*$ を $\mathbb{N}$ と同一視する．すなわち，空語 $= 0$，$(0) = 1$，$(1) = 2$，$(0,0) = 3$，$(0,1) = 4$，$(1,0) = 5$，$(1,1) = 6$，$(0,0,0) = 7$，....

3.1.4 枚挙に関する定理と停止問題

与えられた多変数関数に同等な 1 変数関数を作るために便利な関数 $G^n : (\{0,1\}^*)^n \to \{0,1\}^*$ を導入しよう．まず $x_i \in \{0,1\}^{m_i}$，$i = 1,2$，が

$$x_i = (x_{i1}, x_{i2}, \ldots, x_{im_i}), \quad x_{ij} \in \{0,1\}, \quad i = 1,2,$$

であるとき

$$G^2(x_1, x_2) := (x_{11}, x_{11}, x_{12}, x_{12}, \ldots, x_{1m_1}, x_{1m_1}, 0, 1, x_{21}, x_{22}, \ldots, x_{2m_2}) \tag{3.2}$$

と定義する．以下，帰納的に

$$G^n(x_1, x_2, \ldots, x_n) := G^2(x_1, G^{n-1}(x_2, \ldots, x_n)), \quad n = 3, 4, \ldots,$$

と定義する．簡単のため

$$\langle x_1, \ldots, x_n\rangle := G^n(x_1, \ldots, x_n)$$

と略記することもある．逆関数は $u = \langle x_1, x_2, \ldots, x_n\rangle$ に対して

$$(u)_i^n := x_i, \quad i = 1, 2, \ldots, n,$$

と書く.[注6]

たとえば $(1) \in \{0,1\}^1$ と $(1,1,0,1,1) \in \{0,1\}^5$ に対して

$$\langle (1), (1,1,0,1,1)\rangle = (1,1,0,1,1,1,0,1,1) =: u \in \{0,1\}^9$$

である．このとき $(u)_1^2 = (1)$ および $(u)_2^2 = (1,1,0,1,1)$ である．同時に $\langle (1),(1),(1)\rangle = u$ でもあるから $(u)_1^3 = (u)_2^3 = (u)_3^3 = (1)$ である．

定義 3.4 $U \subset \mathbb{N}^n$ が**帰納的可算集合** (recursively enumerable set) であるとは，ある部分帰納的関数 $f : \mathbb{N}^n \to \mathbb{N}$ が存在して，U が f の定義域に等しいことをいう．すなわち

$$U = \{(x_1, x_2, \ldots, x_n) \in \mathbb{N}^n \mid \exists y \text{ s.t. } f(x_1, x_2, \ldots, x_n) = y\}.$$

[注6] これらの関数はゲーデル (Gödel) 関数と呼ばれるものの簡単な類似である．

帰納的可算集合という用語の由来は次の定理 (とくにその中の (ii)(iii)) にある.

定理 3.5　集合 $U \subset \mathbb{N}^n$ に関する次の条件は互いに同値である.
(i) U は帰納的可算集合である.
(ii) U は空集合であるか，またはある原始帰納的関数 $f : \mathbb{N} \to \mathbb{N}^n$ の像に等しい，すなわち $U = f(\mathbb{N})$.
(iii) U はある部分帰納的関数 $f : \mathbb{N} \to \mathbb{N}^n$ の像に等しい，すなわち $U = f(\mathbb{N})$.
(iv) ある原始帰納的関数 $p : \mathbb{N}^{n+1} \to \mathbb{N}$ が存在して

$$U = \{(x_1, x_2, \ldots, x_n) \mid \exists y \text{ s.t. } p(x_1, x_2, \ldots, x_n, y) = 0\}.$$

(v) ある全域帰納的関数 $p : \mathbb{N}^{n+1} \to \mathbb{N}$ が存在して

$$U = \{(x_1, x_2, \ldots, x_n) \mid \exists y \text{ s.t. } p(x_1, x_2, \ldots, x_n, y) = 0\}.$$

証明. 簡単のため，$n = 1$ の場合について証明する.

(i)$\Longrightarrow$(ii): $U \neq \emptyset$ とする．定理 3.3 により，ある原始帰納的関数 $g, p : \mathbb{N}^2 \to \mathbb{N}$ が存在して

$$U = \{x \in \mathbb{N} \mid \exists y \text{ s.t. } g(x, \mu_z(p(x, z))) = y\}.$$

そこで，任意の $a \in U$ を固定して

$$h(u) = \begin{cases} (u)_1^2 & (u \in G^2(\mathbb{N}^2) \text{ かつ } p((u)_1^2, (u)_2^2) = 0) \\ a & (\text{そうでないとき}) \end{cases}$$

という関数を考えると，これは原始帰納的関数であり，$h(\mathbb{N}) = U$ を満たす.

(ii)$\Longrightarrow$(iii): 明らか.

(iii)$\Longrightarrow$(iv): 定理 3.3 により，$f(x) = g(x, \mu_y(q(x, y)))$，$g, q$ は原始帰納的関数，で $U = f(\mathbb{N})$ であるとする．すなわち

$$U = \{z \in \mathbb{N} \mid \exists x, y \text{ s.t. } z = g(x, y),\ q(x, y) = 0,\ \forall w < y,\ q(x, w) > 0\}.$$

ここで

$$q'(x, y, z) := \begin{cases} 0 & (z = g(x, y),\ q(x, y) = 0,\ \forall w < y,\ q(x, w) > 0) \\ 1 & (\text{そうでないとき}) \end{cases}$$

とすれば，これは原始帰納的関数．さらに,

$$p(z, u) := \begin{cases} q'((u)_1^2, (u)_2^2, z) & (u \in G^2(\mathbb{N}^2)) \\ 1 & (u \notin G^2(\mathbb{N}^2)) \end{cases}$$

とすれば，これも原始帰納的関数で $U = \{z \in \mathbb{N} \mid \exists u \text{ s.t. } p(z, u) = 0\}$.

(iv)$\Longrightarrow$(v): 明らか.

(v)$\Longrightarrow$(i): $f(x) := \mu_y(p(x, y))$ とすれば f は全域帰納的関数であり,

$$U = \{x \in \mathbb{N} \mid \exists y \text{ s.t. } y = f(x)\}.$$

□

定理 3.6 (枚挙定理) ある部分帰納的関数 $univ_n : \mathbb{N} \times \mathbb{N}^n \to \mathbb{N}$ が存在して，任意の部分帰納的関数 $f : \mathbb{N}^n \to \mathbb{N}$ に対して次を満たすような $e_f \in \mathbb{N}$ が存在する;

$$univ_n(e_f, x_1, \ldots, x_n) \;=\; f(x_1, \ldots, x_n), \qquad (x_1, \ldots, x_n) \in \mathbb{N}^n.$$

定理 3.6 における $univ_n$ は**枚挙関数** (enumerating function) または**万能関数** (universal function)，e_f は f 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 [20, 53]) に譲るとして，ここでは証明のアイデアを述べる．まず，与えられた部分帰納的関数 f がどのようにして基本関数から作られるのかを具体的に符号化し，それに一つの自然数 e_f を対応させる．具体的には万能チューリング機械 (universal Turing machine，cf. [20]) のプログラムを $\{0, 1\}$-列として書き，さらにそれを 2 進数の自然数と見ればよい．ゲーデル数 e_f はそのような自然数なのである．枚挙関数 $univ_n(e, x_1, \ldots, x_n)$ は，まず，e が n 変数の部分帰納的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 の表す部分帰納的関数を基本関数から再現して，それに $x_1, \ldots, x_n$ を代入してその答えを返すような関数である．

枚挙定理の成立には“部分関数”という概念が本質的である．次の定理がある．

定理 3.7 枚挙関数 $univ_n$ の拡張となっている任意の全域関数は帰納的でない．

証明. 背理法による．[注7] $univ_n$ の拡張となっている全域帰納的関数が存在すると仮定し，それを g とする．このとき

$$h(z, x_2, \ldots, x_n) \;:=\; g(z, z, x_2, \ldots, x_n) + 1 \tag{3.3}$$

と定義すれば，h は n 変数の全域帰納的関数である．従って h のゲーデル数 e_h を用いて

$$h(z, x_2, \ldots, x_n) \;=\; univ_n(e_h, z, x_2, \ldots, x_n)$$

と書ける．h が全域関数なので，$univ_n(e_h, z, x_2, \ldots, x_n)$ もすべての $(z, x_2, \ldots, x_n) \in \mathbb{N}^n$ について定義されている．g は $univ_n$ の拡張であるから

$$h(z, x_2, \ldots, x_n) \;=\; univ_n(e_h, z, x_2, \ldots, x_n) \;=\; g(e_h, z, x_2, \ldots, x_n)$$

この等式に $z = e_h$ を代入すれば,

$$h(e_h, x_2, \ldots, x_n) \;=\; g(e_h, e_h, x_2, \ldots, x_n).$$

しかし h の定義 (3.3) によれば

$$h(e_h, x_2, \ldots, x_n) \;=\; g(e_h, e_h, x_2, \ldots, x_n) + 1,$$

[注7] この定理や後述の定理 3.15 のような計算機科学に見られる不可能性の証明は対象自身に言及する命題から矛盾を導くある種の対角線論法がよく使われる．以下の証明では，(3.3) で定義された関数 h の第一変数に h 自身のゲーデル数 e_h を代入するところが自己言及的である．

これは矛盾である. □

定理 3.7 の応用として次の停止問題を考えよう. $univ_n : \mathbb{N} \times \mathbb{N}^n \to \mathbb{N}$ を定理 3.6 の枚挙関数として, 関数 $halt_n : \mathbb{N} \times \mathbb{N}^n \to \{0, 1\}$ を

$$halt_n(z, x_1, \ldots, x_n) := \begin{cases} 1 & (univ_n(z, x_1, \ldots, x_n) \text{ が定義されている}) \\ 0 & (univ_n(z, x_1, \ldots, x_n) \text{ が定義されていない}) \end{cases}$$

と定義する. $halt_n$ は $univ_n$ の定義域の定義関数であり, 全域で定義された関数である. 実際には, z がゲーデル数の場合に, z に対応する部分帰納的関数が入力 $x_1, \ldots, x_n$ に対して定義されているかどうかを判定する関数である. 実際のコンピュータでいうと, 定義されていないということは無限ループに陥ることを意味するので, このような問いかけをプログラムの**停止問題**という. これに関して次の重要な定理がある.

定理 3.8 ([57]) $halt_n$ は全域帰納的関数ではない.

証明. 次の関数 $g : \mathbb{N} \times \mathbb{N}^n \to \mathbb{N}$ を考える;

$$g(z, x_1, x_2, \ldots, x_n) := \begin{cases} univ_n(z, x_1, x_2, \ldots, x_n) & (halt_n(z, x_1, \ldots, x_n) = 1) \\ 0 & (halt_n(z, x_1, \ldots, x_n) = 0) \end{cases}$$

もし $halt_n$ が全域帰納的関数であれば, この g も全域帰納的関数である. しかし g は枚挙関数 $univ_n$ を拡張した全域関数なので定理 3.7 よりそれは不可能である. □

定理 3.8 は $halt_n$ を計算するプログラムが存在しない, すなわち, 任意にプログラムとその入力が与えられたとき, それが無限ループに陥ってしまうかどうか, を判定するプログラムが存在しない, ということを意味する. 卑近な言葉でいえば, 任意に与えられたプログラムの不具合を検出するプログラムは存在しない, ということである.

—— ◇ —— ◇ ——

部分帰納的関数に関する理論を厳密に検証していくのはかなり大変な作業となるが, コンピュータの仕組みについて多少の知識があれば, それが正しいに違いないと確信することはそれほど困難ではない.

たとえば, クリーネの標準形は実際にコンピュータの設計に取り入れられている. そこの説明で新たに導入した変数 u は中央演算処理装置 (CPU) のプログラムカウンタと呼ばれるものに対応し, コンピュータがメモリ上に格納されたプログラムのどのアドレスを実行中かを指し示すのに使われる. 関数 $\langle x_1, \ldots, x_n \rangle$ は, じつはコマンドラインや関数に複数のパラメータを渡すときの方法を数学的に表したものである. たとえば, n 変数の関数 $f(x_1, \ldots, x_n)$ の表記でも分かるように, 区切り記号 "," によってパラメータを区別して, 一括して文字列, たとえば "1.5, 20.0, –2.1" を関数 $f(\cdot)$ に入力する. ここで入力データをすべて有限 $\{0, 1\}$-列, 従って自然数と思えば,

“1.5, 20.0, −2.1” は一つの自然数と思うことができ，それは三つのデータを結合したものである．ここで区切り記号 “,” は関数 $\langle x_1, x_2\rangle$ の定義式 (3.2) における “0, 1” に相当する．部分帰納的関数の枚挙関数は，さまざまなプログラム (= ゲーデル数に相当) をインストールして様々な用途に使われる汎用コンピュータの数学的モデルに他ならない．

停止問題については，たとえば数論の知識があると納得しやすい．いま，x 以上の偶数で二つの素数の和にならないものを探索するプログラム $f(x)$ を考える．f の入力は x で，もしそのような偶数が見つかったら，それを出力し，停止する．もし $halt_1$ が全域帰納的関数だったならば，それは $f(4)$ が停止するかどうかも判定する ($halt_1(e_f, 4)$ を計算する)．このことはよく知られた未解決のゴールトバッハ (Goldbach) の問題を解くことを意味する．この例のみならず，$halt_1$ は数論の未解決問題をいくらでも解くことができるだろう．そんなことはとても考えられないではないか．

3.2 コルモゴロフ複雑度と乱数

定義 3.9 $p \in \{0,1\}^*$ に対して，$p \in \{0,1\}^n$ となる n を $L(p) \in \mathbb{N}$ と書く (p の長さ)．$p \in \mathbb{N}$ とするときは p に対応する $\{0,1\}$-列の長さとする．たとえば $L(5) = L((1,0)) = 2$．一般に $L(p) = \lfloor \log_2(p+1) \rfloor$．

定義 3.10 (アルゴリズムに依存した計算の複雑度) $A : \{0,1\}^* \times \{0,1\}^* \to \{0,1\}^*$ は $\mathbb{N} \times \mathbb{N} \to \mathbb{N}$ と見て部分帰納的関数とする．A をアルゴリズムと呼ぶ．アルゴリズム A のもとで $y \in \{0,1\}^*$ を入力としたときの $x \in \{0,1\}^*$ の計算の複雑度を

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

と定義する．ただし，$A(p,y) = x$ となる p が存在しない場合は $K_A(x|y) := \infty$ と定義する．

定義 3.10 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 y から x を計算して出力する．従って，$K_A(x|y)$ はプログラム言語 A の下で入力 y から出力 x 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，このままでは普遍的な計算の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 3.11 ある部分帰納的関数 $A_0 : \{0,1\}^* \times \{0,1\}^* \to \{0,1\}^*$ が存在して，任意の部分帰納的関数 $A : \{0,1\}^* \times \{0,1\}^* \to \{0,1\}^*$ に対して以下を満たす定数 $c_{A_0A} \in \mathbb{N}$ が存在する;

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

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 algorithm)) と呼ばれる．

証明. 枚挙関数 $univ_2$ を用いてアルゴリズム A_0 を次のように定義する.

$$A_0(z,y) := univ_2((z)_1^2,(z)_2^2,y), \qquad z,y\in\{0,1\}^*.$$

もし $z=\langle e,p\rangle$ の形をしていなかったら $A_0(z,y)$ は定義しない. $A_0(\langle e_A,p\rangle,y)=A(p,y)$ であるから (3.2) より

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

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

万能アルゴリズムは一意的ではない. しかし, A_0 と A_0' を二つの万能アルゴリズムとするとき, $c>0$ が存在して

$$\forall x,\ y\in\{0,1\}^*,\quad \left|K_{A_0}(x|y)-K_{A_0'}(x|y)\right|\ <\ c, \tag{3.4}$$

が成り立つので, c に比べて $K_{A_0}(x|y)$, $K_{A_0'}(x|y)$ が大きいときは, この二つはほとんど同じとみなすことができる.

定義 3.12 万能アルゴリズム A_0 を一つ固定し

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

を y を与えたときの x の計算の複雑度と呼ぶ. とくに y が空語のとき, $K(x)$ と書いて x の**コルモゴロフ複雑度** (Kolmogorov complexity) と呼ぶ.[注8]

$K(x|y)$ と $K(x)$ はともに有限な値をとる全域関数である.

定理 3.13 (i) ある定数 $c>0$ が存在して, 任意の $x\in\{0,1\}^n$, $y\in\{0,1\}^*$ に対して $K(x|y)\le n+c$.
(ii) $n>c'>0$ のとき, 任意の $y\in\{0,1\}^*$ に対して $\#\{x\in\{0,1\}^n\mid K(x|y)\ge n-c'\}>2^n-2^{n-c'}$.

証明. (i) アルゴリズム $A(x,y):=p_1^2(x,y)=x$ に対して $x\in\{0,1\}^n$ ならば $K_A(x|y)=n$ だから定理 3.11 より $K(x|y)\le n+c$. (ii) $L(p)<n-c'$ なる $p\in\{0,1\}^*$ の個数は $2^0+2^1+\cdots+2^{n-c'-1}=2^{n-c'}-1$ 個だから, $K(x|y)<n-c'$ となる $x\in\{0,1\}^n$ の個数は高々 $2^{n-c'}-1$ 個である. これより (ii) の主張が従う.[注9] □

$K(x)$ は $K(x|y)$ の特別な場合だから定理 3.13 の主張 (i)(ii) は $K(x)$ についても成り立つ. 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 $x\in\{0,1\}^n$ のコルモゴロフ複雑度 $K(x)$ はほぼ n である. そのように $K(x)$ がほぼ最大の n となる $x\in\{0,1\}^n$ を**乱数** (random number) と呼ぶ.[注10]

注8 コルモゴロフの複雑さ, コルモゴロフ記述量, コルモゴロフ-チャイティンの複雑性, などいろいろな呼び名がある.
注9 主張 (i)(ii) は § 2.3 で述べた個数の議論そのものである.
注10 $K(x)$ は一意的には決まらず, (3.4) のように定数の曖昧さを伴うので, 乱数の定義もこのように曖昧さを残したものにならざるを得ない.

例 3.14 円周率πの公式計算記録は，2009年8月現在，2進小数で8兆5605億4349万桁 (10進小数で2兆5769億8037万桁) である．それを算出したプログラムは，8兆5605億4349万ビットよりずっと短いので，πの2進8兆5605億4349万桁小数表示における0と1の並びは乱数ではない．

例 3.14 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 3.15 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を実際に判定する手続きは存在しないことを主張する．

定理 3.15 各 $y \in \{0,1\}^*$ に対して，$K(x|y)$ は x の関数として全域帰納的関数でない．とくに $K(x)$ は全域帰納的関数でない．

証明. $\{0,1\}^*$ を $\mathbb{N}$ と同一視して論じる．$y \in \{0,1\}^*$ を一つ固定する．背理法で示す．$K(x|y)$ が x の関数として全域帰納的関数であると仮定する．この仮定により関数 $\psi(x) := \min\{z \in \mathbb{N} \mid K(z|y) \geq x\}$，$x \in \mathbb{N}$，は全域帰納的関数である．定義より $x \leq K(\psi(x)|y)$．アルゴリズム A を $A(p,y) := \psi(\langle p,y\rangle)$ と定めれば

$$K_A(\psi(x)|y) \;=\; \min\{L(p) \mid p \in \mathbb{N},\, \psi(\langle p,y\rangle) = \psi(x)\}$$

だから，任意の $x = \langle p,y\rangle$ の形をした無限個の x に対して $K_A(\psi(x)|y) \leq L(p) \leq L(x)$．従って定理 3.11 より，ある $c > 0$ が存在して，そのような無限個の x に対して

$$x \leq K(\psi(x)|y) \leq L(x) + c \tag{3.5}$$

が成り立つ．しかし $L(x) = \lfloor \log_2(x+1) \rfloor$ なので (3.5) は十分大きい x に対しては不可能な不等式である． □

定理 3.15 が停止問題の計算不可能性 (定理 3.8) と深く関係することを説明しよう．次のプログラム *complexity* は，一見，$K(x|y)$ を計算するプログラムに見える．

```
function complexity(x, y : {0,1}*) : integer;
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 z ∈ {0,1}^l に関する繰り返し，
      もし A_0(z, y) = x ならば l を出力して停止する．
end;
```

ここに A_0 は K を定義する万能アルゴリズムである．*complexity* は短い順にすべての {0,1}-列 z (プログラム) を考えて，入力が y のとき出力が x になっているかどうかを確かめる．しかしそれを実際に実行したときに停止するとは限らない．じつは，ある {0,1}-列 x に対しては $K(x|y)$ を出力する前に *complexity* は無限ループに陥ってしまう．定理 3.8 により我々にはこのことを事前に察知しそれを防ぐ手立ては存在しない．

定理 3.16　ある原始帰納的関数 $K'(t,x,y)$ が存在して，(i) 各 $t,x,y \in \mathbb{N}$ に対して $K'(t,x,y) \geq K(x|y)$，(ii) 各 $x,\, y \in \mathbb{N}$ に対して $K'(t,x,y)$ は $t \to \infty$ のとき単調減少で $K(x|y)$ に収束する．

証明. $c > 0$ を $K(x|y) = K_{A_0}(x|y) < L(x) + c$ を満たす定数，万能アルゴリズム A_0 のクリーネの標準形を

$$A_0(p,y) = g(p,y,\mu_z(q(p,y,z))), \quad g,q : \text{原始帰納的関数},$$

とする．このとき

$$\begin{aligned} \mu_{z<t}(q(p,y,z)) &:= \min(\{z < t \mid q(p,y,z) = 0\} \cup \{t\}) \\ g'(t,x,p,y,z) &:= \begin{cases} g(p,y,z) & (z < t) \\ x+1 & (z \geq t) \end{cases} \\ A(t,x,p,y) &:= g'(t,x,p,y,\mu_{z<t}(q(p,y,z))) \end{aligned}$$

と定義する．このとき $\mu_{z<t}(q(p,y,z))$ は変数 (t,p,y) に関して原始帰納的関数[注11]なので $A(t,x,p,y)$ も原始帰納的関数である．そこで

$$K'(t,x,y) := \min(\{\, L(p) \mid L(p) < L(x) + c,\ A(t,x,p,y) = x \,\} \cup \{L(x) + c\})$$

とすれば，これが求める関数である．　□

3.3　検定とマルティン=レーフの定理

数理統計学では個々の $\{0,1\}$-列がランダムかどうか (硬貨投げの確率過程のサンプルに見えるかどうか) は検定によって判定される．しかしながら，検定の全体はそのままでは無矛盾でない．

例 3.17　(検定の矛盾性) $\{0,1\}^n$ の元に対して，以下のような $n-9$ 個の棄却域

$$R^{(j)} := \{\, (x_1,\ldots,x_n) \in \{0,1\}^n \mid (x_j, x_{j+1}, \ldots, x_{j+9}) \neq (0,\ldots,0) \,\}, \quad j = 1,\ldots,n-9,$$

に対応する検定 $U^{(1)},\ldots,U^{(n-9)}$ を考える．各検定 $U^{(j)}$ の危険率は 2^{-10} である．いま $x \in \{0,1\}^n$ はすべての $j = 1,\ldots,n-9$ に対して $x \notin R^{(j)}$，すなわちこれらのすべての検定に採択されたとしよう．そのとき，次の棄却域

$$R := \{\, (x_1,\ldots,x_n) \in \{0,1\}^n \mid 1 \leq \exists\, j \leq n-9, \quad (x_j, x_{j+1}, \ldots, x_{j+9}) = (0,\ldots,0) \,\}$$

に対する検定 U は，危険率が n を大きくするといくらでも小さくなるにもかかわらず，x を棄却する．すなわち任意の x はいずれかの検定 $U^{(1)},\ldots,U^{(n-9)}$ あるいは U で棄却される．

こうした検定の矛盾性を解決するために，マルティン=レーフ (Martin-Löf) は各検定の危険率を調整して，一つのある普遍性を持つ検定 (万能検定，定理 3.19) を構成し，x が前節で述べた意味で乱数であることと，x がその万能検定に採択されることが同等であることを示した (定理 3.20).

[注11] $\mu_{z<t}(q(p,y,z))$ は有界 μ-作用素と呼ばれる．

3.3.1 検定の定式化と万能検定

まず，検定を一般的に定義する．

定義 3.18 $\mathbb{N}\times\{0,1\}^*\supset U$ が**検定** (test)[注12]とは，
(i) U は帰納的可算集合，
(ii) $U_m:=\{x\in\{0,1\}^*\,|\,(m,x)\in U\}$ とするとき，$U_m\supset U_{m+1}$，$m\in\mathbb{N}$，
(iii) $\#(U_m\cap\{0,1\}^n)\le 2^{n-m}$，$n>m>0$，
(iv) $(0,0\,(=\text{空語}))\in U$，[注13]
が成り立つことと定義する．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

$$m_U(x):=\max\{m\in\mathbb{N}\,|\,x\in U_m\} \tag{3.6}$$

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 3.19 ([31]) 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

$$\forall m\in\mathbb{N},\quad U_{m+c}\subset V_m. \tag{3.7}$$

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．

証明. 最初に証明のアイデアを述べよう．$\{U_e\}_{e\in\mathbb{N}}$ をすべての検定を何らかの方法で列挙したものとする．各 $m\in\mathbb{N}$ に対して，集合 $V_m\subset\{0,1\}^*$ を

$$V_m:=\bigcup_{e\in\mathbb{N}}(U_e)_{m+e+1},\quad \text{ただし } (U_e)_{m+e+1}:=\{x\,|\,(m+e+1,x)\in U_e\},$$

と定義する．このとき，$V_{m+1}\subset V_m$ は明らかであり，任意の検定 U に対して，ある e が存在し，$U=U_e$ かつ $U_{m+e+1}=(U_e)_{m+e+1}\subset V_m$ が成り立つ．さらに

$$\#(V_m\cap\{0,1\}^n)\le\sum_{e\in\mathbb{N}}\#((U_e)_{m+e+1}\cap\{0,1\}^n)\le\sum_{0\le e\le n-m-1}2^{n-m-e-1}=2^{n-m}-1.$$

そこで

$$V:=\bigcup_{m\in\mathbb{N}}\{(m,x)\,|\,x\in V_m\}$$

とすれば，これが帰納的可算ならば万能検定である．

[注12] ここでいう検定は [16, 21, 41] などに見られる検定の一般化，形式化であって，実用的な意味合いは薄い．
[注13] 条件 (iv) は本質的でなく削除しても実質的に変わりはない．この条件は定理 3.19 の証明を簡単にするために有用である．

このアイデアを実現しよう．

$$univ_1(e,k) \;=\; g(e,k,\mu_z(q(e,k,z))), \quad e,k\in\mathbb{N},$$

を枚挙関数 $univ_1$ のクリーネの標準形とする．ここに g と q は原始帰納的関数である．このとき原始帰納的関数 $\psi:\mathbb{N}^3\to\mathbb{N}$ を

$$\psi(e,t,k) \;:=\; g(e,k,\mu_{z<t}(q(e,k,z)))$$

で定める．次にやはり原始帰納的関数 $\psi':\mathbb{N}^3\to\mathbb{N}\times\{0,1\}^*$ を以下のように定義する; 各 $e,t,k\in\mathbb{N}$ に対して，$y:=\psi(e,t,k)$ とし，

$$\psi'(e,t,k) \;:=\; \begin{cases} ((y)_1^2,(y)_2^2) & (\mu_{z<t}(q(e,k,z))<t \text{ かつ } y\in G^2(\mathbb{N}^2)),\\ (0,0) & (\text{そうでないとき}), \end{cases}$$

と定義する．さて，各 $e,t\in\mathbb{N}$ に対し，

$$\widetilde{U}_{e,t} \;:=\; \{(m',x)\,|\,m'\le m,\ \text{ただし } \psi'(e,t,k)=:(m,x),\ k<t\}\;\subset\;\mathbb{N}\times\{0,1\}^*,$$

とし，さらに

$$\begin{aligned} U_{e,0} \;&:=\; \{(0,0)\},\\ U_{e,t+1} \;&:=\; \begin{cases} U_{e,0}\cup\widetilde{U}_{e,t+1} & (U_{e,0}\cup\widetilde{U}_{e,t+1}\text{ は検定}),\\ U_{e,t} & (\text{そうでないとき}), \end{cases} \qquad t\in\mathbb{N}, \end{aligned}$$

とする．容易に分かるように，$\{U_{e,t}\}_{t\in\mathbb{N}}$ は検定の増大列であり，極限 $U_e:=\bigcup_{t\in\mathbb{N}}U_{e,t}$ もまた検定である．一方，任意の検定 U に対して，ある e が存在して $U=U_e$ である．

いま

$$V_m \;:=\; \bigcup_{e\in\mathbb{N}}(U_e)_{m+e+1} \;=\; \bigcup_{t\in\mathbb{N}}\bigcup_{e\in\mathbb{N}}(U_{e,t})_{m+e+1}, \quad m\in\mathbb{N},$$

そして

$$\begin{aligned} V \;&:=\; \bigcup_{m\in\mathbb{N}}\{(m,x)\,|\,x\in V_m\} \;=\; \bigcup_{m\in\mathbb{N}}\bigcup_{t\in\mathbb{N}}\bigcup_{e\in\mathbb{N}}\{(m,x)\,|\,x\in(U_{e,t})_{m+e+1}\}\\ &=\; \bigcup_{m\in\mathbb{N}}\bigcup_{t\in\mathbb{N}}\bigcup_{e\in\mathbb{N}}\{(m,x)\,|\,(m+e+1,x)\in U_{e,t}\}, \end{aligned}$$

とすれば，V は帰納的可算集合なら万能検定である．では V を枚挙する部分帰納的関数 $f:\mathbb{N}\to\mathbb{N}\times\{0,1\}^*$ を構成しよう．まず $n\in\mathbb{N}$ とする．もし $n\in G^4(\mathbb{N}^4)$ ならば $m:=(n)_1^4$, $e:=(n)_2^4$, $t:=(n)_3^4$, $x:=(n)_4^4$, とおく．ψ' を用いて $\{\widetilde{U}_{e,s}\}_{s\le t}$ の各集合を，従って $U_{e,t}$ も枚挙することができる．それで $(m+e+1,x)\in U_{e,t}$ かどうか判定することができる．もし，そうなら $f(n):=(m,x)$ と定義する．このように定義された f は部分帰納的関数であり，[注14] V を枚挙する．従って V は帰納的可算集合である．□

[注14] この f のような枚挙のアルゴリズムはしばしば *dovetailing* と呼ばれる．

(3.7) を (3.6) を使って書けば,

$$m_U(x) \le m_V(x) + c \tag{3.8}$$

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 $c > 0$ が存在して

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

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．

3.3.2 マルティン=レーフの定理

定理 3.20 (マルティン=レーフの定理，[31]) ある定数 $c > 0$ が存在して任意の $x \in \{0,1\}^*$ に対して

$$|L(x) - K(x|L(x)) - m(x)| \le c. \tag{3.9}$$

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

$$U := \{(m,x) \,|\, K(x|L(x)) < L(x) - m\} \cup \{(0,0)\}$$

としたとき，もし U が検定であれば

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

だから (3.8) によって，$m(x) + c > L(x) - K(x|L(x))$ となる．

そこで U が検定であることを示そう．定義 3.18 の条件 (ii) が満たされるのは明らか. (iii) は定理 3.13(ii) により分かる．それで (i)，すなわち U が帰納的可算集合であることを示せばよい.[注15] 定理 3.16 の関数 K' を用いて $K''(t,x) := K'(t,x,L(x))$ とおく．すると $K''(t,x)$ は原始帰納的関数であり，$t \to \infty$ のとき，単調減少で $K(x|L(x))$ に収束する．とくに，各 x に対して，ある $t_x > 0$ が存在して $t \ge t_x$ ならば $K''(t,x) = K(x|L(x))$ となる．従って，原始帰納的関数

$$x \dot{-} y := \begin{cases} x - y & (x > y), \\ 0 & (x \le y), \end{cases}$$

を使って U は

$$U = \{(m,x) \,|\, \exists t \text{ s.t. } K''(t,x) \dot{-} (L(x) - m - 1) = 0\} \cup \{(0,0)\} \tag{3.10}$$

と表される．これは定理 3.5(iv) により帰納的可算集合である．

[注15] もし，$K(x|L(x))$ が全域帰納的関数であれば定理 3.5(v) によって U が帰納的可算集合であることが分かる (後述の (3.10) 式参照)．しかし，定理 3.15 と同様の論法によって $K(x|L(x))$ は帰納的でないことが分かるので，このあとの議論が必要となる．

第 2 段 $K(x|L(x)) \leq L(x) - m(x) + c$ の証明: 万能検定 V を枚挙する原始帰納的関数を ϕ とする (定理 3.5(ii)). すなわち, $\phi(\mathbb{N}) = V$. ここで ϕ は単射であるとしてよい. この ϕ を用いて, アルゴリズム $A : \{0,1\}^* \times \{0,1\}^* \to \{0,1\}^*$ を以下のように定義する. $\phi(i) =: (m_i, x_i) \in V$ 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

$$A(\underbrace{0\ldots01}_{L(x_2)-m_2}, L(x_2)) := x_2 \tag{3.11}$$

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. このとき明らかに

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

従って, $K(x|L(x)) \leq L(x) - m(x) + c$ が成り立つ. □

この定理は, $K(x|L(x))$ が $L(x)$ に近ければそれだけ x はランダムに見える, ということを巧みに述べた定理である. (3.9) 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ランダムに見える, ということを意味する.

3.4　コルモゴロフ複雑度とエントロピー

確率変数 X が離散的分布

$$\Pr(X = a_i) = p_i, \quad i = 1, 2, \ldots, M, \quad (a_i \neq a_j, \text{ if } i \neq j)$$

を持つとき

$$H(X) = -\sum_{i=1}^{M} p_i \log_2 p_i \geq 0$$

を X のエントロピーと呼ぶ. たとえば一般の ("公平でない" 場合も含めて) 硬貨投げの確率過程について考えてみよう. $0 < p < 1$ とし, $\{X_i^{(p)}\}_{i=1}^n$ を i.i.d. で共通の分布を

$\Pr(X_1^{(p)} = 1) = p$, $\Pr(X_1^{(p)} = 0) = q := 1 - p$, とする．このとき

$$\begin{aligned} H\left(\{X_i^{(p)}\}_{i=1}^n\right) &= -\sum_{x\in\{0,1\}^n} \Pr\left(\{X_i^{(p)}\}_{i=1}^n = x\right) \log_2 \Pr\left(\{X_i^{(p)}\}_{i=1}^n = x\right) \\ &= -\sum_{r=0}^{n} \Pr\left(S_n = r\right) \log_2 p^r q^{n-r}, \quad S_n := \sum_{i=1}^{n} X_i, \\ &= -\sum_{r=0}^{n} \Pr\left(S_n = r\right) \left(r \log_2 p + (n-r) \log_2 q\right) \\ &= -\mathbf{E}\left[S_n \log_2 p + (n - S_n) \log_2 q\right] \\ &= n(-p \log_2 p - q \log_2 q) \end{aligned}$$

である．[注16]最後の等式は $\mathbf{E}[S_n] = np$ より従う．以下では

$$h(p) := -p \log_2 p - q \log_2 q$$

と書くことにする．$\{X_i^{(1/2)}\}_{i=1}^n$ は“公平な”硬貨投げで $\{0,1\}^n$ 上一様に分布し，任意の p に対して

$$H\left(\{X_i^{(p)}\}_{i=1}^n\right) \le H\left(\{X_i^{(1/2)}\}_{i=1}^n\right) = nh(1/2) = n$$

が成り立つ．

荒っぽく言えば，$H(X)$ は確率変数 X を実現するために必要なおおよその硬貨投げの長さを表す (マクミラン (McMillan) の定理，cf. [37])．一方，コルモゴロフ複雑度 $K(x)$，$x \in \{0,1\}^*$，は x を生成するために最小限必要なプログラムの長さであった．コルモゴロフは両者の深い関係を見抜き，しばしば $K(x)$ を“x のエントロピー”とも呼んだ．次の補題とそれに続く定理はそれらの関係について明らかにする．

補題 3.21 (cf. [64]) $x = (x_1, x_2, \ldots, x_n) \in \{0,1\}^n$ は $\sum_{i=1}^n x_i = np$ であるとする．このとき，n にも x にもよらない定数 $c > 0$ が存在して

$$K(x) \le nh(p) + \frac{7}{2} \log_2 n + c$$

が成り立つ．

証明. 条件を満たすような $\{0,1\}^n$ の元の総数は $\binom{n}{np}$ 個であり，それらを標準的順序で並べたとき当該 x は n_1 番目であるとする．データ n，np，n_1 から x を生成するアルゴリズム (部分帰納的関数) が存在するので，それを A とする．すなわち

$$A(\langle n, np, n_1\rangle, 0) = x.$$

このとき

$$\begin{aligned} K_A(x) &= L(\langle n, np, n_1\rangle) \\ &= 2L(n) + 2 + 2L(np) + 2 + L(n_1) \\ &\le 2\lfloor \log_2(n+1)\rfloor + 2\lfloor \log_2(np+1)\rfloor + \left\lfloor \log_2\left(\binom{n}{np} + 1\right)\right\rfloor + 4. \end{aligned}$$

[注16]定常過程のエントロピーというとき，通常，ここでいう $H\left(\{X_i^{(p)}\}_{i=1}^n\right)$ を長さ n で割ったものをいう．

ここでスターリングの公式 $n! \sim n^n e^{-n}\sqrt{2\pi n}$ により

$$\binom{n}{np} = \frac{n!}{(np)!(nq)!} \sim \frac{1}{\sqrt{2\pi npq}} \cdot \frac{1}{p^{np}q^{nq}}, \quad n \gg 1,$$

つまり

$$\log_2 \binom{n}{np} \approx -\frac{1}{2}(\log_2 2\pi pq + \log_2 n) - np\log_2 p - nq\log_2 q, \quad n \gg 1.$$

以上から，ある $c' > 0$ が存在して

$$K_A(x) \le nh(p) + \frac{7}{2}\log_2 n + c'.$$

定理 3.11 より，あらたに $c > c'$ をとれば

$$K(x) \le nh(p) + \frac{7}{2}\log_2 n + c.$$

とできる．□

補題 3.21 の条件を満たす x は全部で $\binom{n}{np}$ 個あって，$n \gg 1$ のとき，それは 2 進数で表せば，およそ

$$nh(p) - \frac{1}{2}\log_2 n - \frac{1}{2}\log_2 2\pi pq$$

桁の数である．従って，定理 3.13(ii) と同様の議論によって，補題 3.21 の条件を満たす x のうち個数の上で少なくとも $1 - 2^{-c''}$，$c'' > 0$，の割合の x は

$$K(x) \ge nh(p) - \frac{1}{2}\log_2 n - \frac{1}{2}\log_2 2\pi pq - c'', \quad c'' > 0.$$

を満たすことが分かる．このことと補題 3.21 ならびに，一般の硬貨投げに関する大数の弱法則

$$\forall\, \varepsilon > 0, \quad \Pr\left(\left|\sum_{i=1}^{n} X_i^{(p)} - np\right| \ge n\varepsilon\right) \to 0, \quad n \to \infty,$$

から次のことが分かる．

定理 3.22　(cf. [64], Proposition 5.1) $0 < p < 1$ とする．任意の $\varepsilon > 0$ に対して

$$\lim_{n\to\infty} \Pr\left(\left|\frac{K\left(\{X_i^{(p)}\}_{i=1}^n\right)}{n} - h(p)\right| \ge \varepsilon\right) = 0$$

が成り立つ．

注意 3.23　$n \gg 1$ とする．コルモゴロフは $K(x) \approx n$ なる $x \in \{0,1\}^n$ を乱数と呼んだが，実際には $K(x) \approx cn$ $(0 < c < 1)$ であるような $x \in \{0,1\}^n$ でもとても人の意思で選ぶことなどできないので，十分にランダムであると考えてよい．それゆえ，公平でない硬貨投げの確率過程の一般的 (典型的) な見本もランダムであると言ってよいだろう．

3.5 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曖昧さがない．硬貨投げの確率過程に関する確率 0 の事象であって計算可能な手続きで規定されるもの (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.[注17]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**無限乱数列** (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，を正確に述べよう． 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 $P_\infty(A) = 0$ を満たす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

$$U_m \;=\; \bigcup_k C(y_k^{(m)}) \supset A \tag{3.12}$$

$$\sum_k P_\infty\left(C(y_k^{(m)})\right) \;=\; \sum_k 2^{-L(y_k^{(m)})} \;<\; 2^{-m} \tag{3.13}$$

が成り立つことである.

$$U \;=\; \{(m,x) \in \mathbb{N} \times \{0,1\}^* \,|\, C(x) \subset U_m\} \tag{3.14}$$

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.

$$(m,x) \in U, \quad n \le m \quad \text{かつ} \quad C(y) \subset C(x) \qquad \Longrightarrow \qquad (n,y) \in U \tag{3.15}$$

そこで，A が帰納的零集合であるとは，条件 (3.12)~(3.15) 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** (sequential 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 V が存在する: 任意の列検定 U に対して

$$\forall m \in \mathbb{N}^+, \quad U_{m+c} \subset V_m \tag{3.16}$$

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 V を**万能列検定** (universal sequence test) という ([31])．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 3.24 円周率 π の小数部分を 2 進小数で表して得られる無限 $\{0,1\}$-列 $\{d_i(\pi-3)\}_{i=1}^\infty$ 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 n を指定したとき，$\{d_i(\pi-3)\}_{i=1}^n$ 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 $\{d_i(\pi-3)\}_{i=1}^\infty$ だけからなる 1 点集合が帰納的零集合であることを示すことができる.

[注17]そもそも，計算可能な手続き全体が可算集合であるから.

注意 3.25　公平でない硬貨投げの確率過程の分布を含むもっと一般の (計算可能な) 確率測度に基づく無限乱数列の概念について [31] が扱っている.

注意 3.26　各無限 $\{0,1\}$-列に対して，それがランダムであるかどうかをコルモゴロフ複雑度によって特徴づけることはできない．しかし，その定義を少し変更すれば，そのランダム性を複雑度の言葉で特徴づけることができる．詳しくは [28] を見よ.

3.6　乱数と確率論

コルモゴロフによる現代確率論はランダムな変量を確率変数という形式で表現するが，これは適当な確率空間 — 第 1 章で述べたことを踏まえて硬貨投げの確率空間 $(\{0,1\}^L, 2^{\{0,1\}^L}, P_L)$ としよう — で定義された関数 $X : \{0,1\}^L \to \mathbb{R}$ のことであり，それ自体はランダム性と無縁の概念である．X をランダムな変量と考える際には，$\omega \in \{0,1\}^L$ が P_L に従ってランダムに選ばれ，その結果として $X(\omega)$ がランダムな値をとる，と解釈する．しかし，確率論では確率変数を常に関数として扱うので，$\omega \in \{0,1\}^L$ が選ばれる過程は関知しない．ランダム性は ω が選ばれる過程にこそ存在するので，その意味で確率論は「ランダムとは何か」という問題を避けながら成立していると言えるだろう.

コルモゴロフは，晩年，パーキンソン氏病という難病を患いながらも，このように自身の確率論が避けてきた「ランダムとは何か」という問題に没頭したという ([54] p.115)．そしてついに彼は，この章で述べたように，乱数という概念を数学的に定式化しこの難問を解いた．乱数を認識するということは，確かに理論的には確率論にとって不要であるものの，それは確率論そのものの深い理解のために非常に有益である.[注18] このことを以下で説明しよう.

コルモゴロフによればランダム性とはすなわち乱数の性質である．乱数の存在は標本空間 $\{0,1\}^L$ が巨大な集合である場合に限り意味を持つ．だから，まず，$L \gg 1$ の場合を調べることがランダム性の研究には重要であることが認識されよう．それで以下，$L \gg 1$ とする．このとき $\{0,1\}^L$ の元のうち大多数を占める乱数を我々は自らの意志では，事実上，選ぶことはできないから，人為的には一様確率測度 P_L に従って ω を選ぶことは不可能である．従って確率測度として P_L を設定するということは，$\{0,1\}^L$ の元が人為的でない方法 — 無作為な方法 — で選ばれることを前提としているわけで，それゆえ，確率空間 $(\{0,1\}^L, 2^{\{0,1\}^L}, P_L)$ はランダムな現象を解明するための枠組みとなり得るのである.

また定理 3.15 からも想像できるように，一般に個々の乱数について何らかの性質を知ることは困難である．しかし，乱数は $\{0,1\}^L$ の元のうち大多数を占めるのだから，乱数の性質を知りたければ $\{0,1\}^L$ の元のうち大多数が持つ性質を調べるのが手っ取り早い．それはすでに確率論で詳しく調べられている．大数の法則，中心極限定理をはじめとする様々な極限定理が記述している性質がそうである．恐らく，極限

[注18] それゆえ数学辞典第 3 版 [41] では存在した “Kolmogorov-Chaitin の複雑性 (コルモゴロフ複雑度)” という用語が第 4 版では削除されたことを大変残念に思う.

定理こそがランダム性について具体的に調べることのできるほとんど唯一の数学的表現形式であろう．極限定理が確率論の中心的研究対象と言われる所以である．[注19]

真に驚くべきことは，乱数の概念が発見されるはるか以前から，慧眼の確率論研究者たちが極限定理の重要性を認識し，その研究に努めてきたことである．

[注19]たとえば教科書 [10] の「まえがき」と第 1 章「確率論を学ぶにあたって」の 1 ページ目に，確率論の真の主題は極限定理である，と明言されている．

第4章　疑似乱数生成器

4.1　計算量的に安全な疑似乱数生成器

4.1.1　定義

計算量的に安全な疑似乱数生成器の定義を述べよう．基本的なアイデアは [2, 58] に見ることができる．詳しくは [29, 40] などを見よ．

定義 4.1

1. この章で現れる関数は部分帰納的関数 $f : \{0,1\}^* \to \{0,1\}^*$ で主として次の形をしている; 各 $n \in \mathbb{N}^+$ に対して $f_n := f|_{\{0,1\}^{r(n)}} : \{0,1\}^{r(n)} \to \{0,1\}^{s(n)}$．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．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．

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. $Y \in_U B$ は Y が集合 B 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 4.2　$g_n : \{0,1\}^n \to \{0,1\}^{\ell(n)}$，$\ell(n) > n$，なる多項式時間関数 $g = \{g_n\}_n$ を**疑似乱数生成器** (pseudorandom generator) という．

注意 4.3　第 2 章の定義 2.5 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 4.2 にあるように，関数の列として定義するのがよい．それは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

$$\delta_{g,A}(n) := \left|\Pr_{Z_{\ell(n)}}(A_n(Z_{\ell(n)}) = 1) - \Pr_{Z_n}(A_n(g_n(Z_n)) = 1)\right| \tag{4.1}$$

とおく.[注1]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．そこで

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

とおく.

定義 4.4　すべての A に対して $S_{g,A}(n)$ が多項式パラメータでないとき (あらゆる多項式より早く増大するとき)，疑似乱数生成器 g は **計算量的に安全**であるという.[注2]

$T_A(n)$ が多項式パラメータでないとき $S_{g,A}(n)$ も必ず多項式パラメータでないので，定義 4.4 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,「多項式時間で計算可能な関数で，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を生成するもの」を意味する.

計算量理論では，多項式時間関数を「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」というように捉えている.[注3] だから，計算量的に安全な疑似乱数生成器は「実際に計算可能な関数で，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を生成するもの」と捉えられている.

[注1] 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，(4.1) における確率の計算にその Y に関する確率も考慮する.

[注2] 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.

[注3] たとえば，$e^{t/1000}$ は多項式パラメータでないが，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.

4.1.2 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. [29, 40]).

疑似乱数の定義 4.4 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観点から生れた概念であるが，じつはモンテカルロ法においても有用である．このことを説明しよう．

まず，モンテカルロ法の対象となる実数値確率変数 S は $Z_{\ell(n)} \in_U \{0,1\}^{\ell(n)}$ の関数であるとする; $S := S(Z_{\ell(n)})$．ここで S 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，S の代わりに $S' := S(g_n(Z_n))$ を数値計算に用いるわけである．そこで問題は S' が果たしてうまく S の代役を無事に務めることができるかどうか，である．それは S と S' 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 $F_S(t) := \Pr_{Z_{\ell(n)}}(S \le t)$，$F_{S'}(t) := \Pr_{Z_n}(S' \le t)$ 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，$F_S(t)$ と $F_{S'}(t)$ は十分近いことが保証される．実際，検定の関数 A_n を $A_n(x) := \mathbf{1}_{\{S(x)\le t\}}$，$x \in \{0,1\}^{\ell(n)}$，とおけば，$S$ 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

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

は十分小さくなければならない．

4.1.3 存在問題

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理論的に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の一つである．

疑似乱数生成器に関してじつは次の定理がある．

定理 4.5 $\mathbf{P} = \mathbf{NP}$ ならば計算量的に安全な疑似乱数生成器は存在しない．[注4]

証明. 疑似乱数生成器 $g = \{g_n\}_n$，$g_n : \{0,1\}^n \to \{0,1\}^{\ell(n)}$，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

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

とおく．$M = \{M_n\}_n$ は多項式時間関数である．

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

とすれば $L \in \mathbf{NP}$．そこで $\mathbf{P} = \mathbf{NP}$ ならば $L \in \mathbf{P}$，つまり，多項式時間関数 $A = \{A_n\}_n$，$A_n : \{0,1\}^{\ell(n)} \to \{0,1\}$ 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，$T_A(n)$ は多項式パラメータだから，$S_{g,A}(n) = T_A(n)/\delta_{g,A}(n)$ も多項式パラメータである．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． □

$\mathbf{P} \neq \mathbf{NP}$ 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，現時点で計算量的に安全であると考えられている疑似乱数生成器は有効である，と．

[注4] なお，$\mathbf{P} \neq \mathbf{NP}$ 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．

4.1.4 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性質に注目する．

定義 4.6 $g = \{g_n\}_n$，$g_n : \{0,1\}^n \to \{0,1\}^{\ell(n)}$ 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 $g_n(Z)$ の第 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 ビットに続けて残りのビットを 0 で埋めたものを表す．すなわち

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

このとき，任意の $\tilde{A}$ に対して

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

が多項式パラメータでないとき，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 4.7 疑似乱数生成器 $g = \{g_n\}_n$ 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. 第 1 段 g は次ビット予測不可能でないとする．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

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

と定める．このとき

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

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．

第 2 段 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

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

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，

$$\begin{aligned}
&\Pr\left(\tilde{A}_n(i, X_{\{1,\dots,i-1\}}) = X_i\right) - \frac{1}{2} \\
&\quad = \Pr\left(X_i = Y_i,\, A_n(X_1,\dots,X_{i-1},Y_i,\dots,Y_{\ell(n)\}}) = 1\right) \\
&\qquad\quad +\Pr\left(X_i = W,\, A_n(X_1,\dots,X_{i-1},Y_i,\dots,Y_{\ell(n)}) = 0\right) - \frac{1}{2} \\
&\quad = \Pr\left(X_i = Y_i,\, A_n(X_1,\dots,X_i,Y_{i+1},\dots,Y_{\ell(n)}) = 1\right) \\
&\qquad\quad +\frac{1}{2}\Pr\left(A_n(X_1,\dots,X_{i-1},Y_i,\dots,Y_{\ell(n)}) = 0\right) - \frac{1}{2} \\
&\quad = \frac{1}{2}\Pr\left(A_n(X_1,\dots,X_i,Y_{i+1},\dots,Y_{\ell(n)}) = 1\right) \\
&\qquad\quad +\frac{1}{2}\left(1 - \Pr\left(A_n(X_1,\dots,X_{i-1},Y_i,\dots,Y_{\ell(n)}) = 1\right)\right) - \frac{1}{2} \\
&\quad = \frac{1}{2}\Pr\left(A_n(X_1,\dots,X_i,Y_{i+1},\dots,Y_{\ell(n)}) = 1\right) \\
&\qquad\quad -\frac{1}{2}\Pr\left(A_n(X_1,\dots,X_{i-1},Y_i,\dots,Y_{\ell(n)}) = 1\right)
\end{aligned}$$

従って

$$\begin{aligned}
\tilde{\delta}_{g,\tilde{A}}(n) &= \frac{1}{\ell(n)}\sum_{i=1}^{\ell(n)}\left(\Pr\left(\tilde{A}_n(i, X_{\{1,\dots,i-1\}}) = X_i\right) - \frac{1}{2}\right) \\
&= \frac{1}{2}\cdot\frac{1}{\ell(n)}\sum_{i=1}^{\ell(n)}\left(\Pr\left(A_n(X_1,\dots,X_i,Y_{i+1},\dots,Y_{\ell(n)}) = 1\right)\right. \\
&\qquad\qquad \left. -\Pr\left(A_n(X_1,\dots,X_{i-1},Y_i,\dots,Y_{\ell(n)}) = 1\right)\right) \\
&= \frac{1}{2\ell(n)}\left(\Pr(A_n(X) = 1) - \Pr(A_n(Y) = 1)\right)
\end{aligned}$$

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生成器 (cf. [11, 21, 32, 34]) の中には $(z_0,\dots,z_{n-1})$ を種として

$$z_i := f(z_{i-n},\dots,z_{i-1}), \quad i = n, n+1,\dots \tag{4.2}$$

のような漸化式で定義されているものが多い．これでは次の項が完全に予測できてしまう．従って定理 4.7 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生成器は対極をなしていることが分かる．

定理 4.7 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** ([2, 40]) を紹介しよう．[注5] p, q は $p = 3 (\mathrm{mod}\ 4)$，$q = 3 (\mathrm{mod}\ 4)$ を満たす素数でその値は非公開 (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

$$x_n := F(x_{n-1}) = x_{n-1}^2 \bmod N, \qquad n = 1, 2, \ldots, \tag{4.3}$$

$$y_n := G(x_n) = x_n \bmod 2 \tag{4.4}$$

とする．p, q が大きい素数のとき，逆関数 F^{-1} の計算は p, q を知っていれば易しいが，知らないと非常に手間が掛かる．そのため，種 x_0 を知らないで $\{y_n\}_{n=1}^m$ 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 $\{y_n\}_n$ が疑似乱数として提案されたわけである．

注意 4.8 (4.3) の関数 F のように，F 自身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．一般に，一方向関数 F の存在を仮定 (それは $\mathbf{P} \neq \mathbf{NP}$ より強い仮定) すれば，それに付随してハードコアビット関数 G(hard core bit function) — BBS 生成器の場合は (4.4) の関数 G — が存在し，(4.3)(4.4) の要領で定義される列 $\{y_n\}_n$ が次ビット予測不可能な疑似乱数となることが知られている ([29]).

—— ◇ —— ◇ ——

以上，述べてきた疑似乱数生成器の安全性は有限回の硬貨投げの関数であるような確率変数 $S(\omega)$ が対象である．しかし応用上はこれだけでは不十分で，一般に模倣可能 (§ 1.3) な確率変数 $S(\omega)$ について，その例外値を与える ω の集合に対して安全であるような疑似乱数生成器が必要である．これについてはまだ適切な数学的定式化ができていない．ただ，モンテカルロ積分に関して言えば，模倣可能な確率変数に対して適用可能な動的ランダム-ワイル-サンプリング (§ 5.4) がそれに当たるものと言えよう．

4.2 ワイル変換による疑似乱数生成器

確率論的考察から見出された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，種の大きさを大きくするに従っ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を示すことができる．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 4.11).

[注5] 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，§ 4.1.1 の定義を少し変更して考える必要がある．

4.2.1　定義

§ 4.2 では，次の定義 4.9 で定義されるルベーグ確率空間 $(\mathbb{T}^1, \mathcal{B}, \mathbb{P})$ 上の $\{0,1\}$-値確率過程の族 $\{Y_n^{(m)}(\bullet;\alpha)\}_{n=0}^{\infty}$，$\alpha \in \mathbb{T}^1$，$m \in \mathbb{N}^+$，に注目する．

定義 4.9　各 $\alpha \in \mathbb{T}^1$ と $m \in \mathbb{N}^+$ に対して

$$Y_n^{(m)}(x;\alpha) := \left(\sum_{i=1}^{m} d_i\,(x+n\alpha)\right) \bmod 2, \quad n = 0,1\ldots, \quad x \in \mathbb{T}^1. \tag{4.5}$$

(4.5) をコンピュータで実現するためには実数を有限 2 進小数で近似するとよい．

定理 4.10 ([63]) 各 $\alpha \in \mathbb{T}^1$ と $j, j_1, m \in \mathbb{N}^+$ に対して

$$\mathbb{P}\Big(0 \le \exists n \le 2^{j_1}-1 \text{ s.t. } Y_n^{(m)}(\bullet;\alpha) \neq Y_n^{(m)}(\lfloor\bullet\rfloor_{m+j}\,;\,\lfloor\alpha\rfloor_{m+j})\Big) < 2^{-(j-2j_1)}.$$

証明.

$$\begin{aligned}\Big|(x+n\alpha) - \big(\lfloor x\rfloor_{m+j} + n\lfloor\alpha\rfloor_{m+j}\big)\Big| &\le \big|x-\lfloor x\rfloor_{m+j}\big| + n\,\big|\alpha-\lfloor\alpha\rfloor_{m+j}\big| \\ &< 2^{-m-j} + n2^{-m-j} = (n+1)2^{-m-j}.\end{aligned}$$

これより

$$\mathbb{P}\Big(\lfloor\bullet+n\alpha\rfloor_m \neq \lfloor\lfloor\bullet\rfloor_{m+j} + n\lfloor\alpha\rfloor_{m+j}\rfloor_m\Big) \le \frac{(n+1)2^{-m-j}}{2^{-m}} = (n+1)2^{-j}.$$

従って

$$\begin{aligned}&\mathbb{P}\Big(0 \le \exists n \le 2^{j_1}-1 \text{ s.t. } Y_n^{(m)}(\bullet;\alpha) \neq Y_n^{(m)}(\lfloor\bullet\rfloor_{m+j}\,;\,\lfloor\alpha\rfloor_{m+j})\Big) \\ &\quad\le \sum_{n=0}^{2^{j_1}-1} \mathbb{P}\Big(\lfloor\bullet+n\alpha\rfloor_m \neq \lfloor\lfloor\bullet\rfloor_{m+j} + n\lfloor\alpha\rfloor_{m+j}\rfloor_m\Big) \\ &\quad< \sum_{n=0}^{2^{j_1}-1} (n+1)2^{-j} \\ &\quad= \frac{(2^{j_1}+1)2^{j_1}}{2}\cdot 2^{-j} < 2^{-(j-2j_1)}.\end{aligned}$$

□

定理 4.10 によれば，離散近似された確率過程 $\{Y_n^{(m)}(\lfloor\bullet\rfloor_{m+j}\,;\,\lfloor\alpha\rfloor_{m+j})\}_{n=0}^{2^{j_1}-1}$ は，j を大きくとることによって分布の意味で元の確率過程 (4.5) に任意に近くできる．[注6] この離散近似された確率過程を疑似乱数生成器

$$\{Y_n^{(m)}(\bullet;\lfloor\alpha\rfloor_{m+j})\}_{n=0}^{2^{j_1}-1} : D_{m+j} \cong \{0,1\}^{m+j} \to \{0,1\}^{2^{j_1}}$$

[注6] ここではエントロピー 0 の変換 $\mathbb{T}^1 \ni x \mapsto x+\alpha \in \mathbb{T}^1$ を用いている．もしカオス的な変換 (エントロピーが正) だと，離散化された確率過程の軌道で元の確率過程の軌道を長時間にわたり近似することが実際にはできなくなる．

と捉えよう.[注7]とくに α が無理数のとき，これを**ワイル変換による疑似乱数生成器**と呼ぶ.[注8]以下で見るように $m \to \infty$ における振る舞いが興味深い.

$\{Y_n^{(m)}(\bullet\,; \lfloor\alpha\rfloor_{m+j})\}_{n=0}^{2^{j_1}-1}$ のサンプルを生成するには，$\tilde{x} \in \{0,1\}^{m+j} \cong D_{m+j}$ を種として写像 $F_{m+j,\alpha} : D_{m+j} \to D_{m+j}$ と $G_m : D_{m+j} \to \{0,1\}$ を

$$F(\tilde{x}) \;=\; F_{m+j,\alpha}(\tilde{x}) \;:=\; \tilde{x} + \lfloor\alpha\rfloor_{m+j}, \tag{4.6}$$

$$G(\tilde{x}) \;=\; G_m(\tilde{x}) \;:=\; \left(\sum_{i=1}^{m} d_i(\tilde{x})\right) \bmod 2, \tag{4.7}$$

と定めて

$$Y_n^{(m)}(\tilde{x}\,; \lfloor\alpha\rfloor_{m+j}) = G(F^n(\tilde{x})), \quad n = 0, 1, \ldots, 2^{j_1} - 1, \tag{4.8}$$

とすればよい．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．$G(\tilde{x})$ は $\tilde{x}$ の上位 m ビットのパリティと呼ばれる量で，コンピュータで高速に計算することができる．この疑似乱数生成器の具体的な実装については § 6.2 または [51] を見よ.

4.2.2 次ビット予測の困難性

(4.6)(4.7) のように，ワイル変換による疑似乱数生成器 $\{Y_n^{(m)}(\bullet;\lfloor\alpha\rfloor_{m+j})\}_{n=0}^{2^j-1}$ も BBS 生成器 (4.3)(4.4) と同様の構造を持つ．後者においては関数 F の逆関数 F^{-1} の複雑さが次ビット予測が困難になると考えられるのに対し，前者では $m \gg 1$ のとき関数 $G = G_m$ の複雑さによって次ビット予測が困難になると考えられる．すなわち，前者においてパラメータ m を大きくすれば,[注9] 関数 G_m の複雑性が増大し，それに伴い次ビット予測がより困難になる．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証明できる.

極限定理を述べるために離散近似ではなく，元の確率過程 (4.5) について考える．$l \geq 2$，$0 \leq k_0 < \cdots < k_{l-1}$ 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．次の確率を考える;

$$F^{(m)}(k_0, \ldots, k_{l-1}; \alpha) := \mathbb{P}\left(\sum_{j=0}^{l-1} Y_{k_j}^{(m)}(\,\bullet\,;\alpha) = \text{奇数}\right). \tag{4.9}$$

あとの定理 4.13 で述べるとおり，この確率を計算する高速なアルゴリズムが存在する．それを利用して，次ビットを予測する関数 $\tilde{A} : \{0,1\}^{l-1} \to \{0,1\}$ を次のように定義する.

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

[注7] $j, j_1 \in \mathbb{N}^+$ は m に伴って大きくなるようにとるべきだが，記法が複雑になるのでここではこのように書いた.

[注8] 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.

[注9] m はおよそ種の大きさを表している.

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

$$\mathbb{P}\left(\tilde{A}(Y^{(m)}_{k_0}(\bullet;\alpha),\ldots,Y^{(m)}_{k_{l-2}}(\bullet;\alpha)) = Y^{(m)}_{k_{l-1}}(\bullet;\alpha)\right) = \left|F^{(m)}(k_0,k_1,\ldots,k_{l-1};\alpha)-\frac{1}{2}\right|+\frac{1}{2} \quad (4.10)$$

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

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

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

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

となって，いずれの場合も (4.10) が成り立つ．

関数 $\tilde{A}$ 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 (4.10) について，次の定理が成り立つ．

定理 4.11 $\mathbb{P}$ に関してほとんどすべての $\alpha\in\mathbb{T}^1$ について，任意の $l\geq 2$，$0\leq k_0<\ldots<k_{l-1}$，に対して，α に依存しないある $0<\rho<1$ が存在し，$m\to\infty$ のとき

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

この定理は $\tilde{A}$ 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，特殊な次ビット予測ではあるとはいえ，大変興味深い．

なお，$l=2$ のときには，任意の $\rho>\sqrt{(1+\sqrt{17})/8}=0.80024\ldots$ について定理 4.11 の主張が成り立つ (§ 4.3.5 定理 4.36)．

—— ◇ —— ◇ ——

定理 4.11，および後述の定理 4.36 と仮説 4.18 に関する実験結果 (表 4.1) を鑑みて，筆者は，ワイル変換による疑似乱数生成器はほとんどすべての無理数 α に対して計算量的に安全ではないか，と予想している．[注10]

4.2.3 有限次元分布の計算公式と従属性の消滅

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．

[注10] この予想が正しいとしても，そのような都合のよい無理数 α を具体的に構成することはできないであろうから，**P≠NP** が従うことにはならない．

補題 4.12 (i) $\epsilon_n \in \{0,1\}$, $n = 0,1,\dots,k-1$, として次の等式が成り立つ.

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

(ii) $l \in \mathbb{N}^+$ が奇数ならば $F^{(m)}(k_0,\dots,k_{l-1};\alpha) = 1/2$ である.
(iii) $F^{(m)}(k_0,\dots,k_{l-1};\alpha) = F^{(m)}(0,k_1-k_0,\dots,k_{l-1}-k_0;\alpha)$. 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, $\alpha \in \mathbb{T}^1$ は無理数とし, $F^{(m)}(0,k_1,\dots,k_{l-1};\alpha)$ 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. 各 $j = 1,2,\dots,l-1$ に対して[注11]

$$\begin{cases}\alpha_j &:= \langle k_j\alpha\rangle,\\ \alpha_j^{(m)L} &:= \lfloor\alpha_j\rfloor_m,\\ \alpha_j^{(m)U} &:= \lceil\alpha_j\rceil_m,\\ \beta_j^{(m)} &:= 2^m(\alpha_j - \alpha_j^{(m)L}),\end{cases}$$

とする. さらに

$$\beta_0^{(m)} := 1, \quad \beta_l^{(m)} := 0,$$

としておく. 次に集合 $\{0,1,\dots,l-1,l\}$ 上の置換 $\sigma(m,\bullet)$ を以下のように定める.

$$1 = \beta_{\sigma(m,0)}^{(m)} > \beta_{\sigma(m,1)}^{(m)} > \beta_{\sigma(m,2)}^{(m)} > \cdots > \beta_{\sigma(m,l-1)}^{(m)} > \beta_{\sigma(m,l)}^{(m)} = 0, \tag{4.11}$$

すなわち, とくに $\sigma(m,0) = 0$, $\sigma(m,l) = l$ である. さらに

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

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 4.13

$$F^{(m)}(0,k_1,\dots,k_{l-1};\alpha) = \sum_{s=0}^{l-1}\left(\beta_{\sigma(m,s)}^{(m)} - \beta_{\sigma(m,s+1)}^{(m)}\right)B(\boldsymbol{\alpha}^{(m),s}). \tag{4.12}$$

[注11] $\langle t\rangle$ は実数 $t \ge 0$ の小数部分を表す. すなわち $\langle t\rangle = t - \lfloor t\rfloor$.

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

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

および次の漸化式で計算される．

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

ただし，s_1, s_2 は次で与えられる．

$$s_1 \;:=\; \sum_{j=1}^{l-1} d_m(\alpha_j^{(m),s}), \qquad s_2 \;:=\; \sum_{j=1}^{s} d_m(\alpha_{\sigma(m,j)}). \tag{4.13}$$

定理 4.11 と補題 4.12 から，次の従属性消滅定理[注12]が従う．

定理 4.14 ([43, 61]) $\mathbb{P}$ 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．正確には，$\mathbb{P}$ に関してほとんどすべての $\alpha \in \mathbb{T}^1$ に対して，任意の $k \in \mathbb{N}^+$ と任意の $\epsilon_n \in \{0,1\}$，$n = 0, 1, \ldots, k-1$ について，α に依存しないある $0 < \rho < 1$ が存在して

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

また，この定理とは別に次の定理も成り立つ．

定理 4.15 ([62]) 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

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

—— ◇ —— ◇ ——

じつは筆者はこれらの従属性消滅定理について後述の § 5.2.2 定理 5.10 から着想を得た．計算量的に安全な疑似乱数生成器という概念について知ったのはそれより後のことであった．

こうした従属性消滅現象は，現実の数値計算において頻繁に起こっているだろうと想像される．すなわち，漸化式 (4.2) で定義されるような疑似乱数であっても，複雑な関数のサンプリングに適用されれば，生成されるサンプル列がランダムに見えることは十分あり得る．このことが，おそらく従来の簡易な疑似乱数生成器 (cf. [21]) でも役に立つサンプリングが可能であることの背景にあると思われる．

[注12] 安富は一連の論文 [59, 60, 61, 62] によって定理 4.14 を様々に拡張した諸定理を証明している．本書では，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．

4.2.4 有限次元分布の事前評価

定理 4.13 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，2 項間の相関について考えよう．

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

とおく．もし $\{Y_n^{(m)}(\bullet;\alpha)\}_{n=0}^{\infty}$ が硬貨投げの確率過程だったと仮定すれば $S_{N;k}^{(m)}$ 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

$$\mathbf{E}\left[S_{N;k}^{(m)}(\bullet;\alpha)\right] \equiv F^{(m)}(0,k;\alpha) = \frac{1}{2} \tag{4.14}$$

の検定を行うために

$$\left|S_{N;k}^{(m)}(\bullet;\alpha) - \frac{1}{2}\right| < 2\sigma_N = \frac{1}{\sqrt{N}} \tag{4.15}$$

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 (4.15) 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．

定理 4.16[注13]

$$N^{(m)}(k;\alpha) := \frac{1}{16\left(F^{(m)}(0,k;\alpha) - \frac{1}{2}\right)^2} \tag{4.16}$$

と定義する．m が十分大きいとき，$N = N^{(m)}(k;\alpha)$ (以下) ならば，事象 (4.15) の確率はおよそ 0.92 (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，$S_{N;k}^{(m)}$ 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 $\mathcal{N}(1/2+a,\sigma_N^2)$ 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，$|a| = 1/\left(4\sqrt{N}\right) = \sigma_N/2$ 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，上の事象の確率は簡単な変数変換によって

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

[注13] この主張は “定理” と呼ぶには曖昧だがご容赦あれ．

にほぼ等しいことが分かる. □

定理 4.16 により，仮説 (4.14) の検定 (危険率 8%) に棄却されずにすむ疑似乱数 $\{Y_n^{(m)}(\bullet;\alpha)\}_{n=0}^{\infty}$ の使用限界が $N^{(m)}(k;\alpha)$ 程度と考えることができよう．定理 4.16 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 でも定理 4.16 の評価はほぼ正しいことが次の例から分かる.

例 4.17 $\alpha = (\sqrt{5}-1)/2$，$m = 40$ および $k = 305$ として定理 4.13 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

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

となる．そこで確率

$$\mathbb{P}\left(\left|S^{(40)}_{7022;305}(\bullet;\alpha)-\frac{1}{2}\right| < \frac{1}{\sqrt{7022}}\right) \tag{4.17}$$

を数値的に求めてみた．計算方法は次の通りである;

$$\{Y_n^{(40)}(0;\lfloor\alpha\rfloor_{150})\}_{n=1}^{7022\times10^6+305}$$

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

$$p_i := \frac{1}{7022}\#\left\{7022(i-1)+1 \le j \le 7022\,i \,|\, \eta^{(40)}_{j;305}(0;\lfloor\alpha\rfloor_{150}) = 1\right\},$$

を計算する．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 $1/(4\times7022) = 0.0000356024$ 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，すなわち確率 (4.17) の値として 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する;

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

定理 4.16 を踏まえて，2 項間の相関について $K \in \mathbb{N}^+$ に対し

$$a^{(m)}(K) := \max_{1\le k\le K}\left|F^{(m)}(0,k;\alpha)-\frac{1}{2}\right|, \qquad N_c^{(m)}(K) := \frac{1}{16\,(a^{(m)}(K))^2}, \tag{4.18}$$

とする．$N_c^{(m)}(K)$ を臨界サンプル数と呼ぶ.[注14] (4.18) を $K = 10000$ のときに計算し，表にしたのが表 4.1 の左半分である．$a^{(m)}(10000)$ 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.[注15]

表 4.1: 2 項間・多項間分布の評価

m	$a^{(m)}(10000)$	(k)	$N_c^{(m)}(10000)$	$b^{(m)}(19)$	$k_1,\ldots$
10	0.4860680	(5473)	2.6×10^{-1}	0.1187876	18
20	0.1084934	(1449)	5.3×10^{0}	0.0088276	4, 5, 13, 14, 18
30	0.0435756	(305)	3.3×10^{1}	0.0009169	18
40	0.0029834	(305)	7.0×10^{3}	0.0000769	9
50	0.0001943	(610)	1.7×10^{6}	1.5×10^{-5}	18
60	0.0000136	(8484)	3.4×10^{8}	6.4×10^{-7}	18
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}	7.7×10^{-9}	18
90	8.5×10^{-9}	(165)	8.7×10^{14}	2.1×10^{-9}	16
100	2.8×10^{-9}	(5201)	8.1×10^{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．すなわち，各偶数 l について

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

を評価する．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 4.1 の右半分は $K = 19$ の場合を計算したものである．左の欄は

$$b^{(m)}(19) := \max_{1\le k_1<\ldots<k_{l-1}\le 19}\left|F^{(m)}(0,k_1,\ldots,k_{l-1};\alpha)-\frac{1}{2}\right|$$

を表わし，右の欄はその最大値がどのような $k_1,\ldots$ によって達成されたかを表わす．表 4.1 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.[注16]

仮説 4.18 各 $K \in \mathbb{N}^+$ に対して m が十分大きいとき

$$\max_{1\le k_1<\cdots<k_{l-1}\le K}\left|F^{(m)}(0,k_1,\ldots,k_{l-1};\alpha)-\frac{1}{2}\right| = \max_{1\le k\le K}\left|F^{(m)}(0,k;\alpha)-\frac{1}{2}\right|. \tag{4.19}$$

注14 [43] 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．

注15 α は無理数なので有限 2 進小数で近似して計算した．具体的には $\lfloor\alpha\rfloor_{150}$ と $\lfloor\alpha\rfloor_{300}$ としたが，両方とも同じ結果 (表 4.1) となった．

注16 詳しくは，$K = 19$ のとき $37 \le m \le 100$ について等式 (4.19) が成り立つことが分かっている．

じつは仮説 4.18 は定理 4.14 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 4.29)，現在のところ厳密な証明はない．もし仮説 4.18 が正しければ，我々は 2 項間の相関の最大値さえ評価すればよいことになる．

4.3　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 4.12 と定理 4.13，定理 4.15，定理 4.11 をこの順序で証明する．[注17] 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，(4.20) 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 $\alpha \in \mathbb{T}^1$ と $m \in \mathbb{N}^+$ に対して

$$X_n^{(m)}(x;\alpha) := \prod_{i=1}^{m} r_i(x + n\alpha), \qquad n \in \mathbb{N}, \tag{4.20}$$

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

$$\mathbb{P}\left((X_0^{(m)}(\bullet;\alpha), \ldots, X_{k-1}^{(m)}(\bullet;\alpha)) = \boldsymbol{\epsilon}\right) = \mathbb{P}\left((X_h^{(m)}(\bullet;\alpha), \ldots, X_{k-1+h}^{(m)}(\bullet;\alpha)) = \boldsymbol{\epsilon}\right). \tag{4.21}$$

この性質は定常性 (詳しくは強定常性) と呼ばれる．(4.21) 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．

4.3.1　補題 4.12 の証明

補題 4.12 を $\{X_n^{(m)}\}_{n=0}^{\infty}$ の言葉で表すと次のようになる．

補題 4.12′　(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．

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

[注17] この節は証明の細部まで技術的なことを述べる．先を急ぐ読者は初回は読み飛ばして次章から読むのがよいだろう．

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

$$\mathbb{P}\left(X_n^{(m)}(\bullet;\alpha) = \epsilon_n, \quad n = 0, \dots, k-1\right) = 2^{-k}\left(\sum_{l=1}^{k} \sum_{0\le k_0<\dots<k_{l-1}\le k-1} \prod_{j=0}^{l-1} \epsilon_{k_j} E^{(m)}(k_0, \dots, k_{l-1}; \alpha) + 1\right). \tag{4.22}$$

(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) 次の等式に注意する．

$$\sum_{l=1}^{k} \sum_{0\le k_0<\dots<k_{l-1}\le k-1} \prod_{j=0}^{l-1} \left(\epsilon_{k_j} X_{k_j}^{(m)}(x;\alpha)\right) = \prod_{n=0}^{k-1} \left(1 + \epsilon_n X_n^{(m)}(x;\alpha)\right) - 1. \tag{4.23}$$

ここで左辺を平均すれば

$$\sum_{l=1}^{k} \sum_{0\le k_0<\dots<k_{l-1}\le k-1} \prod_{j=0}^{l-1} \epsilon_{k_j} E^{(m)}(k_0, \dots, k_{l-1}; \alpha). \tag{4.24}$$

一方，右辺の平均は

$$\mathbf{E}\left[\prod_{n=0}^{k-1} \left(1 + \epsilon_n X_n^{(m)}(\bullet;\alpha)\right)\right] - 1. \tag{4.25}$$

ところが，(4.25) 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，(4.25) の値は次に等しいことが分かる．

$$2^k \mathbb{P}\left(X_n^{(m)}(\bullet;\alpha) = \epsilon_n, \quad n = 0, \dots, k-1\right) - 1. \tag{4.26}$$

(4.24) と (4.26) が等しいことより (4.22) 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

$$X_k^{(m)}\left(x + \frac{1}{2}; \alpha\right) = -X_k^{(m)}(x;\alpha), \qquad x \in \mathbb{T}^1, \tag{4.27}$$

であることがすぐ分かる．それで，l が奇数のとき

$$X_{k_0}^{(m)}(x;\alpha) \times \cdots \times X_{k_{l-1}}^{(m)}(x;\alpha) = -1$$

と

$$X_{k_0}^{(m)}\left(x + \frac{1}{2}; \alpha\right) \times \cdots \times X_{k_{l-1}}^{(m)}\left(x + \frac{1}{2}; \alpha\right) = 1$$

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

$$X_{k_0}^{(m)}(x;\alpha) \times \cdots \times X_{k_{l-1}}^{(m)}(x;\alpha) = 1$$

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 (ii) の主張が従う．(iii) は定常性 (4.21) により明らか． □

4.3.2　定理 4.13 の証明

定理 4.13 を $\{X_n^{(m)}\}_{n=0}^{\infty}$ の言葉で述べると次のようになる．

定理 4.13′

$$E^{(m)}(0, k_1, \ldots, k_{l-1}; \alpha) \;=\; \sum_{s=0}^{l-1} \left(\beta^{(m)}_{\sigma(m,s)} - \beta^{(m)}_{\sigma(m,s+1)}\right) A(\boldsymbol{\alpha}^{(m),s}). \tag{4.28}$$

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

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

および次の漸化式で帰納的に計算される．

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

ただし，s_1, s_2 などの記号は定理 4.13 と同じ．

定理 4.13′ の証明を行う．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

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

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

$$A^{(m)}(\boldsymbol{\alpha}) \;:=\; \mathbf{E}\left[\prod_{i=1}^{m} r_i(\bullet) r_i(\bullet + \alpha_1) \times \cdots \times r_i(\bullet + \alpha_{l-1})\right]. \tag{4.29}$$

補題 4.19　各 $\boldsymbol{\alpha} = (\alpha_1, \ldots, \alpha_{l-1}) \in (D_m)^{l-1}$ に対して

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

が成り立つ．

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

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

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

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

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 4.20 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 4.20 は 補題 4.19 によって正当化される．定理 4.13′ で述べられた関数 A は定義 4.20 で定義されたものに外ならない．以下の補題 4.21 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．各ベクトル $\boldsymbol{\alpha} \in \mathbb{T}^{l-1}$ に対して

$$\begin{aligned}\boldsymbol{\alpha}^{(m)L} &:= (\alpha_1^{(m)L},\ldots,\alpha_{l-1}^{(m)L}), \quad \alpha_j^{(m)L} := \lfloor \alpha_j \rfloor_m,\\ \boldsymbol{\alpha}^{(m)U} &:= (\alpha_1^{(m)U},\ldots,\alpha_{l-1}^{(m)U}), \quad \alpha_j^{(m)U} := \lceil \alpha_j \rceil_m,\end{aligned}$$

と定義する．このとき $\boldsymbol{\alpha}^{(m)L}, \boldsymbol{\alpha}^{(m)U} \in (D_m)^{l-1}$ は明らか．以上の定義の下で次の補題が成り立つ．

補題 4.21 (i) $A(\overbrace{0,\ldots,0}^{l-1}) = 1$.
(ii) 各 $\boldsymbol{\alpha} = (\alpha_1,\ldots,\alpha_{l-1}) \in (D_m)^{l-1}$ に対して $j_0 := \sum_{j=1}^{l-1} d_m(\alpha_j)$ と置く．このとき，次が成り立つ．

$$A(\boldsymbol{\alpha}) \;=\; \frac{(-1)^{j_0}}{2}\Big(A(\boldsymbol{\alpha}^{(m-1)U}) + A(\boldsymbol{\alpha}^{(m-1)L})\Big). \tag{4.30}$$

証明. (i) l は偶数なので

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

(ii) を示そう． 次が成り立つことに注意せよ．

$$\prod_{i=1}^{m} r_i(x+\alpha_j) \;=\; \begin{cases} \displaystyle\prod_{i=1}^{m-1} r_i\left(x+\alpha_j+2^{-m}\right) & (\,d_m(\alpha_j)=1,\ d_m(x)=1\,),\\ \displaystyle -\prod_{i=1}^{m-1} r_i\left(x+\alpha_j-2^{-m}\right) & (\,d_m(\alpha_j)=1,\ d_m(x)=0\,),\\ \displaystyle -\prod_{i=1}^{m-1} r_i\left(x+\alpha_j\right) & (\,d_m(\alpha_j)=0,\ d_m(x)=1\,),\\ \displaystyle\prod_{i=1}^{m-1} r_i\left(x+\alpha_j\right) & (\,d_m(\alpha_j)=0,\ d_m(x)=0\,).\end{cases} \tag{4.31}$$

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．以上で (4.31) が確かめられた．

記号を簡単にするため，次の状況を仮定しよう．

$$d_m(\alpha_j) \;=\; \begin{cases} 1 & (1\le j\le j_0), \\ 0 & (j_0+1\le j\le l-1). \end{cases}$$

$j_0=\sum_{j=1}^{l-1} d_m(\alpha_j)$ となることに注意せよ．このとき

$$\begin{aligned} A(\boldsymbol{\alpha}) \;&=\; \mathbf{E}\left[\prod_{i=1}^{m}\left(r_i(\bullet)\prod_{j=1}^{j_0} r_i(\bullet+\alpha_j)\prod_{j=j_0+1}^{l-1} r_i(\bullet+\alpha_j)\right)\right] \\ &=\; \mathbf{E}\left[-\prod_{i=1}^{m-1} r_i(\bullet)\prod_{j=1}^{j_0}\prod_{i=1}^{m-1} r_i\left(\bullet+\alpha_j+2^{-m}\right)\prod_{j=j_0+1}^{l-1}\left(-\prod_{i=1}^{m-1} r_i(\bullet+\alpha_j)\right);\, d_m(\bullet)=1\right] \\ &\quad +\mathbf{E}\left[\prod_{i=1}^{m-1} r_i(\bullet)\prod_{j=1}^{j_0}\left(-\prod_{i=1}^{m-1} r_i\left(\bullet+\alpha_j-2^{-m}\right)\right)\prod_{j=j_0+1}^{l-1}\prod_{i=1}^{m-1} r_i(\bullet+\alpha_j);\, d_m(\bullet)=0\right]. \end{aligned}$$

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

$$\begin{aligned} &=\; \frac{1}{2}\,\mathbf{E}\left[\prod_{i=1}^{m-1} r_i(\bullet)\prod_{j=1}^{l-1}\prod_{i=1}^{m-1} r_i(\bullet+\alpha_j^{(m-1)U})\times(-1)^{l-j_0}\right] \\ &\quad +\frac{1}{2}\,\mathbf{E}\left[\prod_{i=1}^{m-1} r_i(\bullet)\prod_{j=1}^{l-1}\prod_{i=1}^{m-1} r_i(\bullet+\alpha_j^{(m-1)L})\times(-1)^{j_0}\right]. \end{aligned}$$

となることが分かる．さて，j_0 が偶数であれば $l-j_0$ 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 j_0 が奇数であれば $l-j_0$ 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． □

定理 4.13′ の証明.

第 1 段

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

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．

$$\begin{aligned} &E^{(m)}(0, k_1, \ldots, k_{l-1}; \alpha) \\ &\quad = \sum_{j=0}^{l-1} \mathbf{E}\left[\prod_{i=1}^{m}\left(r_i(\bullet)\prod_{p=1}^{j} r_i(\bullet + \alpha^{(m)U}_{\sigma(m,p)})\prod_{p=j+1}^{l-1} r_i(\bullet + \alpha^{(m)L}_{\sigma(m,p)})\right);\ C_j\right] \end{aligned}$$

ここで被積分関数は C_j と独立だから

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

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

定理 4.13′ の前半部分は $\mathbb{P}(C_j) = \beta^{(m)}_{\sigma(m,j)} - \beta^{(m)}_{\sigma(m,j+1)}$ であることを用いれば証明が終わる．

第 2 段 定理 4.13′ の後半部分は補題 4.21 に訴えて示すことができる．そのためには

$$\left(\boldsymbol{\alpha}^{(m),s}\right)^{(m-1)U} = \boldsymbol{\alpha}^{(m-1),s_1+s_2}, \tag{4.32}$$

$$\left(\boldsymbol{\alpha}^{(m),s}\right)^{(m-1)L} = \boldsymbol{\alpha}^{(m-1),s_2}, \tag{4.33}$$

を示せばよい.

第 2–1 段 はじめに (4.33) を示そう. そのために

$$\#\left\{ j \,\middle|\, \left(\alpha_j^{(m),s}\right)^{(m-1)L} = \alpha_j^{(m-1)U} \right\} = s_2 \tag{4.34}$$

であることを示す. 次の四つの論理包含 (complications) を確認しよう.

$$\begin{aligned} \alpha_j^{(m),s} \neq \alpha_j^{(m)U} &\implies \alpha_j^{(m),s} = \alpha_j^{(m)L} \\ &\implies \left(\alpha_j^{(m),s}\right)^{(m-1)L} = \left(\alpha_j^{(m)L}\right)^{(m-1)L} = \alpha_j^{(m-1)L} \\ &\implies \left(\alpha_j^{(m),s}\right)^{(m-1)L} \neq \alpha_j^{(m-1)U}, \end{aligned} \tag{4.35}$$

$$\alpha_j^{(m),s} = \alpha_j^{(m)U} \implies \left(\alpha_j^{(m),s}\right)^{(m-1)U} = \left(\alpha_j^{(m)U}\right)^{(m-1)U} = \alpha_j^{(m-1)U}, \tag{4.36}$$

$$d_m(\alpha_j^{(m),s}) = 1 \iff \left(\alpha_j^{(m),s}\right)^{(m-1)L} \neq \left(\alpha_j^{(m),s}\right)^{(m-1)U} \tag{4.37}$$

$$\implies \left(\alpha_j^{(m),s}\right)^{(m-1)L} \neq \alpha_j^{(m-1)U}, \tag{4.38}$$

$$d_m(\alpha_j^{(m),s}) = 0 \iff \left(\alpha_j^{(m),s}\right)^{(m-1)L} = \left(\alpha_j^{(m),s}\right)^{(m-1)U}. \tag{4.39}$$

(4.35) と (4.38) 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 (4.36) と (4.39) 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

$$\begin{aligned} J_0 &:= \left\{ j \,\middle|\, \left(\alpha_j^{(m),s}\right)^{(m-1)L} = \alpha_j^{(m-1)U} \right\} \\ &= \left\{ j \,\middle|\, \alpha_j^{(m),s} = \alpha_j^{(m)U},\ d_m(\alpha_j^{(m),s}) = 0 \right\} \\ &= \left\{ \sigma(m,j) \,\middle|\, \alpha_{\sigma(m,j)}^{(m),s} = \alpha_{\sigma(m,j)}^{(m)U},\ d_m(\alpha_{\sigma(m,j)}^{(m),s}) = 0 \right\} \\ &= \left\{ \sigma(m,j) \,\middle|\, 1 \le j \le s,\ d_m(\alpha_{\sigma(m,j)}^{(m),s}) = d_m(\alpha_{\sigma(m,j)}^{(m)U}) = 0 \right\} \\ &= \left\{ \sigma(m,j) \,\middle|\, 1 \le j \le s,\ d_m(\alpha_{\sigma(m,j)}^{(m)L}) = d_m(\alpha_{\sigma(m,j)}) = 1 \right\} =: J_1. \end{aligned}$$

集合 J_1 の元の個数は s_2 に等しい．従って (4.34) が示された．

第 2–2 段 $\beta_i^{(m)}$ の定義より等式

$$\beta_i^{(m-1)} = \frac{1}{2}\beta_i^{(m)} + \frac{1}{2}d_m(\alpha_i), \quad 1 \le i \le l-1,$$

が容易に導かれる．これより，$\{\beta_i^{(m-1)}\}_i$ を大きい方から順に並べるとき，$d_m(\alpha_i) = 1$ であるような i に対する $\beta_i^{(m-1)}$ が上位に来る．それで

$$\beta_i^{(m-1)} > \beta_j^{(m-1)} \iff \begin{cases} 1 = d_m(\alpha_i) > d_m(\alpha_j) = 0, \\ \text{または} \\ d_m(\alpha_i) = d_m(\alpha_j) \quad \text{かつ} \quad \beta_i^{(m)} > \beta_j^{(m)}. \end{cases}$$

第 2–3 段 次に

$$J_2 := \{\, i \,|\, 1 \le \exists\, j \le s_2 \ \text{ s.t. } \ i = \sigma(m-1, j) \,\}$$

とするとき $J_1 = J_2$ を示す．$i \in J_1$ のとき，ある $1 \le j \le s$ が存在して $i = \sigma(m, j)$ かつ $d_m(\alpha_i) = 1$ だから，s_2 の定義と 第 2–2 段 に注意すれば，$\beta_i^{(m-1)}$ は $\{\beta_j^{(m-1)}\}_j$ の中で大きい方から s_2 番目以内であることが分かる．従って $i \in J_2$，すなわち $J_1 \subset J_2$．ところが $\#J_1 = \#J_2 = s_2$ であるから，$J_1 = J_2$ である．

第 2–4 段 $i \in J_2$ のとき，$\alpha_i^{(m-1),s_2} = \alpha_i^{(m-1)U}$ であり，$i \notin J_2$ ならば $\alpha_i^{(m-1),s_2} = \alpha_i^{(m-1)L}$ である．$J_2 = J_1 = J_0$ だから，$i \in J_0$ ならば $\alpha_i^{(m-1),s_2} = \alpha_i^{(m-1)U} = \left(\alpha_i^{(m),s}\right)^{(m-1)L}$ であり，$i \notin J_0$ ならば $\alpha_i^{(m-1),s_2} = \alpha_i^{(m-1)L} = \left(\alpha_i^{(m),s}\right)^{(m-1)L}$ である．このことは (4.33) を示している．

第 2–5 段 では (4.32) を示すために，はじめに

$$\#\left\{ j \,\middle|\, \left(\alpha_j^{(m),s}\right)^{(m-1)U} = \alpha_j^{(m-1)U} \right\} = \#\left\{ j \,\middle|\, \left(\alpha_j^{(m),s}\right)^{(m-1)U} \neq \alpha_j^{(m-1)L} \right\} = s_1 + s_2 \qquad (4.40)$$

を示そう．明らかに

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

$$\iff \begin{cases} \left(\alpha_j^{(m),s}\right)^{(m-1)U} \neq \left(\alpha_j^{(m),s}\right)^{(m-1)L} \\ \text{または}^{\text{注18}} \\ \left(\alpha_j^{(m),s}\right)^{(m-1)L} \neq \alpha_j^{(m-1)L} \end{cases}$$

すなわち，

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

が成り立ち，これは互いに排反な集合の和である．ところが (4.37) より

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

一方，(4.34) より

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

これらより (4.40) が従う．

第 2–6 段

$$J_3 := \left\{ j \,\middle|\, d_m(\alpha_j^{(m),s}) = 1 \right\}$$

とおけば，いままでの議論から

$$\left\{ j \,\middle|\, \left(\alpha_j^{(m),s}\right)^{(m-1)U} = \alpha_j^{(m-1)U} \right\} = J_3 \cup J_0 = J_3 \cup J_2$$

であり，これは排反な集合の和である．ここで，

$$J_4 := \{ i \,|\, 1 + s_2 \le \exists\, j \le s_1 + s_2 \ \text{ s.t. } \ i = \sigma(m-1, j) \}$$

とすれば $J_3 = J_4$ であることを示そう．$\alpha_j^{(m),s}$ の定義から

$$\begin{aligned} J_3 &= \left\{ \sigma(m,j) \,\middle|\, 1 \le j \le s,\ d_m(\alpha_{\sigma(m,j)}) = 0 \right\} \\ &\quad \cup \left\{ \sigma(m,j) \,\middle|\, s+1 \le j \le l-1,\ d_m(\alpha_{\sigma(m,j)}) = 1 \right\} \\ &=: J_5 \cup J_6 \end{aligned}$$

が分かる．$\{\beta_i^{(m-1)}\}_i$ を大きい順に並べると，最初の s_2 位までは $J_0 = J_2$ に属する i に対する $\beta_i^{(m-1)}$ が来るのであった．さらに Step 2.2 に注意すれば，次に来るのが J_6 に属する i に対する $\beta_i^{(m-1)}$ であり，そしてその次が J_5 に属する i に対する $\beta_i^{(m-1)}$ であることが分かる．このことと，Step 2.5 の結果から $J_3 = J_4$ が従う．

以上をまとめると

$$\left\{ j \,\middle|\, \left(\alpha_j^{(m),s}\right)^{(m-1)U} = \alpha_j^{(m-1)U} \right\} = J_2 \cup J_4 = \{ i \,|\, 1 \le \exists\, j \le s_1 + s_2 \ \text{ s.t. } \ i = \sigma(m-1, j) \}$$

が分かるが，これは (4.32) を示している． □

注18 ここでの “または” は「どちらか一方のみが成り立つ」の意．

4.3.3 定理 4.15 の証明

定理 4.15 の証明は補題 4.12′ 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 $l-1$ 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.

証明に入る前にそのアイデアを述べておく．まず，定理 4.13′ のアルゴリズムを具体的な場合に書き下してみよう．$l = 4$ 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

$$1 = \beta_0^{(6)} > \beta_1^{(6)} = 0.110\ldots > \beta_2^{(6)} = 0.101\ldots > \beta_3^{(6)} = 0.011\ldots > \beta_4^{(6)} = 0$$

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 4.13′ および補題 4.21 を忠実に図化すると図 4.1 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.

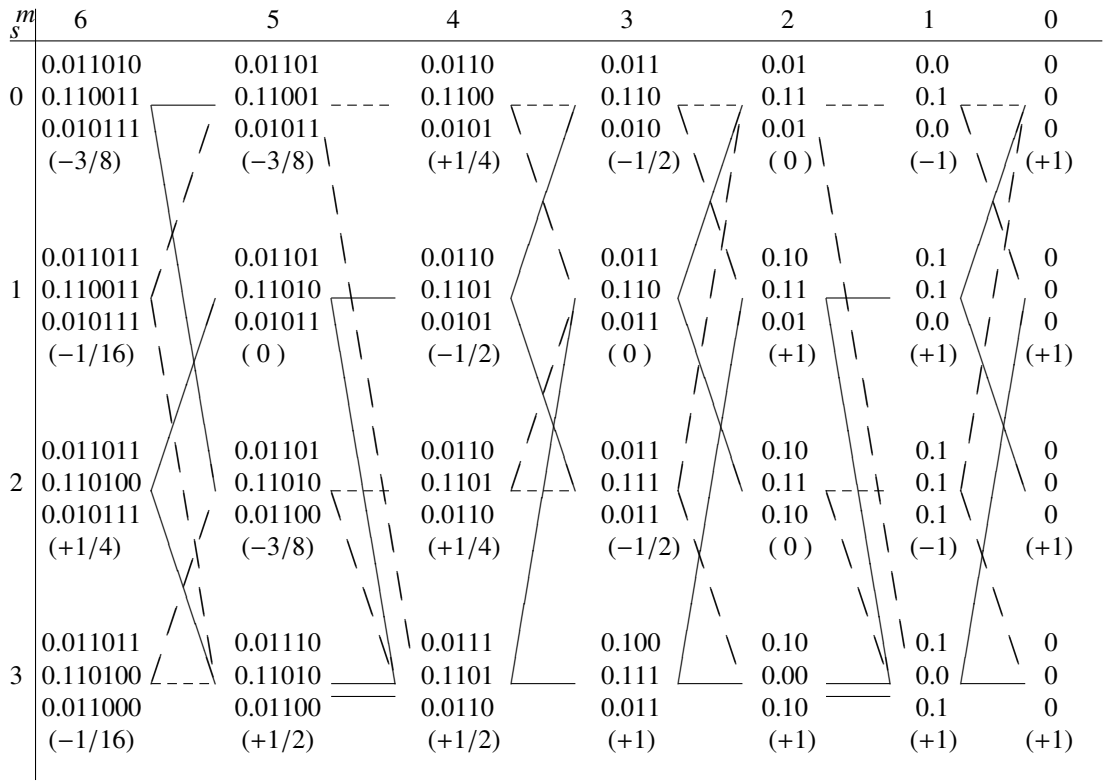
図 4.1 のダイヤグラムの右から m 番目 ($m = 0, \ldots, 6$)，上から s 番目 ($s = 0, \ldots, 3$) のベクトルは $\boldsymbol{\alpha}^{(m),s}$ を表す．たとえば $\boldsymbol{\alpha}^{(6),0}$ は左上角のベクトルである．各ベクトルのすぐ下の ( ) の中の数は，関数 $A(\boldsymbol{\alpha}^{(m),s})$ の値を表す．補題 4.21 によれば $A(\boldsymbol{\alpha}^{(m),s})$ は $m-1$ の二つの同様の量によって (4.30) のように書き表されるが，そのとき (4.30) の符号 $(-1)^{j_0}$ が正のときは実線で，負のときは破線で記した．このダイヤグラムから，たとえば

$$A(\boldsymbol{\alpha}^{(6),2}) \;=\; \frac{1}{2}\Big(A(\boldsymbol{\alpha}^{(5),1}) + A(\boldsymbol{\alpha}^{(5),3})\Big) \tag{4.41}$$

が分かる.

定理 4.15 を証明するために，$m \to \infty$ のとき，任意の s に対して $|A(\boldsymbol{\alpha}^{(m),s})| \to 0$ 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.

図 4.1: 定理 4.13′ と補題 4.21 のダイアグラム



たとえば (4.41) 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 4.2 で示すような二つのルートでキャンセル (打ち消しあい) が起こっていることが分かる．すなわち，図 4.2 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

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

が分かる．

定理 4.15 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 4.27 を見よ) ことにより達成される．

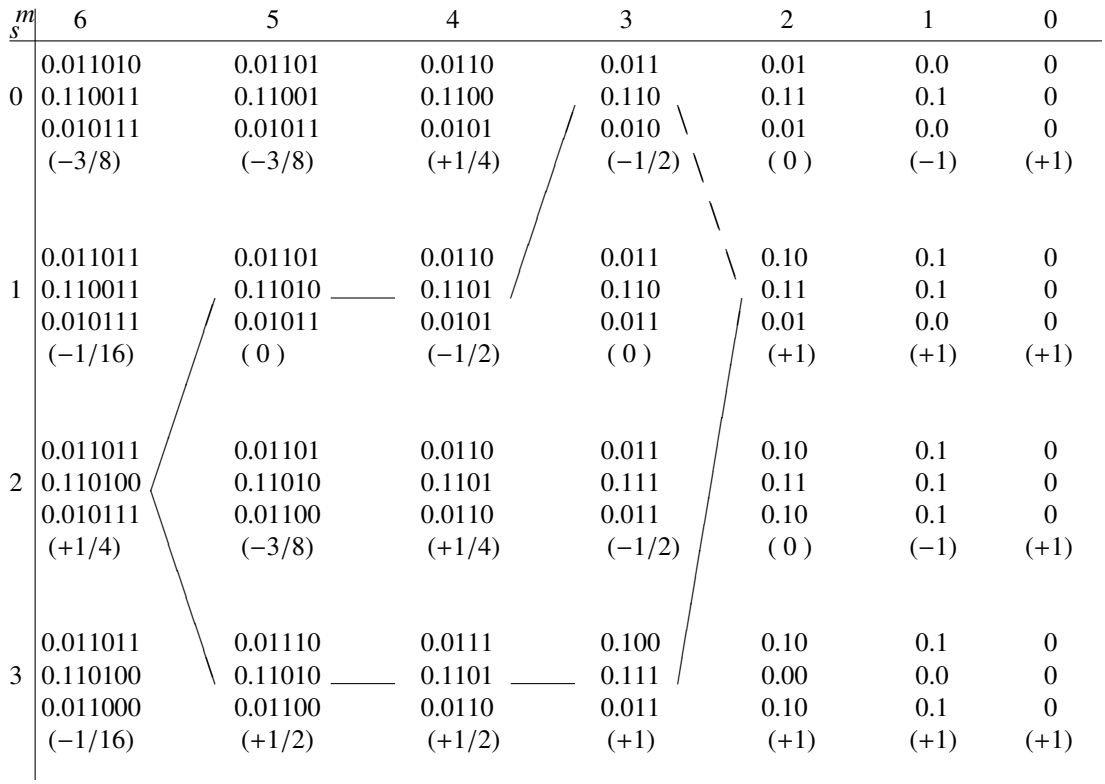
定理 4.15 を証明するためにいくつかの補題を準備する．

補題 4.22　$\boldsymbol{\alpha} = (\alpha_1, \dots, \alpha_{l-1}) \in (\mathbb{T}^1 \setminus D)^{l-1}$ とする．$m \ge 1$ のとき

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

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

図 4.2: キャンセルする二つのルート



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

(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 (4.13) 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

$$\begin{aligned}\sum_{j=1}^{l-1} d_m(\alpha_j^{(m),0}) &= \sum_{j=1}^{l-1}\left(1 - d_m(\alpha_j^{(m),l-1})\right)\\ &= l-1-\sum_{j=1}^{l-1} d_m(\alpha_j^{(m),l-1}).\end{aligned} \tag{4.42}$$

$l-1$ 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，(4.13) 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 4.23　記法を簡潔にするため，次のような写像を導入する．

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

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 4.24　$\boldsymbol{\alpha} = (\alpha_1, \ldots, \alpha_{l-1}) \in (\mathbb{T}^1 \setminus D)^{l-1}$ 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)

$$\forall j = 1, \ldots, l-1, \quad 1 \le \exists p \le r, \quad d_{m+p}(\alpha_j) = 0 \tag{4.43}$$

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

(iv)

$$\forall j = 1, \ldots, l-1, \quad 1 \le \exists p \le r, \quad d_{m+p}(\alpha_j) = 1 \tag{4.44}$$

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

証明. (i) (4.13) より， s_2 は s の増加関数である．このことより (i) が従う．
(ii) (4.40) より，$s_1 + s_2$ は s の増加関数である．このことより (ii) が従う．
(iii) 条件 (4.43) と補題 4.22(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 4.22(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) が従う．
(iv) 条件 (4.44) 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 4.22(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) が従う．
□

補題 4.25 ([62]) $r := 3k_{l-1}$ 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，(4.43) と (4.44) 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．(4.43) と (4.44) 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 $m \geq N$ 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 α の 2 進展開の第 $N+1$ 桁以下が周期的になることを示す．

第 1 段 $m \geq N$ を固定するとき，有限列 $d_{m+1}(\alpha), d_{m+2}(\alpha), \ldots, d_{m+r}(\alpha)$ 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} で割ることによって 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 $0 \leq k_{j_m}\langle 2^{m+1}\alpha\rangle - \langle 2^{m+1}k_{j_m}\alpha\rangle < k_{j_m}$ なので，$2R_1 + d_{m+1}(k_{j_m}\alpha)$ を k_{j_m} で割ったときの商を Q_1，余りを R_2 とすれば

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

すなわち

$$R_2 + \langle 2^{m+1}k_{j_m}\alpha\rangle = k_{j_m}\langle 2^{m+1}\alpha\rangle.$$

次に $2R_2 + d_{m+2}(k_{j_m}\alpha)$ を k_{j_m} で割ったときの商を Q_2，余りを R_3 とおけば，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 $2R_u + d_{m+u}(k_{j_m}\alpha) = k_{j_m}Q_u + R_{u+1}$，$0 \le R_{u+1} < k_{j_m}$，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，列 $\{(Q_u, R_{u+1})\}_{u=1}^{r}$ は列 $\{R_u\}_{u=1}^{r}$ 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．

第2段 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 $2w \le p$ 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 $0 \le u \le p-w'-1$ に対して $u - q = wj + v$，$j \in \mathbb{Z}$，$0 \le v < w$，とできるので，$v + w' < w + w' \le 2w \le p'$ 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．

第3段 補題の主張を示す．$m \ge N$ とする．Step 1 より，$d_{m+1}(\alpha), \ldots, d_{m+r}(\alpha)$ は周期高々 k_{j_m} なる周期列で，その最小周期を w_m とする．同様にある $k_{j_{m+1}}$ が存在し $d_{m+2}(\alpha), \ldots, d_{m+1+r}(\alpha)$ は周期高々 $k_{j_{m+1}}$ なる周期列で，その最小周期を w_{m+1} とする．さて $d_{m+2}(\alpha), \ldots, d_{m+r}(\alpha)$ は周期列でその最小周期は Step 2 により w_m にも w_{m+1} にも等しい．すなわち $w_m = w_{m+1}$ である．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 $w := w_m = w_{m+1}$ で循環することが分かる． □

定理 4.15 の証明に戻ろう．補題 4.12′ によって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

$$\lim_{m\to\infty} E^{(m)}(0, k_1, \ldots, k_{l-1}; \alpha) = 0 \tag{4.45}$$

を示すことであった．α を無理数とし，再び

$$\boldsymbol{\alpha} := (\alpha_1, \ldots, \alpha_{l-1}), \qquad \alpha_j := \langle k_j\alpha\rangle, \tag{4.46}$$

とおく．このとき，定理 4.13′ によれば (4.45) を示すには

$$\lim_{m\to\infty} \max_{0\le s\le l-1} \left|A(\boldsymbol{\alpha}^{(m),s})\right| = 0 \tag{4.47}$$

を示せばよい．

補題 4.21 によって

$$A(\boldsymbol{\alpha}^{(m),s}) \;=\; \pm\frac{1}{2}\left\{A(\mathcal{U}\boldsymbol{\alpha}^{(m),s}) + A(\mathcal{L}\boldsymbol{\alpha}^{(m),s})\right\} \tag{4.48}$$

である．次の補題は (4.48) から直ちに導かれる．

補題 4.26 $\max_{0\le s\le l-1}|A(\boldsymbol{\alpha}^{(m'),s})| \le \max_{0\le s\le l-1}|A(\boldsymbol{\alpha}^{(m),s})|, \qquad m' > m.$

そこで次の鍵になる補題が示される．

補題 4.27 α を無理数，$r := 3k_{l-1}$ とする．$\{m_n\}_{n=0}^{\infty}$ を補題 4.25 で述べられた無限個の $m \ge 2$ のうち $m_n + r + 2 \le m_{n+1}$ となるものをとって並べたものとする．このとき

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

証明. (4.48) を r 回用いれば次の表現を得る．

$$\begin{aligned} A(\boldsymbol{\alpha}^{(m_n+r),s}) \;=\; & \frac{1}{2^r}\epsilon_{\mathcal{U}^r}A(\mathcal{U}^r\boldsymbol{\alpha}^{(m_n+r),s}) + \frac{1}{2^r}\epsilon_{\mathcal{L}\mathcal{U}^{r-1}}A(\mathcal{L}\mathcal{U}^{r-1}\boldsymbol{\alpha}^{(m_n+r),s}) + \\ & \qquad\vdots \\ & + \frac{1}{2^r}\epsilon_{\mathcal{U}\mathcal{L}^{r-1}}A(\mathcal{U}\mathcal{L}^{r-1}\boldsymbol{\alpha}^{(m_n+r),s}) + \frac{1}{2^r}\epsilon_{\mathcal{L}^r}A(\mathcal{L}^r\boldsymbol{\alpha}^{(m_n+r),s}), \end{aligned} \tag{4.49}$$

ここで $\epsilon_{\mathcal{U}^r},\dots,\epsilon_{\mathcal{L}^r} = \pm 1$ である．補題 4.24 によって，各 s に対して

$$\mathcal{U}^r\boldsymbol{\alpha}^{(m_n+r),s} = \boldsymbol{\alpha}^{(m_n),l-1}, \qquad \mathcal{L}^r\boldsymbol{\alpha}^{(m_n+r),s} = \boldsymbol{\alpha}^{(m_n),0}, \tag{4.50}$$

である．

第 1 段 まず，$\epsilon_{\mathcal{U}^r} = \epsilon_{\mathcal{L}^r}$ の場合．$\epsilon, \epsilon' = \pm 1$ として (4.48) より

$$\begin{cases} \epsilon_{\mathcal{U}^r}A(\mathcal{U}^r\boldsymbol{\alpha}^{(m_n+r),s}) \;=\; \epsilon_{\mathcal{U}^r}\epsilon\left\{\dfrac{1}{2}A(\mathcal{U}^{r+1}\boldsymbol{\alpha}^{(m_n+r),s}) + \dfrac{1}{2}A(\mathcal{L}\mathcal{U}^r\boldsymbol{\alpha}^{(m_n+r),s})\right\}, \\ \epsilon_{\mathcal{L}^r}A(\mathcal{L}^r\boldsymbol{\alpha}^{(m_n+r),s}) \;=\; \epsilon_{\mathcal{L}^r}\epsilon'\left\{\dfrac{1}{2}A(\mathcal{U}\mathcal{L}^r\boldsymbol{\alpha}^{(m_n+r),s}) + \dfrac{1}{2}A(\mathcal{L}^{r+1}\boldsymbol{\alpha}^{(m_n+r),s})\right\}. \end{cases} \tag{4.51}$$

補題 4.21，補題 4.22(iv)，(4.50) によって，$\epsilon \ne \epsilon'$ である．(4.50) と補題 4.22(v) から

$$\mathcal{L}\mathcal{U}^r\boldsymbol{\alpha}^{(m_n+r),s} \;=\; \mathcal{U}\mathcal{L}^r\boldsymbol{\alpha}^{(m_n+r),s} \tag{4.52}$$

が従うことに注意せよ． さて，(4.51) を用いて (4.49) をもう一度展開すれば

$$\begin{aligned} A(\boldsymbol{\alpha}^{(m_n+r),s}) \;=\; & \frac{1}{2^{r+1}}\epsilon_{\mathcal{U}^r}\epsilon A(\mathcal{U}^{r+1}\boldsymbol{\alpha}^{(m_n+r),s}) + \frac{1}{2^{r+1}}\epsilon_{\mathcal{U}^r}\epsilon A(\mathcal{L}\mathcal{U}^r\boldsymbol{\alpha}^{(m_n+r),s}) + \\ & \qquad\vdots \\ & + \frac{1}{2^{r+1}}\epsilon_{\mathcal{L}^r}\epsilon' A(\mathcal{U}\mathcal{L}^r\boldsymbol{\alpha}^{(m_n+r),s}) + \frac{1}{2^{r+1}}\epsilon_{\mathcal{L}^r}\epsilon' A(\mathcal{L}^{r+1}\boldsymbol{\alpha}^{(m_n+r),s}) \end{aligned} \tag{4.53}$$

が得られる．いま，$\epsilon_{\mathcal{U}^r}\epsilon \neq \epsilon_{\mathcal{L}^r}\epsilon'$ だから (4.52) より

$$\frac{1}{2^{r+1}}\epsilon_{\mathcal{U}^r}\epsilon A(\mathcal{L}\mathcal{U}^r\boldsymbol{\alpha}^{(m_n+r),s}) + \frac{1}{2^{r+1}}\epsilon_{\mathcal{L}^r}\epsilon' A(\mathcal{U}\mathcal{L}^r\boldsymbol{\alpha}^{(m_n+r),s}) \;=\; 0. \tag{4.54}$$

従って次の評価が得られる．

$$\left|A(\boldsymbol{\alpha}^{(m_n+r),s})\right| \;\leq\; \left(1 - \frac{1}{2^{r+1}} \times 2\right) \max_{0\leq q\leq l-1} \left|A(\boldsymbol{\alpha}^{(m_n-1),q})\right|. \tag{4.55}$$

第 2 段　$\epsilon_{\mathcal{U}^r} \neq \epsilon_{\mathcal{L}^r}$ の場合．このときは，展開 (4.53) において $\epsilon_{\mathcal{U}^r}\epsilon = \epsilon_{\mathcal{L}^r}\epsilon'$ だから，(4.49) の代わりに (4.53) から出発して第 1 段と同じ方法で，評価

$$\left|A(\boldsymbol{\alpha}^{(m_n+r),s})\right| \;\leq\; \left(1 - \frac{1}{2^{r+2}} \times 2\right) \max_{0\leq q\leq l-1} \left|A(\boldsymbol{\alpha}^{(m_n-2),q})\right| \tag{4.56}$$

を得ることができる．

補題 4.26 によれば，(4.55) も評価 (4.56) を導く．すなわち，いずれの場合にも評価 (4.56) を得る．□

もはや定理 4.15 の証明は容易である．補題 4.27 から，結局，

$$\begin{aligned}
\max_{0\leq s\leq l-1} \left|A(\boldsymbol{\alpha}^{(m_n+r),s})\right| \;&\leq\; \left(1 - \frac{1}{2^{r+1}}\right) \max_{0\leq s\leq l-1} \left|A(\boldsymbol{\alpha}^{(m_n-2),s})\right| \\
&\leq\; \left(1 - \frac{1}{2^{r+1}}\right) \max_{0\leq s\leq l-1} \left|A(\boldsymbol{\alpha}^{(m_{n-1}+r),s})\right| \\
&\leq\; \left(1 - \frac{1}{2^{r+1}}\right)^2 \max_{0\leq s\leq l-1} \left|A(\boldsymbol{\alpha}^{(m_{n-2}+r),s})\right| \\
&\leq\; \cdots\cdots\cdots \\
&\leq\; \left(1 - \frac{1}{2^{r+1}}\right)^n \max_{0\leq s\leq l-1} \left|A(\boldsymbol{\alpha}^{(m_0+r),s})\right| \\
&\leq\; \left(1 - \frac{1}{2^{r+1}}\right)^n \longrightarrow\; 0, \qquad n \to \infty.
\end{aligned} \tag{4.57}$$

が得られる．これから (4.47) が分かる．□

注意 4.28　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 (4.57) は，ほとんどすべての α に対して収束 (4.47) が指数的に早いことを意味している (定理 4.11′ 参照).

注意 4.29　$l \geq 4$ に対して上の証明で述べたキャンセル (4.54) は非常に特殊なもので，展開 (4.49) 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．

4.3.4 定理 4.11 の証明

エルゴード理論を用いて定理 4.11 を示す．ただし，ここでも証明するのは，$\{Y_n^{(m)}\}_{n=0}^{\infty}$ 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 4.11′ $\mathbb{P}$ に関してほとんどすべての $\alpha \in \mathbb{T}^1$ について，任意の $l \in \mathbb{N}^+$，$0 \le k_0 < \ldots < k_{l-1}$，に対して，$\alpha$ に依存しないある $0 < \rho < 1$ が存在し

$$\mathbf{E}\left[X_0^{(m)}(\bullet;\alpha) X_{k_1}^{(m)}(\bullet;\alpha) \times \cdots \times X_{k_{l-1}}^{(m)}(\bullet;\alpha) \right] = o(\rho^m), \quad m \to \infty, \tag{4.58}$$

が成り立つ．

群拡大による定式化

示すべき (4.58) の左辺をあるマルコフ連鎖の 2 項相関に関係する量として表し (cf. (4.64))，その混合性に訴えて定理 4.11′ を証明する．そのために以下の設定を行う．

最初に，補題 4.12′ により，定理 4.11′ は l が偶数のときに示せばよいことに注意する．任意に $0 < k_1 < \cdots < k_{l-1} \in \mathbb{N}^+$ をとり固定する．2 次元トーラス上の関数 f を

$$f(x,\alpha) := r_1(x) r_1(x + k_1\alpha) \times \cdots \times r_1(x + k_{l-1}\alpha), \qquad (x,\alpha) \in \mathbb{T}^2, \tag{4.59}$$

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

$$\beta(x,y,\alpha) := (2x, 2y, 2\alpha) \tag{4.60}$$

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

定義 4.30 $\Omega := \mathbb{T}^3 \times \{-1,1\}^2$，$\mu$ を Ω 上の一様確率測度，すなわち

$$\mu := \mathbb{P}^3 \otimes \frac{\delta_{-1} + \delta_1}{2} \otimes \frac{\delta_{-1} + \delta_1}{2} \tag{4.61}$$

とする．ここに $\otimes$ は確率測度の直積，δ_i は i に集中した δ-測度を表す．変換 $T_f : \Omega \to \Omega$ を

$$T_f(x,y,\alpha,\epsilon_1,\epsilon_2) := (2x, 2y, 2\alpha, \epsilon_1 f(x,\alpha), \epsilon_2 f(y,\alpha)) \tag{4.62}$$

で定義する．[注19]

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

$$C := \{(x,y,\alpha) \,|\, (x,y,\alpha) \text{ は } f(x,\alpha) \text{ または } f(y,\alpha) \text{ の不連続点}\} \tag{4.63}$$

[注19] ここで紹介する群拡大の方法では安富のアイデア [59, 60] を参考にした．なお，先駆的仕事として高信 [55] 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．

とすれば，C は測度 0 で $\beta C \subset C$ を満たす．$\mathbb{T}^3 - C$ の各連結成分を E_j，$j = 1, \dots, J$，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 4.31　(Ω, μ) 上で定義された $\{1, 2, 3, \dots, 4J\}$-値確率過程 $\{\zeta_m\}_{m=0}^{\infty}$ を次のように定義する; $T_f^m(x, y, \alpha, \epsilon_1, \epsilon_2) \in F_j$ のとき，$\zeta_m(x, y, \alpha, \epsilon_1, \epsilon_2) := j$ とする．

次の補題が鍵である．

補題 4.32　$\{\zeta_m\}_{m=0}^{\infty}$ 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．[注20]

補題 4.32 は後に示す．$p^m(i, j) := \mu(\zeta_m = j \,|\, \zeta_0 = i)$ とおく．すると，補題 4.32 より直ちに次の系を得る (cf. [1] Theorem 8.9).

系 4.33　任意の $i, j = 1, \dots, 4J$ に対して

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

が成り立ち，しかもこの収束は指数関数的である．

系 4.33 によって定理 4.11′ は以下のようにして示される．まず，次の四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

$$\begin{aligned}X_0^{(m)}(x; \alpha) \times \cdots \times X_{k_{l-1}}^{(m)}(x; \alpha) &= f(x, \alpha) \times \cdots \times f(2^{m-1}x, 2^{m-1}\alpha) \\ &= \Phi_1(x, y, \alpha, \epsilon_1, \epsilon_2)\Phi_1(T_f^m(x, y, \alpha, \epsilon_1, \epsilon_2)) \\ &= \widetilde{\Phi}_1(\zeta_0(x, y, \alpha, \epsilon_1, \epsilon_2))\widetilde{\Phi}_1(\zeta_m(x, y, \alpha, \epsilon_1, \epsilon_2))\end{aligned}$$

[注20] 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，[27] 第 11 章や [1] Section 8 などを見よ．

と表される．この式の右辺は y, ϵ_1, ϵ_2 によらないことに注意せよ．これより

$$\mathbf{E}\left[X_0^{(m)}(\bullet;\alpha)\times\cdots\times X_{k_{l-1}}^{(m)}(\bullet;\alpha)\right]=\int_{\mathbb{T}^1}dx\,\widetilde{\Phi}_1(\zeta_0(x,y,\alpha,\epsilon_1,\epsilon_2))\widetilde{\Phi}_1(\zeta_m(x,y,\alpha,\epsilon_1,\epsilon_2)).$$

次のような計算をする．

$$\begin{aligned}&\int_{\mathbb{T}^1}d\alpha\left(\int_{\mathbb{T}^1}dx\,\widetilde{\Phi}_1(\zeta_0(x,y,\alpha,\epsilon_1,\epsilon_2))\widetilde{\Phi}_1(\zeta_m(x,y,\alpha,\epsilon_1,\epsilon_2))\right)^2\\&\quad=\int_{\mathbb{T}^1}d\alpha\left(\int_{\mathbb{T}^1}dx\,\widetilde{\Phi}_1(\zeta_0(x,y,\alpha,\epsilon_1,\epsilon_2))\widetilde{\Phi}_1(\zeta_m(x,y,\alpha,\epsilon_1,\epsilon_2))\right.\\&\qquad\left.\times\int_{\mathbb{T}^1}dy\,\widetilde{\Phi}_2(\zeta_0(x,y,\alpha,\epsilon_1,\epsilon_2))\widetilde{\Phi}_2(\zeta_m(x,y,\alpha,\epsilon_1,\epsilon_2))\right).\end{aligned}\tag{4.64}$$

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確率測度 $\mathbb{P}^2\otimes(\delta_{-1}+\delta_1)/2\otimes(\delta_{-1}+\delta_1)/2$ の下で独立だから，式 (4.64) の値は

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

ここで，$\widetilde{\Phi}_3:=\widetilde{\Phi}_1\times\widetilde{\Phi}_2$ とおいた．系 4.33 によれば，$m\to\infty$ のとき

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

であり，この収束は指数関数的である．よって

$$\begin{aligned}&\sum_{m=1}^{\infty}\int_{\mathbb{T}^1}d\alpha\left(\int_{\mathbb{T}^1}dx\,\widetilde{\Phi}_1(\zeta_0(x,y,\alpha,\epsilon_1,\epsilon_2))\widetilde{\Phi}_1(\zeta_m(x,y,\alpha,\epsilon_1,\epsilon_2))\right)^2\\&\quad=\sum_{m=1}^{\infty}\sum_{i,j}\widetilde{\Phi}_3(i)\widetilde{\Phi}_3(j)p^m(i,j)\mu(F_i)\ <\ \infty\end{aligned}\tag{4.65}$$

だが，(4.65) 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

$$\sum_{m=1}^{\infty}\rho_1^{-m}\int_{\mathbb{T}^1}d\alpha\left(\int_{\mathbb{T}^1}dx\,\widetilde{\Phi}_1(\zeta_0(x,y,\alpha,\epsilon_1,\epsilon_2))\widetilde{\Phi}_1(\zeta_m(x,y,\alpha,\epsilon_1,\epsilon_2))\right)^2\ <\ \infty.\tag{4.66}$$

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

$$\rho_1^{-m/2} \int_{\mathbb{T}^1} dx \widetilde{\Phi}_1(\zeta_0(x,y,\alpha,\epsilon_1,\epsilon_2)) \widetilde{\Phi}_1(\zeta_m(x,y,\alpha,\epsilon_1,\epsilon_2)) \to 0, \quad m \to \infty, \qquad \text{a.e.}\alpha. \tag{4.67}$$

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 (4.58) ができることを示す. □

マルコフ性の証明

補題 4.32 の証明を行う．T_f が μ を保存するので $\{\zeta_m\}_{m=0}^{\infty}$ 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

$$T_f^{-m} := (T_f^m)^{-1}, \qquad \beta^{-m} := (\beta^m)^{-1} \tag{4.68}$$

と略記する．

補題 4.34 $T_f^{-m} F_j$，$j = 1,\dots,4J$，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.[注21] $T_f \tilde{C} \subset \tilde{C}$ 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 $A \cap T_f^{-m'} \tilde{C} \neq \emptyset$ となる．それで

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

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．$m \geq 2$ と $\mu(\zeta_0 = i_0, \dots, \zeta_{m-1} = i_{m-1}) > 0$ を仮定する．

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

[注21] C は (4.63) で定義された集合.

$T_f^{-m+1}F_{i_{m-1}}$ は 8^{m-1} 個の同測度の連結成分よりなり，補題 4.34 によれば，そのうちのいくつか，s 個としよう，が $F := F_{i_0} \cap T_f^{-1}F_{i_1} \cap \cdots \cap T_f^{-m+2}F_{i_{m-2}}$ にすっぽり含まれ，他の $8^{m-1} - s$ 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

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

これより，T_f の μ-不変性を用いて

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

が従い，$\{\zeta_m\}_{m=0}^{\infty}$ のマルコフ性が分かる． □

エルゴード性の証明

$\{\zeta_m\}_{m=0}^{\infty}$ 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 4.35 $\phi_i : \mathbb{T}^3 \to \mathbb{C}$, $i = 1, 2$, は可測関数で

$$\phi_1(x, y, \alpha) = \phi_1(2x, 2y, 2\alpha)f(x, \alpha), \quad \text{a.e.}, \tag{4.69}$$

$$\phi_2(x, y, \alpha) = \phi_2(2x, 2y, 2\alpha)f(x, \alpha)f(y, \alpha), \quad \text{a.e.}, \tag{4.70}$$

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) の符号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 $A \subset \mathbb{T}^3$ に注目する．

$$A := \left\{(x, y, \alpha) \;\middle|\; \begin{array}{lll} \frac{1}{2} < x < 1, & \frac{1}{2} < x + k_{l-2}\alpha < 1, & 1 < x + k_{l-1}\alpha < \frac{3}{2}, \\ \frac{1}{2} < y < 1, & \frac{1}{2} < y + k_{l-1}\alpha < 1 & \end{array}\right\}. \tag{4.71}$$

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

$$\begin{aligned} r_1(x) &= r_1(x + k_1\alpha) = \cdots = r_1(x + k_{l-2}\alpha) = -1, \quad r_1(x + k_{l-1}\alpha) = 1,\\ r_1(y) &= r_1(y + k_1\alpha) = \cdots = r_1(y + k_{l-1}\alpha) = -1, \end{aligned}$$

だから，l が偶数であることと関数 f の定義 (4.59) より

$$f(x,\alpha) \;=\; -1, \quad f(y,\alpha) = 1, \qquad (x,y,\alpha) \in A, \tag{4.72}$$

が分かる．(4.68) 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

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

は A の部分集合である．従って (4.72) 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 (4.69)(4.70) より

$$\phi_i(x,y,\alpha)\phi_i(2x,2y,2\alpha) = -1, \qquad (x,y,\alpha) \in B_0^{(-1)}, \tag{4.73}$$

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，(4.73) 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

$$\frac{1}{|A|}\int_A \phi_i(x,y,\alpha)dxdyd\alpha \;=:\; a_i \;\in\; (-1,1) \tag{4.74}$$

である．ここに $|A|$ は A のルベーグ測度．

次に各 m につき，補題 4.34 によって，$\beta^{-m}A$ の任意の連結成分 $B^{(-m)}$ 上で $f(x,\alpha)$ と $f(y,\alpha)$ は一定符号であることに注意しよう．従って (4.69)(4.70) より $B^{(-m)}$ 上では

$$\phi_i(x,y,\alpha)\phi_i(2x,2y,2\alpha) \;\equiv\; 1 \quad \text{または} \quad \phi_i(x,y,\alpha)\phi_i(2x,2y,2\alpha) \;\equiv\; -1. \tag{4.75}$$

そこで

$$\frac{1}{\left|B^{(-m)}\right|}\int_{B^{(-m)}} \phi_i(x,y,\alpha)dxdyd\alpha \;=\; \pm a_i \tag{4.76}$$

であることを帰納法で示そう．

まず，$m = 1$ のときは (4.75) と変数変換 $x' = 2x$, $y' = 2y$, $\alpha' = 2\alpha$ によって

$$\int_{B^{(-1)}} \phi_i(x,y,\alpha)dxdyd\alpha \;=\; \pm\int_{B^{(-1)}} \phi_i(2x,2y,2\alpha)dxdyd\alpha \;=\; \pm\frac{1}{8}\int_A \phi_i(x',y',\alpha')dx'dy'd\alpha' \tag{4.77}$$

である．$|B^{(-1)}| = \frac{1}{8}|A|$ だから

$$\frac{1}{\left|B^{(-1)}\right|}\int_{B^{(-1)}} \phi_i(x,y,\alpha)dxdyd\alpha \;=\; \pm a_i. \tag{4.78}$$

$m-1$ まで (4.76) が正しい仮定して m のときは (4.75) より (4.77) 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 (4.76) 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 $S\subset\mathbb{T}^3$ に対して

$$-1+\delta \;<\; \frac{1}{|S|}\int_S \phi_i(x,y,\alpha)dxdyd\alpha \;<\; 1-\delta. \tag{4.79}$$

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

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

これは (4.79) と矛盾する．従って (4.69)(4.70) 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 4.32 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

$$\phi(x,y,\alpha,\epsilon_1,\epsilon_2)=\phi(2x,2y,2\alpha,\epsilon_1 f(x,\alpha),\epsilon_2 f(y,\alpha)),\quad \mu\text{-a.e.}, \tag{4.80}$$

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

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

$\psi_2(2x,2y,2\alpha,-1)+\psi_2(2x,2y,2\alpha,1)=\psi_1(x,y,\alpha)\equiv 0$ だったから

$$\begin{aligned}\psi_2(x,y,\alpha,-1) &= \psi_2(2x,2y,2\alpha,-1)\left(\mathbf{1}_{\{f(x,\alpha)=1\}}-\mathbf{1}_{\{f(x,\alpha)=-1\}}\right)\\ &= \psi_2(2x,2y,2\alpha,-1)f(x,\alpha).\end{aligned}$$

補題 4.35 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 (4.80) 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 4.35 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 $\{\zeta_m\}_{m=0}^{\infty}$ の既約性の証明が完了した． □

非周期性の証明

補題 4.32 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．はじめに

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

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

とする．$F:=F'\times\{1\}\times\{1\}\subset\Omega$ 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 4.32 の証明，従って定理 4.11′(定理 4.11) の証明が完了した． □

4.3.5 2項間相関の指数的収束の精密評価

定理 4.11′(定理 4.11) における指数的収束の速度を表す ρ を評価しよう．2項間相関の場合には次のような定理がある．[注22]

定理 4.36 (cf. [47]) 任意の $\rho > \rho_0 := \sqrt{\left(1+\sqrt{17}\right)/8} = 0.80024\ldots$ について

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

実際にはもっと詳しく次の等式が成り立つ．

定理 4.37

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

定理 4.37 から定理 4.36 が従うことは前節の (4.67) から (4.66) が従うのと同様にして分かる．定理 4.36 における定数 ρ_0 は最良であって，これより小さくすることはできない．

定理 4.37 の証明. ここでは，漸化式の方法 ([47]) 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 4.37 を証明する．

$$a_1 = a_2 = \frac{1}{3} \tag{4.81}$$

$$a_{m+2} = \frac{1}{4}a_{m+1} + \frac{1}{4}a_m, \quad m\in\mathbb{N}^+ \tag{4.82}$$

(4.81) の証明. まず,

$$\int_{\mathbb{T}^1} r_1(x)r_1(x+\alpha)dx = |2-4\alpha| - 1 \tag{4.83}$$

より

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

[注22] 高信は特別な 4 項間相関の場合に指数 ρ を求めている．

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

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

(4.82) の証明. (4.29) 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．この節では以下，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 (4.82) を発見的に導く方法を与えよう．

補題 4.38

$$a_m = \frac{1}{3}\xi_m + \frac{1}{3}\eta_m, \tag{4.84}$$

$$\begin{pmatrix} \xi_{m+1} \\ \eta_{m+1} \end{pmatrix} = \begin{pmatrix} \frac{3}{4} & \frac{1}{2} \\ -\frac{1}{4} & -\frac{1}{2} \end{pmatrix} \begin{pmatrix} \xi_m \\ \eta_m \end{pmatrix}. \tag{4.85}$$

証明. 定理 4.13′ によれば

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

ここで $A(\bullet)$ は定義 4.20 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

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

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．$2^m\langle\alpha\rangle_m = \langle 2^m\alpha\rangle$ なので，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

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

となって (4.84) が示された．次に補題 4.21 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 (4.85) が示された． □

それでは (4.82) の証明を再開しよう．まず，(4.85) 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

$$\begin{aligned} a_{m+2} &= \frac{1}{3}\xi_{m+2} + \frac{1}{3}\eta_{m+2} \\ &= \frac{1}{3}\left(\frac{7}{16}\xi_m + \frac{1}{8}\eta_m\right) + \frac{1}{3}\left(-\frac{1}{16}\xi_m + \frac{1}{8}\eta_m\right) \\ &= \frac{1}{8}\xi_m + \frac{1}{12}\eta_m. \end{aligned} \tag{4.86}$$

また同様に

$$a_{m+1} = \frac{1}{3}\xi_{m+1} + \frac{1}{3}\eta_{m+1} = \frac{1}{6}\xi_m \tag{4.87}$$

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 (4.84)(4.86)(4.87) 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 (4.82) の証明，従って定理 4.37 の証明が完了した．□

第5章　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．

本章では，主にとても複雑な関数にモンテカルロ積分を適用する場合を想定して議論する．§ 5.1 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．これは 2 進数展開写像を通じて m 回の硬貨投げの関数と見ることもできる．一変数関数ではあるが，とても複雑で決定論的な数値積分法が適用でないが確率論的に意味のある例が存在する (例 5.8 や例 5.9)．この節では $2^m \gg 1$ 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．§ 5.2 では § 2.5.2 で導入した $\mathcal{B}_m$-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である RWS について，§ 2.5.2 では紹介しなかった重要な事項について述べる．§ 5.4 では任意の 2 乗可積分かつ模倣可能な確率変数 (§ 1.3) に対して有効な，ペアごとに独立なサンプルによるモンテカルロ積分法 (動的ランダム-ワイル-サンプリング) を紹介する．これは筆者が知る限り最も適用範囲の広い最も信頼性の高いモンテカルロ積分法である．

5.1　L^2-ロバスト性

§ 2.5 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.-サンプリングとランダム-ワイル-サンプリング (RWS, 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．

§ 2.5 で扱われた確率変数 X は $\{0,1\}^m$ 上の関数であったが，§ 1.1 で述べたように，それは 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．第 5 章を通じて，被積分関数は主として $\mathbb{T}^1$ 上の関数を扱う．

定理 5.1 (サンプリングに関する基本不等式 [44]) $\{\psi_l\}_{l=1}^{2^m-1}$ を $L^2(\mathcal{B}_m) := L^2(\mathbb{T}^1, \mathcal{B}_m, \mathbb{P})$ 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

$$\sum_{l=1}^{2^m-1} \mathbf{E}\left[\left|\frac{1}{N}\sum_{n=1}^{N}\psi_l(X_n)\right|^2\right] \geq \frac{2^m}{N} - 1, \qquad 1 \leq N \leq 2^m, \tag{5.1}$$

が成り立つ．

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

まず,

$$g(y) := \frac{2^m}{N}\sum_{n=1}^{N}\mathbf{1}_{[x_n,x_n+2^{-m})}(y) \tag{5.2}$$

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 $\{1, \psi_1, \ldots \psi_{2^m-1}\}$ 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

$$\|g\|^2_{L^2(\mathcal{B}_m)} = \langle g, 1\rangle^2_{L^2(\mathcal{B}_m)} + \sum_{l=1}^{2^m-1}\langle g, \psi_l\rangle^2_{L^2(\mathcal{B}_m)}. \tag{5.3}$$

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

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

を (5.3) 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 (5.1) が従う． □

系 5.2 ([44]) $2^m \geq N > 1$ とする．任意の確率変数列 $\{X_n\}_{n=1}^{2^m} \subset \mathbb{T}^1$ に対して，定数関数でない $f \in L^2(\mathcal{B}_m)$ が存在して次の不等式を満たす.

$$\mathbf{E}\left[\left|\frac{1}{N}\sum_{n=1}^{N} f(X_n) - \int_{\mathbb{T}^1} f(x)dx\right|^2\right] \geq \left(\frac{1}{N} - 2^{-m}\right)\mathbf{V}[f]. \tag{5.4}$$

定理 5.1 より，少なくとも一つの ψ_l は不等式 (5.4) を満たすから系 5.2 が従う．

系 5.2 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 5.1 によって必ず近似誤差が大きくなる被積分関数が — 不等式 (5.1) を満たすために — 存在するからである．これは “High risk, high return” な数値積分法といえる．

例 5.3 $n \in \mathbb{N}$ の 2 進数表示第 i 桁目を $d_{-i}(n)$ と書く．すなわち，$n = \sum_{i=1}^{\infty} d_{-i}(n)2^i$ (この和は実際には有限和)．このとき，

$$x_n := \sum_{i=0}^{\infty} d_{-i}(n)2^{-i-1}, \quad n \in \mathbb{N}^+,$$

で定義される数列 $\{x_n\}_{n=1}^{\infty} \subset \mathbb{T}^1$ を**ヴァン デル コルプト列** (van der Corput sequence, cf. [26]) という．具体的に書き下せば

$$\{x_n\}_{n=1}^{\infty} = \left\{\frac{1}{2}, \frac{1}{4}, \frac{3}{4}, \frac{1}{8}, \frac{5}{8}, \frac{3}{8}, \frac{7}{8}, \frac{1}{16}, \frac{9}{16}, \frac{5}{16}, \frac{13}{16}, \cdots\right\}.$$

この数列によるサンプリングの収束の速さに関しては，任意の有界変動関数 f に対して次の評価が成り立つことが知られている．

$$\left|\frac{1}{N}\sum_{n=1}^{N} f(x_n) - \int_0^1 f(t)dt\right| \le c(N)\|f\|_{BV} \times \frac{\log N}{N}, \quad N \ge 2,$$

ここに $\|f\|_{BV}$ は f の $\mathbb{T}^1$ 上の全変動ノルム，$c(N)$ は有界な係数で $c(N) = \log(N+1)/(\log 2 \cdot \log N)$ でよい ([26] 参照)．一般にこのような性質を持つ数列を小さい差異の一様分布列 (low discrepancy sequence) または準乱数という．

ヴァン デル コルプト列を用いた数値積分 (準モンテカルロ法，quasi-Monte Carlo method) は $\|f\|_{BV}$ が小さいときは，i.i.d.-サンプリングよりはるかに小さい誤差評価を持つが，$\|f\|_{BV}$ が大きいときは場合によっては i.i.d.-サンプリングよりずっと大きな誤差を生じることがある．実際，$f(x) = d_{30}(x)$ の場合だと

$$\frac{1}{2^{28}}\sum_{n=1}^{2^{28}} d_{30}(x_n) = 0$$

であり，これは $\int_0^1 d_{30}(t)dt = 1/2$ と大きく隔たっている．

“High risk, high return” な数値積分法とは逆に，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 5.4 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

$$\mathbf{E}\left[\left|\frac{1}{N}\sum_{n=1}^{N} f(X_n) - \int_{\mathbb{T}^1} f(x)dx\right|^2\right] \le \frac{\mathbf{V}[f]}{N}, \qquad 1 \le N \le 2^m, \tag{5.5}$$

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 $\{x_n\}_{n=1}^{2^m}$ を用いて (5.2) 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 (5.5) 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 5.1 と系 5.2 によれば，$2^m \gg N \gg 1$ のとき，不等式 (5.5) はこれ以上ほとんど改良されないことが分かる．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，RWS の場合も (5.5) が等式として成り立つから，L^2-ロバストである．その上，§ 2.5.2 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機能する．

5.2 ランダム-ワイル-サンプリング (その 2)

§ 2.5.2 で紹介したランダム-ワイル-サンプリング (RWS) について詳しい考察を行う．

5.2.1 中心極限定理のスケーリング極限の退化

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では正規分布に収束するのではなく，0 に確率収束することを示す (定理 5.6).

この目的のために，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 5.5 は定理 2.8 の連続版である．

定理 5.5 ([17, 52]) $(x, \alpha) \in \mathbb{T}^1 \times \mathbb{T}^1 = \mathbb{T}^2$ を $\mathbb{P}^2$ に従う確率変数とするとき，$\mathbb{T}^1$-値確率変数列 $\{x + n\alpha\}_{n\in\mathbb{Z}}$ は次の性質を持つ: $n \neq n'$ ならば $(x + n\alpha)$ と $(x + n'\alpha)$ は独立 (ペアごとに独立) であり，さらに各 $(x + n\alpha)$ は $\mathbb{T}^1$ 上に一様分布する．

証明. $\mathbb{T}^1$ 上の任意の有界ボレル関数 F, G に対して

$$\begin{aligned}
\int_{\mathbb{T}^1} d\alpha \int_{\mathbb{T}^1} dx\, F(x + n\alpha)G(x + n'\alpha) &= \int_{\mathbb{T}^1} d\alpha \int_{\mathbb{T}^1} dx\, F(x)G(x + (n' - n)\alpha) \\
&= \int_{\mathbb{T}^1} dx\, F(x) \int_{\mathbb{T}^1} d\alpha\, G(x + (n' - n)\alpha) \\
&= \int_{\mathbb{T}^1} dx\, F(x) \int_{\mathbb{T}^1} d\alpha\, G((n' - n)\alpha) \\
&= \int_{\mathbb{T}^1} dx\, F(x) \int_{\mathbb{T}^1} d\alpha\, G(\alpha). \qquad \square
\end{aligned}$$

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 ([26])．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，これは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 5.6 ([15, 52]) 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 $\varepsilon > 0$ について

$$\mathbb{P}^2\left(\left\{ (x,\alpha) \in \mathbb{T}^2 \;\middle|\; \left| \frac{1}{\sqrt{N}} \sum_{n=1}^{N} \left(F(x+n\alpha) - \int_{\mathbb{T}^1} F(y)dy \right) \right| > \varepsilon \right\} \right) \longrightarrow 0, \quad N \to \infty. \tag{5.6}$$

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. 一般性を失うことなく $\int_{\mathbb{T}^1} dx\, F(x) = 0$ と仮定してよい．$M \in \mathbb{N}^+$ に対して関数 $F_M : \mathbb{T}^1 \to \mathbb{R}$ 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

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

$\int_{\mathbb{T}^1} dtF(t) = 0$ 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 5.5 によって

$$\left\| \frac{1}{\sqrt{N}} \sum_{n=1}^{N} F(x+n\alpha) \right\|_p := \left(\iint_{\mathbb{T}^1\times\mathbb{T}^1} d\alpha dx \left| \frac{1}{\sqrt{N}} \sum_{n=1}^{N} F(x+n\alpha) \right|^p \right)^{\frac{1}{p}}$$

$$
\begin{aligned}
&\leq \left\| \frac{1}{\sqrt{N}} \sum_{n=1}^{N} F_M(x+n\alpha) \right\|_p + \left\| \frac{1}{\sqrt{N}} \sum_{n=1}^{N} (F-F_M)(x+n\alpha) \right\|_p \\
&\leq \left\| \frac{1}{\sqrt{N}} \sum_{n=1}^{N} F_M(x+n\alpha) \right\|_p + \left\| \frac{1}{\sqrt{N}} \sum_{n=1}^{N} (F-F_M)(x+n\alpha) \right\|_2 \\
&= \left\| \frac{1}{\sqrt{N}} \sum_{n=1}^{N} F_M(x+n\alpha) \right\|_p + \sqrt{\mathbf{V}[F-F_M]}. \qquad (5.7)
\end{aligned}
$$

(5.7) の最後の辺の第 1 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

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

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

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

□

注意 5.7 定理 5.6 は 2 乗可積分の多変数関数 $F : \mathbb{T}^k \to \mathbb{R}$ について成り立つ．詳しくは定理 5.20 を見よ．

数値積分の観点からは 0 への確率収束 (5.6) は中心極限定理よりもずっと望ましい．しかし用心深い人のために一つ注意を与えよう．RWS は

$$\int_{\mathbb{T}^2} \left| \frac{1}{N} \sum_{n=1}^{N} F(x+n\alpha) - \int_{\mathbb{T}^1} F(y)dy \right|^2 dxd\alpha = \frac{\mathbf{V}[F]}{N} \tag{5.8}$$

を満たす (cf. 定理 2.8) ので，もし，運悪く (5.6) 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．

定義 2.7 および定理 2.8 で述べた RWS の場合で考えよう．もし，$\alpha = 0 \in D_{m+j}$ と選んでしまったら，すべての n で $X_n(x,\alpha) = \lfloor x \rfloor_m \in D_m$ 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 2.11)．その確率は $2^{-(m+j)}$ 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．もっとも，そのような「とんでもなく悪いサンプリング点を生成してしまう確率」は，m がある程度大きければ，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．一方で，(5.8) 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．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ずっと適していると考えられるのである．

例 5.8 定理 5.6 の効果およびすぐ上の段落で述べたことを見るために，例 2.9 の $S_{10^6}(g(\omega'))/10^6$ の分布をモンテカルロ法で求めた．すなわち $\omega' = (x,\alpha) \in D_{119} \times D_{119}$ のサンプル 50,000 個を “ワイル変換による疑似乱数生成器” を用いて生成して，$S_{10^6}(g(\omega'))/10^6$ の度数分布について調べた．

表 5.1: RWS サンプルの平均と標準偏差

値の範囲	平均	標準偏差	$\mathcal{N}(0,1)$ の標準偏差
全体	0.546095	0.000434905	1.000000
中央 99.9%	0.546094	0.000307281	0.993631
中央 99%	0.546095	0.000262653	0.956823

サンプルの標準偏差の理論値はおよそ

$$\sqrt{\frac{0.546095 \cdot (1-0.546095)}{10^6}} = 0.000497871$$

である．これは表 5.1 の第 1 行目 (全体) の数値とおよそ合っている．表 5.1 の第 2 行目 (中央 99.9%) は，全サンプルのうち，小さい順に 25 個と大きい順に 25 個を除いた残り 99.9%のサンプルについて，その平均と標準偏差を計算したものである．この場合，平均は変わらないが標準偏差は全体の場合の約 3/4 に減少していることが分かる．このことは分布の両端の 0.1%のサンプルが平均から大きく隔たっていることを表している．第 3 行目 (中央 99%) は，全サンプルのうち，小さい順に 250 個と大きい順に 250 個を除いた残り 99%のサンプルについて同様に計算したものである．標準偏差が全体の場合の約 3/5 になっている．(同様の計算を標準正規分布 $\mathcal{N}(0,1)$ の場合に行った結果を表 5.1 の最右列に記した．)

図 5.1: $S_{10^6}(g(\omega'))/10^6$ の 50,000 個のサンプルの度数分布

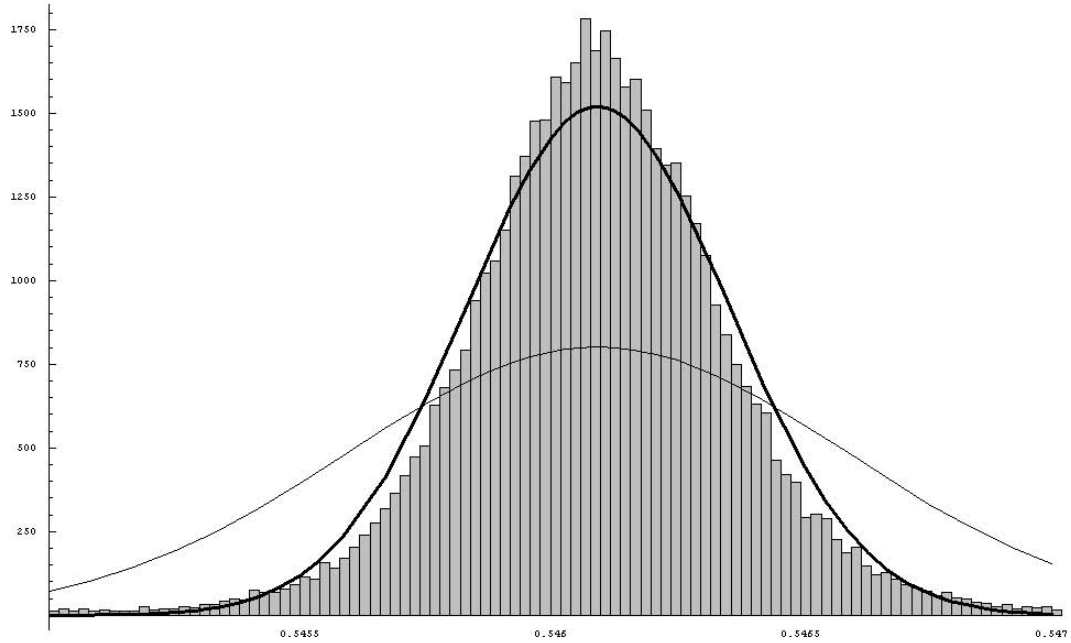


図 5.1 は $S_{10^6}(g(\omega'))/10^6$ の 50,000 個のサンプルの度数分布のヒストグラムである．この図で，太い曲線は平均 0.546095，標準偏差 0.000262653 (表 5.1 第 3 行目と同じ) の正規分布の密度関数である．これに比べて $S_{10^6}(g(\omega'))/10^6$ の分布は平均付近に集中する一方，裾野の分布が厚い．また，細い曲線は $S_{10^6}(\omega)/10^6$，すなわち i.i.d.-サンプリングの場合の分布と同じ平均・分散を持つ正規分布の密度関数である．明らかに i.i.d.-サンプリングに比べて RWS の方がずっと優れている．

5.2.2 $m \gg 1$ 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，アリスは種 $\omega' \in \{0,1\}^{2m+2j}$ を選ぶ．ここでもし，m が巨大だと，再び乱数の問題によって，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 2.12). このとき;

(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．

例 5.9 サイズの大きい二項分布を RWS を利用して数値的に求める．

$$S^{(m)}(x) := \sum_{i=1}^{m} d_i(x), \quad x \in \mathbb{T}^1,$$

とする．ルベーグ確率測度 $\mathbb{P}$ の下での $S^{(500)}$ の分布を計算するために，種 $(x,\alpha) \in D_{523} \times D_{523}$ に[注1]対して $S^{(500)}(x+n\alpha)$, $n = 1,2,\ldots,10^7$, をコンピュータで生成し，次の相対度数分布

$$p_k^{(500)}(N) := \frac{1}{N}\#\left\{1 \le n \le N \,\middle|\, S^{(500)}(x+n\alpha) = k\right\}, \quad k = 0,1,\ldots,500,$$

のヒストグラムを作成する (図 5.2). ここで種 (x,α) は 1046 ビットと長いので“ワイル変換による疑似乱数生成器”を用いて生成した (実装は § 6.1.2 を見よ).

N を大きくするとき，この相対度数分布は二項分布に近づく．実際，誤差

$$E_N := \sqrt{\sum_{k=0}^{500} \left|p_k^{(500)}(N) - q_k^{(500)}\right|^2}, \qquad q_k^{(500)} := \frac{500!}{(500-k)!k!} \times 2^{-500},$$

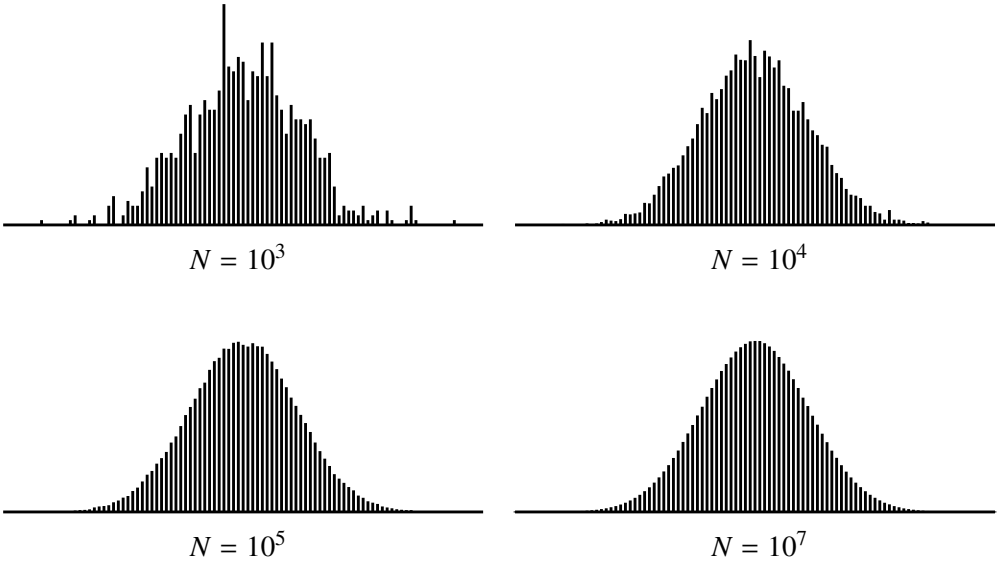
は $N \to \infty$ にともなって図 5.3 の黒い点列で示されるように小さくなっていく．横軸はサンプル数 N (最大 10^7) の対数，縦軸は誤差の対数である．図 5.3 に引かれた斜めの直線は i.i.d.-サンプリングの場合の理論値

$$\sqrt{\sum_{k=0}^{500} q_k^{(500)}(1-q_k^{(500)})} \times N^{-1/2} = 0.03122139 \times N^{-1/2}$$

のグラフである．この図から，RWS を $S^{(500)}$ の分布の計算に適用した場合，その誤差は i.i.d.-サンプリングの場合とほぼ同程度であることが見て取れる．

[注1] $523 = 500 + \lceil \log_2 10^7 \rceil - 1$.

図 5.2: 相対度数分布，$p_k^{(500)}(N)$,　$k = 0, \ldots, 500$.



定理 5.6 は RWS の収束は理論上 i.i.d.-サンプリングよりずっと速いことを主張する．しかし，一般に独立変数の数が多くそれらにほぼ均等に依存するような多変数関数に準モンテカルロ法を適用すると，その収束速度は i.i.d.-サンプリングの場合と変わらなくなる，といわれている (cf. [38] p.99)．そのような場合，サンプルサイズ N を天文学的に大きくしなければ RWS の優位が現われないのである．

例 5.9 の $S^{(m)}(x)$ の場合も各ビット $d_i(x)$, $i = 1, \ldots, m$, を独立変数と思えば，$S^{(m)}(x)$ は m 個の独立変数に均等に依存する関数である．実際，$S^{(m)}(x)$ の RWS によるサンプル列は $m \gg 1$ のとき i.i.d. に近いことが次の定理から分かる．

定理 5.10　([8])　ほとんどすべての無理数 α に対して，ルベーグ確率空間 $(\mathbb{T}^1, \mathcal{B}, \mathbb{P})$ 上の確率過程 $\{2m^{-1/2}(S^{(m)}(\bullet + n\alpha) - \frac{m}{2})\}_{n=0}^{\infty}$ は $m \to \infty$ のとき，標準正規分布に従う i.i.d. 確率変数列に有限次元分布の意味で収束する．

$m \to \infty$ のとき，この確率過程の各 1 次元分布は中心極限定理によって標準正規分布に収束する．確率変数列としてガウス系 (Gaussian system) に収束することを見るのは，すぐにはできないが，その代わり，ここでは相関が 0 に収束する仕組みを見よう．

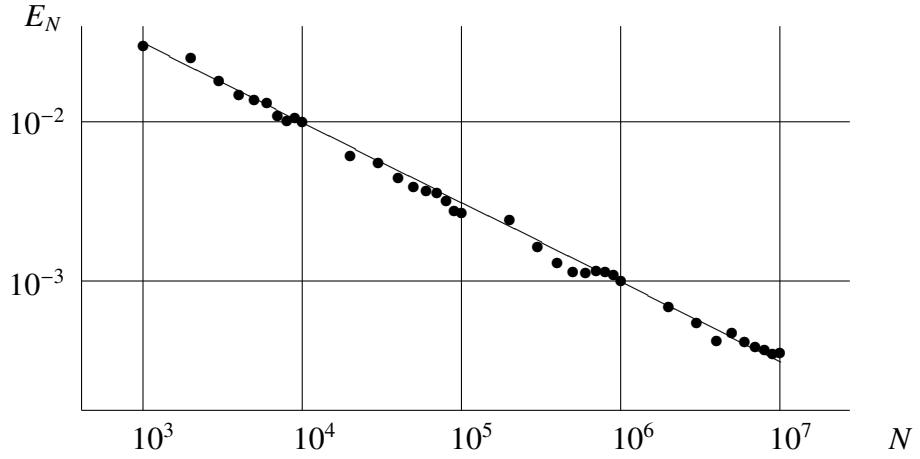
ラデマッハ関数列 $\{r_i(x) := 1 - 2d_i(x)\}_{i=1}^{\infty}$ の性質

$$r_i(x) \;=\; r_1(2^{i-1}x), \qquad \forall c, \quad \int_{\mathbb{T}^1} r_i(x) r_j(x+c)dx \;=\; 0, \quad i \neq j,$$

に注意して相関関数を計算する．関数 φ を

$$\varphi(\alpha) \;:=\; \int_{\mathbb{T}^1} r_1(x) r_1(x+\alpha)dx \;=\; |2 - 4\alpha| - 1$$

図 5.3: 誤差の減少 (log-log グラフ)



と定義しておく (cf. (4.83)). $S^{(m)}(x) - \frac{m}{2} = -\frac{1}{2}\sum_{i=1}^{m} r_i(x)$ だから

$$\begin{aligned}
&\int_{\mathbb{T}^1} \left(m^{-1/2}\left(S^{(m)}(x) - \frac{m}{2}\right)\right)\left(m^{-1/2}\left(S^{(m)}(x+n\alpha) - \frac{m}{2}\right)\right)dx \\
&\quad = \frac{1}{4m}\sum_{i=1}^{m}\sum_{j=1}^{m}\int_{\mathbb{T}^1} r_i(x)r_j(x+n\alpha)dx \\
&\quad = \frac{1}{4m}\sum_{i=1}^{m}\int_{\mathbb{T}^1} r_i(x)r_i(x+n\alpha)dx \\
&\quad = \frac{1}{4m}\sum_{i=1}^{m}\int_{\mathbb{T}^1} r_1(2^{i-1}x)r_1(2^{i-1}x+2^{i-1}n\alpha)dx \\
&\quad = \frac{1}{4m}\sum_{i=1}^{m}\int_{\mathbb{T}^1} r_1(x)r_1(x+2^{i-1}n\alpha)dx = \frac{1}{4m}\sum_{i=1}^{m}\varphi(2^{i-1}n\alpha). \qquad (5.9)
\end{aligned}$$

ここで最後の式は，変換 $x \mapsto 2x$ のエルゴード性のために，$m \to \infty$ のとき，式の値はほとんどすべての α に対して積分値 $\frac{1}{4}\int_{\mathbb{T}^1}\varphi(x)dx = 0$ に収束する ([19]). このように極限において従属性が消滅する.

もっとも，(5.9) の収束は遅く，ほぼ $O(m^{-1/2})$ 程度であることが分かる．これは中心極限定理および重複対数の法則が (5.9) のような和についても成り立つからである ([8, 19, 30]). じつは $m = 500$ くらいだと，$\{S^{(m)}(\bullet + n\alpha)\}_{n=0}^{\infty}$ の分布は i.i.d. にそれほど近いとはいえない．直感的には，m が大きいとき $S^{(m)}(x)$ と $S^{(m+1)}(x)$ は各 x ごとに (相対的に) あまり変わらないから，従属性の消滅は速くないのだろうと想像できる．ならば，多数個の独立変数に均等に依存する関数で，変数が一つ増えるたびに大きく値の変化するような関数列を考えればもっと速く従属性消滅が起こるだろう．たとえば関数 $G_m(x) = S^{(m)}(x) \mod 2$ (cf. (4.7)) は各変数 $d_i(x)$ に均等に依存するばかりでなく，どの一つの $d_i(x)$ の値が反転 (0 と 1 が入れ替わる) しても $G_m(x)$ の値も反転するから，もっと速く従属性消滅が起こる．これは定理 4.14 の現象を別の面から説明している.

—— ◇ —— ◇ ——

例 5.9 の計算を行ったコンピュータは Panasonic Let's note CF-Y5 (CPU 1.66GHz, RAM 1.49GB，HD 55.8GB) で，$\{p_k^{(500)}(10^7)\}_{k=0}^{500}$ の算出を 50 秒でやってのけた．[42] にある数値実験はこの例と同じ (ただし $x = \lfloor \pi - 3 \rfloor_{540}$, $\alpha = \lfloor (\sqrt{5}-1)/2 \rfloor_{540}$ として計算している) であるが，1993 年当時，筆者の研究室のコンピュータは NEC PC9801RA (CPU 20MHz，RAM 1.6MB，HD 40MB) で 19 時間を要した．この 15 年間でコンピュータの性能は速度，記憶容量とも約 1,000 倍になった．表 5.1 と図 5.1 のデータを作成するためのモンテカルロ積分の計算はもっと大規模であって，Let's note CF-Y5 をもってしても 24 時間かかった．PC9801RA なら 2 年 9ヶ月かかっていたであろう．

5.2.3　ペアごとに独立な確率変数列の他の例

GF(2^m) を位数 2^m の有限体とし，任意の二つの全単射 $\phi : \mathrm{GF}(2^m) \to \{0,1\}^m$ 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 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

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

とおくとき，P_{2m} の下で $\{\tilde{Z}_n\}_{n=1}^{2^m}$ はペアごとに独立であり，各 $\tilde{Z}_n$ は $\{0,1\}^m$ 上の一様分布に従う ([29] Lecture 5)．このことを見るには，任意の $a, b \in \mathrm{GF}(2^m) \cong \{0,1\}^m$, $1 \le n < n' \le 2^m$ に対して

$$P_{2m}(\tilde{Z}_n(\omega') = a,\, \tilde{Z}_{n'}(\omega') = b) \;=\; 2^{-2m}$$

となることを確かめればよい．実際，未知数 (x,α) に関する 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

$$P_{2m}(\tilde{Z}_n(\omega') = a,\, \tilde{Z}_{n'}(\omega') = b) \;=\; P_{2m}\left(\{(x_0',\alpha_0')\}\right) \;=\; 2^{-2m}.$$

定義 2.7 のように，疑似乱数生成器 $\tilde{g} : \{0,1\}^{2m} \to \{0,1\}^{Nm}$, $N \le 2^m$, を

$$\tilde{g}(\omega') := (\tilde{Z}_1(\omega'), \tilde{Z}_2(\omega'), \ldots, \tilde{Z}_N(\omega')) \in \mathrm{GF}(2^m)^N \cong \{0,1\}^{Nm}$$

とすれば，$(\{0,1\}^m, 2^{\{0,1\}^m}, P_m)$ 上の確率変数のモンテカルロ積分のための安全な疑似乱数生成器となる．

少なくとも二つ独立な D_m-値確率変数が存在するためには確率空間は $D_m \times D_m$ 以上の大きさが必要だから，$\tilde{g}$ の種の長さ $2m$ はペアごとに独立な複数個の D_m-値確率変数が存在するための最小の長さである．とくに RWS の場合の $2m + 2j$ より種が短くて済む．しかし GF(2^m) の演算が m が少し大きいと非常に大変なので実用には向かない．なお，[18] では GF(2^m) ではなく素体 F_p ですでに同様の議論によってペアごとに独立な確率変数列を構成している．[注2]ペアごとに独立な確率変数列によるサンプリング法に関しては [12, 36] に総合報告がある．

[注2]従って [18] の発表された 1974 年には，モンテカルロ積分のための安全な疑似乱数生成器の構成法が (それがそうだとは認識されないまま) 開発されていたことになる．

5.3 模倣可能な確率変数の i.i.d.-サンプリング

模倣可能な確率変数，すなわち，ある停止時刻に関して可測 (§ 1.3) な関数に対して 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 5.11 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 $\{f(y_n)\}_{n=1}^{\infty}$ 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．(1.7) より $f(y_n) = f(\lfloor y_n \rfloor_{\tau(y_n)})$ だから，$\{\lfloor y_n \rfloor_{\tau(y_n)}\}_{n=1}^{\infty}$ が i.i.d. であることを示めせばよい．[注3]

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 $\mathbb{P}\left(\lfloor y_i \rfloor_{\tau(y_i)} = a_i,\ 1 \le i \le n\right) = \prod_{i=1}^{n} \mathbb{P}\left(\lfloor x \rfloor_{\tau(x)} = a_i\right)$ を得る． □

定理 5.11 の仮定の下で

$$\frac{1}{N} \sum_{i=1}^{N} f(y_i(x)) \tag{5.10}$$

によって f の数値積分を行うことができる．

注意 5.12 (5.10) 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，すなわち $\mathbf{V}[\tau] = \infty$ の場合，実際の計算時間は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 1.11) 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．

[注3] 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．

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 (5.10) 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 f は仮定 1.9 を満たすとする．ただし，仮定 1.9 で登場する i.i.d. 確率変数列 $\{Z_1, Z_2, \ldots\}$ は以下では，計算精度 (2^{-K} とする) に合わせて $Z_l^{(K)} \in D_K$ 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 f のサンプルの総数を N とする．以下にそのアルゴリズムを示す.

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_f : Real;
begin
   S := 0.0;
   For i := 1 to N do
      begin
         Z :=Random_K;
         f を計算しようと試みる;
         while ( f を計算するためにさらに別の Z が必要 ) do
            begin
               Z :=Random_K;
               f を計算しようと試みる;
            end;      // f の計算が終了
         S := S + f ;
      end;
   result:= S/N;
end;
```

関数 Mean_of_f は変数 result に代入された値，すなわち，S/N を返す．ここで f のサンプルの生成に必要な Z はすべて関数 Random$_K$ により生成される．

5.4　動的ランダム-ワイル-サンプリング

ここでは，模倣可能な確率変数，すなわち，ある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-サンプリングは停止時刻に関して可測な確率変数のモンテカルロ積分専用の疑似乱数生成器と見なすことができる．

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 (§ 6.2，[51])，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 (§ 5.4.4)．

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 (§ 6.2.4)．

5.4.1 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

$$(x_l, \alpha_l) \in D_{K+j} \times D_{K+j} \subset \mathbb{T}^1 \times \mathbb{T}^1, \quad l \in \mathbb{N}^+, \tag{5.11}$$

を $D_{K+j} \times D_{K+j}$ 上の一様分布に従う i.i.d. 確率変数列とする．確率変数 $\mathbf{x}_n$ を

$$\mathbf{x}_n := \sum_{l=1}^{\infty} 2^{-(l-1)K} \lfloor x_l + \nu_{n,l}\, \alpha_l \rfloor_K \in \mathbb{T}^1, \quad n = 1, \dots, 2^{j+1}, \tag{5.12}$$

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:

$$\nu_{n,l} := \begin{cases} n & (l = 1), \\ \#\{\, 1 \le u \le n \mid \tau(\mathbf{x}_u) > (l-1)K \,\} & (l > 1). \end{cases} \tag{5.13}$$

τ が $\{\mathcal{B}_m\}_m$-停止時刻であることから，$\nu_{n,l}$ 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 5.13 ([45]) 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 $(\mathbb{T}^1, \mathcal{B}, \mathbb{P})$ 上の確率変数 f 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．定理 5.13 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．

$\mathbf{E}[\tau] < \infty$ と $f \in L^1(\mathcal{B}_\tau)$ を仮定する．このとき，$\mathbf{E}[f]$ の推定を

$$\frac{1}{N} \sum_{n=1}^{N} f(\mathbf{x}_n), \quad 1 \le N \le 2^{j+1},$$

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．[注4]

[注4]“動的” という形容詞はサンプル点列 $\{\mathbf{x}_n\}_{n=1}^{2^{j+1}}$ が同じプログラム (たとえば § 6.2 のもの) で生成されていても，それが適用される被積分関数 (詳しくは停止時間 τ) によって異なることを示している．

系 5.14 $f \in L^2(\mathcal{B}_\tau)$ 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，

$$\mathbf{E}\left[\left|\frac{1}{N}\sum_{n=0}^{N-1} f(\mathbf{x}_n) - \mathbf{E}[f]\right|^2\right] = \frac{\mathbf{V}[f]}{N}, \quad 1 \le N \le 2^{j+1}. \tag{5.14}$$

注意 5.15 注意 5.12 で述べたように，$\mathbf{E}[\tau^2] < \infty$ であることが望ましい．

5.4.2 定理 5.13 の証明

以下では $1 \le n < n' \le 2^{j+1}$ を満たす n, n' を任意に固定して考える．

$$m(n,n') := \max\{\, l \,|\, \nu_{n,l} < \nu_{n',l} \,\}$$

とおく．このとき $\nu_{n,l} < \nu_{n',l}$ より $i = 1, \dots, l$ に対して $\nu_{n,i} < \nu_{n',i}$ が従うから

$$m(n,n') = \max\{\, l \,|\, \nu_{n,i} < \nu_{n',i},\ i = 1, \dots, l \,\}. \tag{5.15}$$

そこで

$$\widetilde{\mathbf{x}}_{n'} := \sum_{l=1}^{\infty} 2^{-(l-1)K} \lfloor x_l + \widetilde{\nu}_{n',l}\, \alpha_l \rfloor_K, \quad \widetilde{\nu}_{n',l} := \begin{cases} \nu_{n',l} & (l \le m(n,n')), \\ n' & (l > m(n,n')), \end{cases} \tag{5.16}$$

とおく．

補題 5.16 (i) $\mathbf{x}_n$ は $\mathbb{T}^1$ 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，$2M$ 個の確率変数たち

$$\lfloor x_l + \nu_{n,l}\, \alpha_l \rfloor_K, \quad \lfloor x_l + \widetilde{\nu}_{n',l}\, \alpha_l \rfloor_K, \qquad l = 1, \dots, M, \tag{5.17}$$

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．

任意の $s_1, t_1, \dots, s_M, t_M \in D_K$ に対して

$$\begin{aligned}
&\Pr(\lfloor x_l + \nu_{n,l}\,\alpha_l \rfloor_K < s_l,\ \lfloor x_l + \widetilde{\nu}_{n',l}\,\alpha_l \rfloor_K < t_l,\ l = 1, \dots, M) \\
&= \sum_{p<p'} \Pr\left(\begin{array}{l} \lfloor x_l + \nu_{n,l}\alpha_l \rfloor_K < s_l, \\ \lfloor x_l + \widetilde{\nu}_{n',l}\alpha_l \rfloor_K < t_l, \end{array} \ l = 1, \dots, M-1,\ \begin{array}{l} \nu_{n,M} = p, \\ \widetilde{\nu}_{n,M} = p', \end{array} \ \begin{array}{l} \lfloor x_M + p\alpha_M \rfloor_K < s_M \\ \lfloor x_M + p'\alpha_M \rfloor_K < t_M \end{array} \right) \\
&= \sum_{p<p'} \Pr\left(\begin{array}{l} \lfloor x_l + \nu_{n,l}\alpha_l \rfloor_K < s_l, \\ \lfloor x_l + \widetilde{\nu}_{n',l}\alpha_l \rfloor_K < t_l, \end{array} \ l = 1, \dots, M-1,\ \begin{array}{l} \nu_{n,M} = p \\ \widetilde{\nu}_{n',M} = p' \end{array} \right) \\
&\qquad \times P\left(\begin{array}{l} \lfloor x_M + p\alpha_M \rfloor_K < s_M \\ \lfloor x_M + p'\alpha_M \rfloor_K < t_M \end{array} \right).
\end{aligned}$$

$p \neq p'$ だから定理 2.8 の証明により，事象 $\{\lfloor x_M + p\alpha_M \rfloor_K < s_M\}$ と $\{\lfloor x_M + p'\alpha_M \rfloor_K < t_M\}$ は独立である．よって

$$\begin{aligned}
&= \sum_{p<p'} \Pr\left(\begin{array}{l} \lfloor x_l + \nu_{n,l}\alpha_l \rfloor_K < s_l, \\ \lfloor x_l + \widetilde{\nu}_{n',l}\alpha_l \rfloor_K < t_l, \end{array} \; l = 1, \dots, M-1, \; \begin{array}{l} \nu_{n,M} = p \\ \widetilde{\nu}_{n',M} = p' \end{array} \right) \\
&\qquad \times \Pr\left(\lfloor x_M + p\alpha_M \rfloor_K < s_M\right) \Pr\left(\lfloor x_M + p'\alpha_M \rfloor_K < t_M\right) \\
&= \sum_{p<p'} \Pr\left(\begin{array}{l} \lfloor x_l + \nu_{n,l}\alpha_l \rfloor_K < s_l, \\ \lfloor x_l + \widetilde{\nu}_{n',l}\alpha_l \rfloor_K < t_l, \end{array} \; l = 1, \dots, M-1, \; \begin{array}{l} \nu_{n,M} = p \\ \widetilde{\nu}_{n',M} = p' \end{array} \right) \times s_M t_M \\
&= \Pr\left(\lfloor x_l + \nu_{n,l}\,\alpha_l \rfloor_K < s_l, \lfloor x_l + \widetilde{\nu}_{n',l}\,\alpha_l \rfloor_K < t_l, \, l = 1, \dots, M-1\right) \times s_M t_M.
\end{aligned}$$

この操作を続ければ，最終的に

$$\Pr\left(\lfloor x_l + \nu_{n,l}\,\alpha_l \rfloor_K < s_l, \lfloor x_l + \widetilde{\nu}_{n',l}\,\alpha_l \rfloor_K < t_l, \, l = 1, \dots, M\right) = \prod_{i=1}^{M} s_i t_i,$$

となって補題の主張を得る． □

定理 5.13 の証明. 補題 5.16(i) により，$f(\mathbf{x}_n)$ と f は同分布である．次に，(5.12) で n を n' で置き換えたものと (5.16) によって

$$\lfloor \mathbf{x}_{n'} \rfloor_{m(n,n')K} = \lfloor \widetilde{\mathbf{x}}_{n'} \rfloor_{m(n,n')K}. \tag{5.18}$$

$s := \lceil \tau(\mathbf{x}_{n'})/K \rceil$ 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，(5.15) より

$$s \leq m(n, n'). \tag{5.19}$$

(5.18) と (5.19) から

$$\lfloor \mathbf{x}_{n'} \rfloor_{sK} = \lfloor \widetilde{\mathbf{x}}_{n'} \rfloor_{sK} \tag{5.20}$$

が従う．一方，$\tau(\mathbf{x}_{n'}) \leq sK$ 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．つまり $\tau(\mathbf{x}_{n'}) = \tau(\lfloor \mathbf{x}_{n'} \rfloor_{sK})$. すると (5.20) より

$$\tau(\widetilde{\mathbf{x}}_{n'}) = \tau(\mathbf{x}_{n'}) \leq sK \tag{5.21}$$

でなければならない．よって，(5.20) と (5.21) より

$$\lfloor \mathbf{x}_{n'} \rfloor_{\tau(\mathbf{x}_{n'})} = \lfloor \widetilde{\mathbf{x}}_{n'} \rfloor_{\tau(\widetilde{\mathbf{x}}_{n'})}. \tag{5.22}$$

f は τ-可測だから，(1.7) と (5.22) より，$f(\mathbf{x}_{n'}) = f(\widetilde{\mathbf{x}}_{n'})$ である．最後に補題 5.16(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． □

5.4.3 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．§ 5.3 での設定を踏襲する．さらに，ここでは次を仮定する．

$$1 \leq N \leq 2^{j+1}. \tag{5.23}$$

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;

- 手続き，関数

```
procedure Set_First_Location;
begin
   l := 0;
end;
```

```
function Drws : D_K-valued;
begin
   l := l + 1;
   if (x_l, α_l) が生成されていなかったら
      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_f : Real;
begin
   S := 0.0;
   For i := 1 to N do
      begin
         Set_First_Location;
         Z :=Drws;
         f を計算しようと試みる;
         while ( f を計算するためにさらに別の Z が必要 ) do
            begin
               Z :=Drws;
               f を計算しようと試みる;
            end;      // f の計算が終了
         S := S + f;
      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 (§ 5.3) 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．違いは次の点だけである; DRWS では関数 Random_K を直接呼んで Z を生成するのはなく，関数 Drws を呼んで生成する．ただし，f の各サンプルの生成を始める前に手続き Set_First_Location を呼び出しておく．

ランダムな関数 Random_{K+j} は，(x_l, 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だけ Drws に呼び出される．そのため DRWS ではランダムな関数を呼び出す回数が i.i.d.-サンプリングの場合よりもずっと少ない．

注意 5.17 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, α_l) たちは，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多くの場合面倒なので，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．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，§ 5.2.2 でのべたことと同じことが DRWS についてもいえる．[注5]

注意 5.18 確率変数 f 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) たちを記憶しておくために，DRWS はコンピュータのメモリを使い果たすかもしれない．具体的な実装におけるメモリの問題の解決については § 6.2.4 を見よ．

5.4.4 i.i.d.-サンプリングと DRWS の性能比較

例 1.12 の確率変数 f の平均を DRWS を用いて具体的に求めてみた．DRWS によって f 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．[注6]確率変数 f の平均と分散は共に 10 である．[注7]

表 5.2 は， DRWS による計算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，[16] 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 (§ 6.2 の `m90randombit()` を用いたもの)) についても同様の計算を行った．[注8]誤差を比較すると DRWS の著しい優位が見える．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．

[注5] § 6.2 での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．

[注6] 具体的には § 6.2.3 のサンプルプログラム `drws.c` において `#define SAMPLE_SIZE` を 10^3 から 10^8 まで変えて計算した．

[注7] ワルドの等式 ([7] (1.6) Wald's equation, p.179 および Exercise 1.15, P.182) から従う．または負の二項分布を用いてもよい．

[注8] ノートパソコン Panasonic Let's Note CF-Y5 (CPU1.66GHz，RAM 1.49GB，HD55.8GB) 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 5.2: 誤差の比較

サンプル数	rand-i.i.d.	MT-i.i.d.	m90-i.i.d.	DRWS
10^3	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

表 5.3: DRWS のサンプルの平均と標準偏差

値の範囲	平均	標準偏差
全体	9.99993	0.003129874
中央 99.9%	9.99999	0.000601414
中央 99%	10.00000	0.000231709

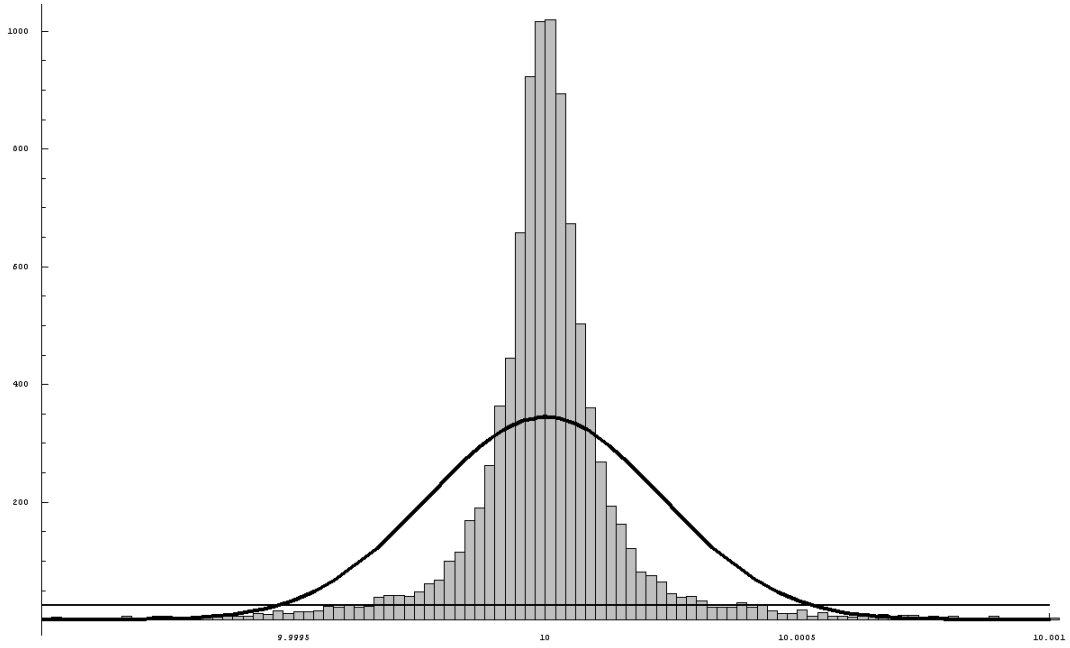
例 5.8 にならって，f に対してサンプル数 10^6 の DRWS を 10,000 回ランダムに適用したときの度数分布を調べた (表 5.3)．サンプルの標準偏差の理論値は

$$\sqrt{\frac{10}{10^6}} = \sqrt{10^{-5}} = 0.00316228$$

である．これは表 5.3 の第 1 行目 (全体) の数値と合っている．表 5.3 の第 2 行目 (中央 99.9%) は，全サンプルのうち，小さい順に 5 個と大きい順に 5 個を除いた残り 99.9%のサンプルについて，その平均と標準偏差を計算したものである．この場合，平均は変わらないが標準偏差は全体の場合の約 1/5 に減少していることが分かる．このことは分布の両端の 0.1%のサンプルが平均から非常に隔たっていることを表している．第 3 行目 (中央 99%) は，全サンプルのうち，小さい順に 50 個と大きい順に 50 個を除いた残り 99%のサンプルについて同様に計算したものである．標準偏差が全体の場合の約 1/13.6 になっている．このように DRWS でも RWS の場合の例 5.8 と同様の現象が起こっていることが分かる．

図 5.4 はこの度数分布をヒストグラムで表したものである．この図で，太い曲線は平均 10，標準偏差 0.000231709 (表 5.3 第 3 行目と同じ) の正規分布の密度関数である．これに比べて DRWS のサンプルの分布は平均付近に集中する一方，裾野の分布が厚い．また，図の低い位置にあるほぼ平坦に見える細い曲線は i.i.d.-サンプリングの場合の分布と同じ平均・分散を持つ正規分布，すなわち $\mathcal{N}(10, 10^{-5})$ の密度関数である．

図 5.4: DRWS のサンプルの度数分布



5.4.5 DRWS の収束に関する極限定理

前節で見た DRWS の計算誤差は高い確率で非常に小さい．その原因は定理 5.6 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．実際，特別な場合，たとえば停止時刻が定数の場合や，フォン・ノイマンの棄却法で生成される確率変数の場合などには，以下に述べるようにそれは正しいことが分かる．

そのような極限定理を述べるために DRWS の定式化を少し変更して議論する．ルベーグ確率空間 $(\mathbb{T}^1, \mathcal{B}, \mathbb{P})$ の可算無限直積を $(\mathbb{T}^\infty, \mathcal{B}^\infty, \mathbb{P}^\infty)$ と書く．$\mathcal{B}^\infty$ の部分完全加法族の増大列 $\{\mathcal{B}^m\}_{m=1}^\infty$ を

$$\mathcal{B}^m := \sigma(Z_1, Z_2, \ldots, Z_m), \quad m = 1, 2, \ldots,$$

で定義する．ここに $Z_i : \mathbb{T}^\infty \to \mathbb{T}^1$ は座標関数 (射影) で

$$Z_i(x) := x_i, \quad x = (x_1, x_2, \ldots) \in \mathbb{T}^\infty.$$

関数 $\tau : \mathbb{T}^\infty \to \mathbb{N}^+ \cup \{\infty\}$ が $\{\mathcal{B}^m\}_m$-停止時刻であるとは

$$\forall m \in \mathbb{N}^+, \quad \{\tau \le m\} \in \mathcal{B}^m,$$

であるこという．以下，$\mathbb{P}^\infty(\tau < \infty) = 1$ を仮定する．また，$\{\mathcal{B}^m\}_m$-停止時刻 τ に対して次の部分完全加法族

$$\mathcal{B}^\tau := \{ A \in \mathcal{B}^\infty \,|\, \forall m \in \mathbb{N}^+, A \cap \{\tau \le m\} \in \mathcal{B}^m \}$$

を定義する．$\mathcal{B}^\tau$-可測関数を単に τ-可測関数と呼ぶ．

さて，直積確率空間 $(\mathbb{T}^\infty \times \mathbb{T}^\infty, \mathcal{B}^\infty \otimes \mathcal{B}^\infty, \mathbb{P}^\infty \otimes \mathbb{P}^\infty)$ 上で確率変数列 $\{\mathbf{x}_n\}_{n=1}^\infty$ を

$$\begin{aligned} \mathbf{x}_n(x,\alpha) &:= (x_1 + \nu_{n,1}\alpha_1, x_2 + \nu_{n,2}\alpha_2, \ldots) \in \mathbb{T}^\infty, \\ & \quad x = (x_1, x_2, \ldots),\ \alpha = (\alpha_1, \alpha_2, \ldots) \in \mathbb{T}^\infty, \end{aligned} \tag{5.24}$$

で定義する，ここに

$$\nu_{n,l} := \begin{cases} n & (l = 1), \\ \#\{1 \le u \le n \,|\, \tau(\mathbf{x}_u) > l - 1\} & (l > 1). \end{cases} \tag{5.25}$$

とする．τ は $\{\mathcal{B}_m\}_m$-停止時刻であるから $\nu_{n,l}$ と $\mathbf{x}_n$ は確かに定義される (well-defined).

定理 5.19 τ を $\{\mathcal{B}_m\}_m$-停止時刻，$f : \mathbb{T}^\infty \to \mathbb{R}$ を τ-可測関数とする．このとき，確率変数列 $\{f(\mathbf{x}_n)\}_{n=1}^\infty$ は同分布列かつペアごとに独立であり，その共通の分布は $(\mathbb{T}^\infty, \mathcal{B}^\infty, \mathbb{P}^\infty)$ 上の確率変数 f の分布に等しい．

この定理の証明は定理 5.13 とほぼ同様であるから省略する．もちろん今の場合も系 5.14 と同じ主張が成り立つ．

まず，$\tau = k$ (定数) の場合を考えよう．このとき

$$\nu_{n,l} = n, \quad n = 1, 2, \ldots, \quad l = 1, 2, \ldots, k,$$

であり，f が τ-可測関数とはそれが $\mathcal{B}^k$-可測関数のことであり，すなわち，実質的に $\mathbb{T}^k$ 上の関数ということである．その場合には次の定理が成り立つ．

定理 5.20 (cf. 注意 5.7) $F \in L^2(\mathbb{T}^k, \mathcal{B}^k, \mathbb{P}^k)$ と $1 \le p < 2$ に対して，

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

従って，任意の $\varepsilon > 0$ について

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

証明. F の k 次元フーリエ級数展開を利用して定理 5.6 の場合と同じように進めると，定理 5.20 は

$$\int_{\mathbb{T}^k} \left| \frac{1}{\sqrt{N}} \sum_{n=1}^N e^{2\sqrt{-1}\pi n(l_1\alpha_1 + \cdots + l_k\alpha_k)} \right|^p d\alpha_1 \cdots d\alpha_k \to 0, \quad N \to \infty,$$

を示すことに帰着される．ここで $l_1, \ldots, l_k \in \mathbb{Z}$ であるが，少なくともどれか一つは 0 ではない．$l_i = 0$ の場合は α_i に関する積分が省かれて 1 次元低い積分になる．こ

の操作を繰り返せば，やがてすべての l_i が 0 でない場合に帰着される．それで，ここではすべての l_i が 0 でないとして証明を続けよう．

このとき変換

$$\mathbb{T}^k \ni (\alpha_1, \dots, \alpha_k) \mapsto (l_1\alpha, \dots, l_k\alpha_k) \in \mathbb{T}^k$$

はルベーグ測度 $\mathbb{P}^k$ を保存する．そのため

$$\int_{\mathbb{T}^k} \left| \frac{1}{\sqrt{N}} \sum_{n=1}^{N} e^{2\sqrt{-1}\pi n(l_1\alpha_1+\cdots+l_k\alpha_k)} \right|^p d\alpha_1 \cdots d\alpha_k = \int_{\mathbb{T}^k} \left| \frac{1}{\sqrt{N}} \sum_{n=1}^{N} e^{2\sqrt{-1}\pi n(\alpha_1+\cdots+\alpha_k)} \right|^p d\alpha_1 \cdots d\alpha_k$$

である．ここで一般に有界可測関数 $f : \mathbb{T}^1 \to \mathbb{R}$ に対して次が成り立つ．

$$\int_{\mathbb{T}^1} \int_{\mathbb{T}^1} f(y_1 + y_2) dy_1 dy_2 = \int_{\mathbb{T}^1} f(y) dy.$$

だから

$$\begin{aligned}
&\int_{\mathbb{T}^k} \left| \frac{1}{\sqrt{N}} \sum_{n=1}^{N} e^{2\sqrt{-1}\pi n(\alpha_1+\cdots+\alpha_k)} \right|^p d\alpha_1 \cdots d\alpha_k \\
&\quad = \int_{\mathbb{T}^{k-2}} d\alpha_1 \cdots d\alpha_{k-2} \int_{\mathbb{T}^2} d\alpha_{k-1} d\alpha_k \left| \frac{1}{\sqrt{N}} \sum_{n=1}^{N} e^{2\sqrt{-1}\pi n(\alpha_1+\cdots+\alpha_{k-2}+\alpha_{k-1}+\alpha_k)} \right|^p \\
&\quad = \int_{\mathbb{T}^{k-1}} \left| \frac{1}{\sqrt{N}} \sum_{n=1}^{N} e^{2\sqrt{-1}\pi n(\alpha_1+\cdots+\alpha_{k-1})} \right|^p d\alpha_1 \cdots d\alpha_{k-1} \\
&\quad = \cdots \\
&\quad = \int_{\mathbb{T}^1} \left| \frac{1}{\sqrt{N}} \sum_{n=1}^{N} e^{2\sqrt{-1}\pi n\alpha_1} \right|^p d\alpha_1.
\end{aligned}$$

最後の式が $N \to \infty$ のときに 0 に収束するのは定理 5.6 の証明で見た通りである．□

それでは次に，フォン・ノイマンの棄却法 (例 1.11) で生成される確率変数に DRWS を適用する場合を考えよう．被積分関数 f に関する次の仮定を設ける．

仮定 5.21 関数 $f : \Omega \to \mathbb{R}$ に対して，ある $\{\mathcal{B}^m\}_m$-停止時刻 τ と $r \in \mathbb{N}^+$ が存在して，
(i) f は τ-可測，
(ii) $\mathbb{P}^\infty(\tau \in r\,\mathbb{N}^+) = 1$．(このとき，$\nu_{n,(k-1)r+1} = \cdots = \nu_{n,kr-1} = \nu_{n,kr}$，$k = 1, 2, \dots$)，
(iii) 任意の $k \in \mathbb{N}^+$ について条件 $\{\tau \geq kr\}$ の下で $\{\tau = kr\}$ は $\mathcal{B}^{(k-1)r}$ と独立，
(iv) 任意の $k \in \mathbb{N}^+$，$s \in \mathbb{R}$ について条件 $\{\tau \geq kr\}$ の下で $\{f \leq s\}$ は $\mathcal{B}^{(k-1)r}$ と独立．

例 5.22 (cf. 例 1.11) $0 \leq p(t) \leq M$ ($M > 0$ は定数) を区間 $[a, b]$ 上の確率密度関数とする．$\{\mathcal{B}^m\}_m$-停止時刻 τ を

$$\tau := \inf\{\, 2l \in 2\mathbb{N}^+ \mid p\,((b-a)Z_{2l-1} + a) \geq MZ_{2l} \,\}$$

で定義する．このとき

$$f(x) := (b-a)Z_{\tau(x)-1}(x) + a, \quad x \in \mathbb{T}^\infty,$$

とすれば，f は上の τ とともに $r = 2$ として仮定 5.21 を満たす．f の分布は密度 $p(t)$ を持つ．

定理 5.23 f は仮定 5.21 を満たす 2 乗可積分関数とし，$\{\mathbf{x}_n\}_n$ は (5.24)(5.25) で定義された確率変数列とする．このとき

$$\forall \varepsilon > 0, \quad \lim_{N\to\infty} \mathbb{P}^\infty \otimes \mathbb{P}^\infty \left(\left| \frac{1}{\sqrt{N}} \sum_{n=1}^{N} (f(\mathbf{x}_n) - \mathbf{E}[f]) \right| > \varepsilon \right) = 0.$$

定理の証明のために補題を準備する．以下，f は τ とともに $r = 2$ として仮定 5.21 を満たすとする．(一般の r でも同様にして証明できる．)

補題 5.24 $l \in \mathbb{N}^+$ と 2 乗可積分関数 $g : \mathbb{T}^2 \to \mathbb{R}$ に対して

$$h_l(x) := \mathbf{1}_{\{\tau \geq 2l\}}(x)\, g(Z_{2l-1}(x), Z_{2l}(x)), \quad x \in \mathbb{T}^\infty,$$

とする．このとき任意の $\varepsilon > 0$ について

$$\lim_{N\to\infty} \mathbb{P}^\infty \otimes \mathbb{P}^\infty \left(\left| \frac{1}{\sqrt{N}} \sum_{n=1}^{N} (h_l(\mathbf{x}_n) - \mathbf{E}[h_l]) \right| > \varepsilon \right) = 0.$$

証明. 数学的帰納法で示す．まず $l = 1$ のときは $h_1(x) = g(Z_1(x), Z_2(x))$ だから，

$$h_1(\mathbf{x}_n) = g(x_1 + n\alpha_1, x_2 + n\alpha_2), \quad n = 1, 2, \ldots,$$

となり，定理 5.20 により補題の主張は正しい．

つぎに $l \geq 2$ の場合を考える．このとき $\mathbf{1}_{\{\tau\geq 2l\}}(x) = \mathbf{1}_{\{\tau\leq 2l-2\}^c}(x)$ は $\mathcal{B}^{2l-2}$-可測，$g(X_{2l-1}(x), X_{2l}(x))$ は $\sigma(X_{2l-1}, X_{2l})$-可測だから，これらは独立で次が成り立つ．

$$\mathbf{E}[h_l] = q_l\, \mathbf{E}[g], \quad q_l := \mathbb{P}^\infty(\tau \geq 2l).$$

さて，$l = 2$ のときは ($\nu_{n,3} = \nu_{n,4}$ に注意して)

$$h_2(\mathbf{x}_n) = \mathbf{1}_{\{\tau\geq 4\}}(\mathbf{x}_n) g(x_3 + \nu_{n,3}\alpha_3, x_4 + \nu_{n,3}\alpha_4), \quad n = 1, 2, \ldots,$$

であるから

$$\frac{1}{\sqrt{N}} \sum_{n=1}^{N} h_2(\mathbf{x}_n) = \frac{1}{\sqrt{N}} \sum_{m=1}^{\nu_{N,3}} g(x_3 + m\alpha_3, x_4 + m\alpha_4).$$

これより次の不等式を得る．

$$\begin{aligned}
\left| \frac{1}{\sqrt{N}} \sum_{n=1}^{N} (h_2(\mathbf{x}_n) - \mathbf{E}[h_2]) \right| &= \left| \frac{1}{\sqrt{N}} \sum_{n=1}^{N} (h_2(\mathbf{x}_n) - q_2 \mathbf{E}[g]) \right| \\
&\leq \left| \sqrt{\frac{\nu_{N,3}}{N}} \cdot \frac{1}{\sqrt{\nu_{N,3}}} \sum_{m=1}^{\nu_{N,3}} (g(x_3 + m\alpha_3, x_4 + m\alpha_4) - \mathbf{E}[g]) \right| \\
&\quad + \left| \frac{\nu_{N,3}}{\sqrt{N}} \mathbf{E}[g] - \sqrt{N} q_2 \mathbf{E}[g] \right| \\
&\leq \left| \frac{1}{\sqrt{\nu_{N,3}}} \sum_{m=1}^{\nu_{N,3}} (g(x_3 + m\alpha_3, x_4 + m\alpha_4) - \mathbf{E}[g]) \right| \\
&\quad + \frac{\mathbf{E}[|g|]}{\sqrt{N}} \left| \nu_{N,3} - N q_2 \right|.
\end{aligned}$$

これより

$$\begin{aligned}
&\mathbb{P}^\infty \otimes \mathbb{P}^\infty \left(\left| \frac{1}{\sqrt{N}} \sum_{n=1}^{N} (h_2(\mathbf{x}_n) - \mathbf{E}[h_2]) \right| > \varepsilon \right) \\
&\quad \leq \quad \mathbb{P}^\infty \otimes \mathbb{P}^\infty \left(\left| \frac{1}{\sqrt{\nu_{N,3}}} \sum_{m=1}^{\nu_{N,3}} (g(x_3 + m\alpha_3, x_4 + m\alpha_4) - \mathbf{E}[g]) \right| > \frac{\varepsilon}{2} \right) \\
&\qquad + \mathbb{P}^\infty \otimes \mathbb{P}^\infty \left(\frac{\mathbf{E}[|g|]}{\sqrt{N}} \left| \nu_{N,3} - Nq_2 \right| > \frac{\varepsilon}{2} \right) \\
&\quad =: \quad I_1 + I_2. \qquad (5.26)
\end{aligned}$$

任意の $\delta > 0$ をとる．まず，定理 5.20 により，ある $K_0 \in \mathbb{N}^+$ が存在し，任意の $K \geq K_0$ に対して

$$\mathbb{P}^\infty \otimes \mathbb{P}^\infty \left(\left| \frac{1}{\sqrt{K}} \sum_{m=1}^{K} (g(x_3 + m\alpha_3, x_4 + m\alpha_4) - \mathbf{E}[g]) \right| > \frac{\varepsilon}{2} \right) < \frac{\delta}{3}$$

が成り立つ．そこで

$$\begin{aligned}
&\mathbb{P}^\infty \otimes \mathbb{P}^\infty \left(\left| \frac{1}{\sqrt{\nu_{N,3}}} \sum_{m=1}^{\nu_{N,3}} (g(x_3 + m\alpha_3, x_4 + m\alpha_4) - \mathbf{E}[g]) \right| > \frac{\varepsilon}{2},\ \nu_{N,3} \geq K_0 \right) \\
&= \quad \sum_{K=K_0}^{\infty} \mathbb{P}^\infty \otimes \mathbb{P}^\infty \left(\left| \frac{1}{\sqrt{K}} \sum_{m=1}^{K} (g(x_3 + m\alpha_3, x_4 + m\alpha_4) - \mathbf{E}[g]) \right| > \frac{\varepsilon}{2},\ \nu_{N,3} = K \right) \\
&= \quad \sum_{K=K_0}^{\infty} \mathbb{P}^\infty \otimes \mathbb{P}^\infty \left(\left| \frac{1}{\sqrt{K}} \sum_{m=1}^{K} (g(x_3 + m\alpha_3, x_4 + m\alpha_4) - \mathbf{E}[g]) \right| > \frac{\varepsilon}{2} \right) \mathbb{P}^\infty \otimes \mathbb{P}^\infty \left(\nu_{N,3} = K \right) \\
&\leq \quad \frac{\delta}{3} \mathbb{P}^\infty \otimes \mathbb{P}^\infty \left(\nu_{N,3} \geq K_0 \right) \\
&< \quad \frac{\delta}{3}. \qquad (5.27)
\end{aligned}$$

一方，$\nu_{N,3} = \sum_{n=1}^{N} \mathbf{1}_{\{\tau \leq 2\}^c}(x_1 + n\alpha_1, x_2 + n\alpha_2)$[注9]，だから，定理 5.20 より，ある $N_0 \in \mathbb{N}^+$ が存在して任意の $N \geq N_0$ について

$$I_2 \ = \ \mathbb{P}^\infty \otimes \mathbb{P}^\infty \left(\frac{\mathbf{E}[|g|]}{\sqrt{N}} \left| \sum_{n=1}^{N} (\mathbf{1}_{\{\tau \leq 2\}^c}(x_1 + n\alpha_1, x_2 + n\alpha_2) - q_2) \right| > \frac{\varepsilon}{2} \right) < \frac{\delta}{3}. \qquad (5.28)$$

I_1 の評価を考えよう．$\mathbf{E}[|g|] > 0$ の場合を考えればよい．そこで

$$\forall N \geq N_1, \quad Nq_2 - \frac{\varepsilon\sqrt{N}}{2\mathbf{E}[|g|]} \ \geq \ K_0$$

となる $N_1 \in \mathbb{N}^+$ をとれば，$N \geq \max(N_0, N_1)$ のとき

$$\mathbb{P}^\infty \otimes \mathbb{P}^\infty(\nu_{N,3} < K_0) \quad \leq \quad \mathbb{P}^\infty \otimes \mathbb{P}^\infty \left(\nu_{N,3} - Nq_2 < -\frac{\varepsilon\sqrt{N}}{2\mathbf{E}[|g|]} \right)$$

[注9] $\mathbf{1}_{\{\tau \leq 2\}^c}$ は $\mathcal{B}^2$-可測だから，このように書く．

$$
\begin{aligned}
&= \mathbb{P}^\infty \otimes \mathbb{P}^\infty \left(\frac{\mathbf{E}[|g|]}{\sqrt{N}} \left(\sum_{n=1}^{N} (\mathbf{1}_{\{\tau \le 2\}^c}(x_1 + n\alpha_1, x_2 + n\alpha_2) - q_2) \right) < -\frac{\varepsilon}{2} \right) \\
&< \frac{\delta}{3}.
\end{aligned}
$$

いま

$$
B := \left\{ \left| \frac{1}{\sqrt{\nu_{N,3}}} \sum_{m=1}^{\nu_{N,3}} (g(x_3 + m\alpha_3, x_4 + m\alpha_4) - \mathbf{E}[g]) \right| > \frac{\varepsilon}{2} \right\}
$$

とすれば，任意の $N \ge \max(N_0, N_1)$ に対して

$$
\begin{aligned}
I_1 &= \mathbb{P}^\infty \otimes \mathbb{P}^\infty(B) \\
&\le \mathbb{P}^\infty \otimes \mathbb{P}^\infty(B \cap \{\nu_{N,3} \ge K_0\}) + \mathbb{P}^\infty \otimes \mathbb{P}^\infty(\nu_{N,3} < K_0) \\
&< \frac{\delta}{3} + \frac{\delta}{3} = \frac{2\delta}{3}.
\end{aligned}
$$

このことと (5.28) から $I_1 + I_2 < \delta$ となり，(5.26) より $l = 2$ の場合の証明が終わる．

では，l の場合に補題の主張が正しい仮定として，$l+1$ の場合を示そう．(5.26) と同様に

$$
\begin{aligned}
&\mathbb{P}^\infty \otimes \mathbb{P}^\infty \left(\left| \frac{1}{\sqrt{N}} \sum_{n=1}^{N} (h_{l+1}(\mathbf{x}_n) - \mathbf{E}[h_{l+1}]) \right| > \varepsilon \right) \\
&\le \mathbb{P}^\infty \otimes \mathbb{P}^\infty \left(\left| \frac{1}{\sqrt{\nu_{N,2l+1}}} \sum_{m=1}^{\nu_{N,2l+1}} (g(x_{2l+1} + m\alpha_{2l+1}, x_{2l+2} + m\alpha_{2l+2}) - \mathbf{E}[g]) \right| > \frac{\varepsilon}{2} \right) \\
&\quad + \mathbb{P}^\infty \otimes \mathbb{P}^\infty \left(\frac{\mathbf{E}[|g|]}{\sqrt{N}} \left| \nu_{N,2l+1} - N q_{l+1} \right| > \frac{\varepsilon}{2} \right) \\
&=: I_3 + I_4. \qquad (5.29)
\end{aligned}
$$

まず，

$$
B' := \left\{ \left| \frac{1}{\sqrt{\nu_{N,2l+1}}} \sum_{m=1}^{\nu_{N,2l+1}} (g(x_{2l+1} + m\alpha_{2l+1}, x_{2l+2} + m\alpha_{2l+2}) - \mathbf{E}[g]) \right| > \frac{\varepsilon}{2} \right\}
$$

とすれば，(5.27) と同様に

$$
\mathbb{P}^\infty \otimes \mathbb{P}^\infty \left(B' \cap \{\nu_{N,2l+1} \ge K_0\} \right) < \frac{\delta}{3}
$$

が成り立つ．一方，仮定 5.21(iii) より，ある $\tilde{g}_l : \mathbb{T}^2 \to \mathbb{R}$ が存在して

$$
\mathbf{1}_{\{\tau \ge 2l+2\}}(x) = \mathbf{1}_{\{\tau \ge 2l\}}(x)\,(1 - \mathbf{1}_{\{\tau = 2l\}}(x)) = \mathbf{1}_{\{\tau \ge 2l\}}(x)\,\tilde{g}_l(Z_{2l-1}(x), Z_{2l}(x)), \quad x \in \mathbb{T}^\infty.
$$

従って帰納法の仮定から，$N \to \infty$ のとき

$$
\mathbb{P}^\infty \otimes \mathbb{P}^\infty \left(\left| \frac{1}{\sqrt{N}} \sum_{n=1}^{N} (\mathbf{1}_{\{\tau \ge 2l+2\}}(\mathbf{x}_n) - q_{l+1}) \right| > \varepsilon \right) \to 0.
$$

$\sum_{n=1}^{N} \mathbf{1}_{\{\tau \geq 2l+2\}}(\mathbf{x}_n) = \nu_{N,2l+1}$ に注意せよ．このとき，ある $N_2 \in \mathbb{N}^+$ が存在して，$N \geq N_2$ ならば，

$$I_4 \;=\; \mathbb{P}^\infty \otimes \mathbb{P}^\infty \left(\frac{\mathbf{E}[|g|]}{\sqrt{N}} \left| \nu_{N,2l+1} - N q_{l+1} \right| > \frac{\varepsilon}{2} \right) \;<\; \frac{\delta}{3} \tag{5.30}$$

が成り立つ．ここで

$$\forall N \geq N_3, \quad N q_{l+1} - \frac{\varepsilon \sqrt{N}}{2\mathbf{E}[|g|]} \;\geq\; K_0$$

なる $N_3 \in \mathbb{N}^+$ をとれば，$N \geq \max(N_2, N_3)$ のとき

$$\mathbb{P}^\infty \otimes \mathbb{P}^\infty (\nu_{N,2l+1} < K_0) \;<\; \frac{\delta}{3}.$$

最後に，以上から任意の $N \geq \max(N_2, N_3)$ に対して

$$\begin{aligned} I_3 \;&=\; \mathbb{P}^\infty \otimes \mathbb{P}^\infty (B') \\ &\leq\; \mathbb{P}^\infty \otimes \mathbb{P}^\infty (B' \cap \{\nu_{N,2l+1} \geq K_0\}) + \mathbb{P}^\infty \otimes \mathbb{P}^\infty (\nu_{N,2l+1} < K_0) \\ &<\; \frac{\delta}{3} + \frac{\delta}{3} \;=\; \frac{2\delta}{3}. \end{aligned}$$

このことと (5.30) から $I_3 + I_4 < \delta$ となり，(5.29) より $l+1$ の場合の証明が終わる．□

定理 5.23 の証明． f は仮定 5.21 を $r = 2$ として満たすとしよう．(一般の r でも同様に証明できる．) $L \in \mathbb{N}^+$ に対して

$$f_L(x) \;:=\; f(x)\mathbf{1}_{\{\tau \leq 2L\}}(x), \quad f'_L(x) \;:=\; f(x) - f_L(x), \quad x \in \mathbb{T}^\infty,$$

とおく．任意の $\varepsilon > 0$ をとる．

$$\begin{aligned} &\mathbb{P}^\infty \otimes \mathbb{P}^\infty \left(\left| \frac{1}{\sqrt{N}} \sum_{n=1}^{N} (f(\mathbf{x}_n) - \mathbf{E}[f]) \right| > \varepsilon \right) \\ &\quad\leq\; \mathbb{P}^\infty \otimes \mathbb{P}^\infty \left(\left| \frac{1}{\sqrt{N}} \sum_{n=1}^{N} (f_L(\mathbf{x}_n) - \mathbf{E}[f_L]) \right| > \frac{\varepsilon}{2} \right) \\ &\qquad + \mathbb{P}^\infty \otimes \mathbb{P}^\infty \left(\left| \frac{1}{\sqrt{N}} \sum_{n=1}^{N} (f'_L(\mathbf{x}_n) - \mathbf{E}[f'_L]) \right| > \frac{\varepsilon}{2} \right) =:\; I_5 + I_6. \end{aligned}$$

さて，$|f'_L(x)|^2 \leq |f(x)|^2$ であり，$L \to \infty$ のとき $f'_L(x) \to 0$，$\mathbb{P}^\infty$-a.s. であるから，ルベーグの収束定理によって

$$\lim_{L \to \infty} \mathbf{E}\left[\left| f'_L \right|^2 \right] \;=\; 0.$$

従って，任意の $\delta > 0$ に対してある $L_0 \in \mathbb{N}^+$ が存在して，$L \geq L_0$ ならば

$$\mathbf{E}\left[\left| f'_L \right|^2 \right] \;<\; \frac{\varepsilon^2 \delta}{8}$$

とできる．チェビシェフの不等式を適用すると，任意の $N \in \mathbb{N}^+$ に対して

$$\begin{aligned} I_6 &= \mathbb{P}^\infty \otimes \mathbb{P}^\infty \left(\left| \sum_{n=1}^{N} (f'_L(\mathbf{x}_n) - \mathbf{E}[f'_L]) \right| > \frac{\varepsilon \sqrt{N}}{2} \right) \\ &\leq \frac{4}{\varepsilon^2 N} \mathbf{V} \left[\sum_{n=1}^{N} f'_L(\mathbf{x}_n) \right] = \frac{4}{\varepsilon^2} \mathbf{V}[f'_L] \\ &\leq \frac{4}{\varepsilon^2} \mathbf{E} \left[\left| f'_L \right|^2 \right] < \frac{\delta}{2}. \end{aligned}$$

なお，ここで定理 5.19 を用いて分散の計算をしている．

以下 $L \geq L_0$ を固定する．I_5 を評価しよう．

$$I_5 \leq \sum_{l=1}^{L} \mathbb{P}^\infty \otimes \mathbb{P}^\infty \left(\left| \frac{1}{\sqrt{N}} \sum_{n=1}^{N} \left(f(\mathbf{x}_n) \mathbf{1}_{\{\tau=2l\}}(\mathbf{x}_n) - \mathbf{E}[f \mathbf{1}_{\{\tau=2l\}}] \right) \right| > \frac{\varepsilon}{2L} \right)$$

の右辺の各項を評価すればよい．f が仮定 5.21($r = 2$) を満たすことから，各 l に対してある 2 変数関数 $g_l : \mathbb{T}^2 \to \mathbb{R}$ が存在して

$$f(x) \mathbf{1}_{\{\tau=2l\}}(x) = \mathbf{1}_{\{\tau \geq 2l\}}(x) \cdot g_l(Z_{2l-1}(x), Z_{2l}(x)), \quad x \in \mathbb{T}^\infty,$$

となっている．そこで補題 5.24 により，ある $N_0 \in \mathbb{N}^+$ が存在して，任意の $N > N_0$ に対して

$$\mathbb{P}^\infty \otimes \mathbb{P}^\infty \left(\left| \frac{1}{\sqrt{N}} \sum_{n=1}^{N} \left(f(\mathbf{x}_n) \mathbf{1}_{\{\tau=2l\}}(\mathbf{x}_n) - \mathbf{E}[f \mathbf{1}_{\{\tau=2l\}}] \right) \right| > \frac{\varepsilon}{2L} \right) < \frac{\delta}{2L}$$

が成り立つ．よってこのとき $I_5 < \delta/2$ となる．

以上から，$N \geq N_0$ に対して $I_5 + I_6 < \delta$ となる． □

注意 5.25 一般に確率変数列が L^2-有界で，しかも 0 に確率収束すれば，任意の $p \in [1, 2)$ に対して，それは 0 に L^p で収束する．だから定理 5.23 の仮定の下で

$$\lim_{N \to \infty} \mathbf{E} \left[\left| \frac{1}{\sqrt{N}} \sum_{n=1}^{N} (f(\mathbf{x}_n) - \mathbf{E}[f]) \right|^p \right] = 0, \quad 1 \leq p < 2,$$

が成り立つ．

第6章　実装

6.1　RWSの実装

6.1.1　例 2.9 の実装例

例 2.9 の計算は次の C 言語プログラムによる．このプログラムは $S_{10^6}(g(\omega'))$ の値 546,177 と求めたい確率の推定値 0.546177 を出力する．

```
/*============================================================*/
/*  file name: rws_example.c                                  */
/*============================================================*/
#include <stdio.h>

#define SAMPLE_NUM 1000000
#define M          100
#define M_PLUS_J   119

/* seed */
char xch[] =
   "1110110101" "1011101101" "0100000011" "0110101001" "0101000100"
   "0101111101" "1010000000" "1010100011" "0100011001" "1101111101"
   "1101010011" "111100100";
char ach[] =
   "1100000111" "0111000100" "0001101011" "1001000001" "0010001000"
   "1010101101" "1110101110" "0010010011" "1000000011" "0101000110"
   "0101110010" "010111111";

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, s = 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_NUM; n++ ){
  longadd();
  if ( maxLength() >= 6 ) s++;
 }
 printf ( "s=%6d, p=%7.6f\n", s, (double)s/(double)SAMPLE_NUM);
 return 0;
}
/*================ End of rws_example.c ==================*/
```

6.1.2 例 5.9 の実装例

例 5.9 での計算例 (図 5.2 のデータ) は，次の C 言語プログラムによる.[注1]このプログラムは § 6.2 の汎用 C 言語ライブラリ *random_samler* の関数 `m90setseeds`, `m90randombit` を使う.

```
/*============================================================*/
/*  file name: rws_S500.c                                     */
/*============================================================*/
#include <stdio.h>
#include "random_sampler.h"

#define SAMPLE_NUM 10000000
#define M          500
#define M_PLUS_J   523

int  x[M_PLUS_J], a[M_PLUS_J], hist[M+1];

void longadd(void)   /* x = x + a ( long digit addition ) */
```

[注1]このプログラムの出力は長いので，テキストファイルに出力する (リダイレクトする) とよい.

```
{
 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 s500(void)
{
 int s = 0, i;
 for ( i = 0; i <= M-1; i++ ) s += x[i];
 return s;
}

int main()
{
 int n,i;

 m90setseeds(664426,5161592,7773372,84171419,1545);
 for( i = 0; i <= M ; i++ ) hist[i] = 0;
 for( i = 0; i <= M_PLUS_J-1; i++ ) x[i] = m90randombit();
 for( i = 0; i <= M_PLUS_J-1; i++ ) a[i] = m90randombit();

 for ( n = 1 ; n <= SAMPLE_NUM ; n++ ){
  longadd();
  hist[s500()]++;
  if ( n == 1000 || n == 10000 || n == 100000 || n == 10000000 ){
   printf ("%d samples:\n",n);
   for( i = 0; i <= M ; i++ )
    printf ( "%3d : %8.7f\n", i,(double)hist[i]/(double)n);
   printf ("\n");
  }
 }
 return 0;
}
/*================ End of rws_S500.c ====================*/
```

6.2 汎用C言語ライブラリ：*random_sampler*

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 *random_sampler* を紹介する.

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 § 4.2.1 の (4.6)(4.7)(4.8) 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．[注2] これは汎用の疑似乱数生成器である．

- DRWS
  $K = j = 31$ として § 5.4.1 の (5.11)(5.12)(5.13) によって DRWS の (すなわちモンテカルロ積分専用の) 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．

6.2.1　ソースコード

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．

• random_sampler.c

```
/*============================================================*/
/*  file name: random_sampler.c                               */
/*============================================================*/
#include <stdlib.h>

#define  LIMIT_30  0x3fffffff
#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"      */
```

[注2] 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．

```
/* 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;
}

/* Sets the first location */
void set_first_location(void)
{
   location = -1; current_ptr = &random_list;
}

/* Generates a dynamic random Weyl sample (31 bit integer) */
unsigned long drws31(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 a dynamic random Weyl sample in [0,1) */
double drwsu(void)
{
   unsigned long drws31copy = drws31();
   if ( drws31() == CARRY_32 ) return -1.0;
   else return (double)drws31copy/(double)CARRY_32;
}
/*================ End of random_sampler.c ====================*/
```

• 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  void           set_first_location(void);
extern  unsigned long  drws31(void);
extern  double         drwsu(void);

/*================ End of random_sampler.h ====================*/
```

6.2.2　定数と関数の仕様

random_sampler 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 0x80000000= 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 § 4.2.1 の (4.6)(4.7) の $\tilde{x}$ 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 (4.6)(4.7)(4.8) によって定義される $Y_n^{(m)}(\tilde{x}; \lfloor\alpha\rfloor_{m+j})$ に相当する．$Y_n^{(m)}(\tilde{x}; \lfloor\alpha\rfloor_{m+j})$ は $\tilde{x} + n\lfloor\alpha\rfloor_{m+j}$ の上位 m ビットのパリティであり，ここではそれを高速で計算する工夫がなされている．
- `unsigend long m90random31();`
  ランダムな符号なし 31bit の整数 ($0 \sim 2^{31}-1$ =RANDMAX-1 = 0x7fffffff) を返す．ここでは 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．ここでは m90random31() の値を RANDMAX で割り算している．

- DRWS
  以下の解説では，被積分関数である確率変数は § 1.3.1 の仮定 1.9 を満たしているとし，§ 5.4.3 で用いた記号も踏襲する．
  - `void init_drws();`
    DRWS の初期化．最初に必ず一度だけ呼び出す．
  - `void end_drws();`
    DRWS が使用したメモリーを解放する．最後に必ず一度だけ呼び出す．
  - `void set_first_location();`
    各々のサンプルを生成するとき，Z_1 を生成する前に一度呼び出す．
  - `unsigned long drws31();`
    符号なし 31bit 整数を返す．一つの Z_l ($l \geq 1$) 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，drws31() 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，drws31() は RANDMAX を返してプログラマに異常事態を知らせる．
  - `double drwsu();`
    31bit 精度の $[0, 1]$-値実数を返す．一つの Z_l ($l \geq 1$) 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，drwsu() 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，drwsu() 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 f に用いた $Z_1,\ldots,Z_T$ のうち，最大の T 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．

6.2.3 サンプルプログラム

• m90random

疑似乱数生成器 m90random の一部の関数はすでに § 6.1.2 のプログラムで用いている．また，次のプログラム drws.c でも用いられている．

• DRWS

サンプルプログラム drws.c は § 1.3.1 の例 1.12 の確率変数 f の平均を求める．すなわち，硬貨を投げ続けて表の出た回数が 5 になるような最初の時刻の平均を求める．

```
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, f;
15:   double sum_of_f;
16:
17:   m90setseeds(0,53,0,0,0);
18:   init_drws();
19:
```

```
20:   sum_of_f=0.0;
21:   for ( i=1; i <= SAMPLE_SIZE; i++ ){
22:      number_of_heads=0;
23:      f=0;
24:      set_first_location();
25:      while ( number_of_heads < 5 ){
26:         f++;
27:         if ( drws31() >= halfmax ) number_of_heads++;
28:      }
29:      sum_of_f += f;
30:   }
31:   printf("Mean of hitting time = %f\n", sum_of_f/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* 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(), drws31()

の値の最大値 +1 の値が RANDMAX であり，その値は 0x80000000 である．従って，この行で定義された halfmax は，その値の半分 0x40000000 である．

17: 疑似乱数生成器の初期化を行う．引数 (疑似乱数の種) は固定せずにユーザが入力できる形にしておくべきだが，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: 最初の Z_1 を生成する準備をする．

27: drws31() が呼ばれるごとに $Z_1, Z_2, \ldots$ に相当するランダムな整数が順に生成される．これが halfmax 以上になる確率は 1/2 で，そのとき number_of_heads 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 number_of_heads の値が 5 になった Macro 2 とき，このループの外に出る．その間，drws31() 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 f 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

```
sum_of_f/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 T 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前に

```
set_first_location()
```

を呼び出すことである．そのあと drws31() または drwsu() を呼んで順に $Z_1, Z_2, \ldots$ の生成を行う．

6.2.4　制限事項

• 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 (4.18) で定義した臨界サンプル数を上限と考えれば，それは $N_c^{(90)}(10000) = 8.7 \times 10^{14}$ 個，m90random31() および m90randomu() の場合はその 1/31 倍で 2.8×10^{13} 個である．[注3]

• DRWS: サンプル数の上限

random_sampler 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．[注4]

[注3] このサンプル数の上限を超えても，実際には疑似乱数を生成することはできるが，その場合，無視できない相関が出てくる可能性がある．

[注4] このサンプル数の上限を超えても，実際にはサンプルを生成することはできるが，その場合，ペアごとの独立性が完全ではなくなる．

• DRWS: メモリの使用量の管理

random_sampler の DRWS の実装では一組の (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．[注5]たとえば前節のサンプルプログラム `drws.c` 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．最近のコンピュータ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 f 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．

[注5] `unsigned long` 型が 32 bit = 4byte の場合．

おわりに

ランダム性の数学的定式化の流れを振り返ると二つの指導原理に気づく．

一つは「ランダム性を量的尺度によってのみ捉えよ」ということである．ランダム性を哲学的に論じることは多い．しかし完全に形式的な数学論理においてランダム性の意味を問うことは不可能であるから，この指導原理は自然な考えである．この指導原理はシャノン (Shannon) の画期的な情報理論に始まり，コルモゴロフ複雑度，計算量的に安全な疑似乱数生成器の概念に至る．

他の一つの指導原理は — 逆説的であるが —「決定論的に定式化せよ」ということである．確率論の公理，乱数の概念，疑似乱数生成器とその安全性の概念，みなそうである．だから，モンテカルロ法をやはり決定論的な考え (つまり確率変数のサンプリングを実行者の意思で行うという考え) に基づいて定式化することは自然な流れというべきだろう (§ 2.2).[注1] ゲーム理論や暗号理論では確率変数のサンプリングを実行者の意思で行うことはごく普通のことである．すなわち，前者ではいくつかの選択可能な戦略のうちどれを選ぶかはプレーヤーの意思に任されるし，後者では暗号通信におけるパスワードは当事者が決める．しかしモンテカルロ法では新機軸であろう．これによって，モンテカルロ法の定式化が乱数の概念や疑似乱数生成器とその安全性の概念と整合性を持ち，その結果，本質的問題の所在が明らかになり，さらにモンテカルロ積分の場合には問題の解決を得たことは本文で述べたとおりである．

暗号理論における疑似乱数の文献はインターネットで知った．疑似乱数生成器のねらいはランダム性を作り出すことではなかったことを知って目から鱗が落ちる思いがした．しかしそれは数ある計算機科学の優れた考え方の一つに過ぎない．今日の計算機科学の重要な課題はすべて確率が絡んでいるといっても過言ではない．決定論的手法で解けるものはもうほとんど解かれている．そこで決定論的手法では埒が明かないような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によるランダムサンプリングによって解決を図る，というのが，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ことが分かる．

当然，計算機科学の分野でよく確率論が使われている．とくに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．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ことがある．

[注1]「はじめに」で述べた神戸大学での集中講義 (2000 年) のときの講義ノートには，この考えがなかった．そのことに筆者自身が無意識のうちに納得していなかったのだろう．この考えの最初の萌芽は [48] に見られる．本書ではそれをさらに徹底して展開した．

実際，§ 5.2.3 で紹介した疑似乱数生成器 $\tilde{g}$ は GF(2^m) の四則演算が複雑すぎて実用的でない．そのアナロジーで，解析学的に容易に得られる § 5.2.1 定理 5.5 を離散化した RWS が実際の計算には数段便利である．一般に，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，連続性に欠け，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．一方，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しれないが，パラメータの変動に伴い連続的に変化する現象に注目する．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．

本書全編を通じて，その理念を技術的に実現させているのがワイル変換 (無理数回転) という力学系である．すなわち，RWS，疑似乱数生成器および DRWS はすべてこの力学系を応用したものである．ワイル変換はエルゴード的という意味ではランダムだが，エントロピー 0 という意味ではほとんど決定論的である．この二面性が小さな入力 (種) から多様で長い出力 (疑似乱数) を生成することを可能にしている．これほど簡単な力学系がモンテカルロ法のために非常に都合のよい性質を有していた幸運を筆者はとても有り難く思う．

参考文献

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

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

[3] 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.

[4] E. Borel, Les probabilités dénombrables et leurs applications arithmétiques, *Rend. Circ. Mat. Palermo*, **27** (1909), 247–271.

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

[6] M. Davis, *Computability and Unsolvability*, McGraw-Hill, (1958). (邦訳) 計算の理論 (渡辺茂，赤攝也訳)，岩波書店，(1966).

[7] R. Durrett, *Probability: theory and examples*, Second edition. Duxbury Press, Belmont, CA, 1996.

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

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

[10] 舟木直久，確率論，朝倉書店，(2004).

[11] 伏見正則，乱数，東京大学出版会，(1989).

[12] 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.

[13] 樋口保成，西尾真喜子，確率過程入門，確率論教程シリーズ3，培風館，(2006).

[14] 広瀬健，横田一正，ゲーデルの世界 — 完全性定理と不完全性定理，海鳴社，(1985).

[15] S. Janson, Some pairwise independent sequences for which the central limit theorem fails, *Stochastics* **23-4** (1988), 439–448.

[16] 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.

[17] A. Joffe, On a sequence of almost deterministic pairwise independent random variables, *Proc. Amer. Math. Soc.* **29** (1971), 381–382.

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

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

[20] 笠井琢美，計算量の理論，近代科学社，(1987).

[21] D.E. Knuth, *The Art of Computer Programming*, 2nd ed., Volume 2, Chapter 3, Addison-Wesley, (1981). (邦訳) 準数値算法/乱数 (渋谷政昭訳)，サイエンス社，(1983).

[22] A.N. Kolmogorov, On tables of random numbers, *Sankhyā, Indian J. Statist. Ser. A*, **25** (1963) 369–376.

[23] A.N. Kolmogorov, Three approaches to the definition of the concept “quantity of information”. (Russian) *Problemy Peredači Informacii* 1 (1965) vyp. 1, 3–11.

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

[25] A.N. Kolmogorov, *Selected works of A. N. Kolmogorov. Vol. III. Information theory and the theory of algorithms*, Edited by A. N. Shiryayev [A. N. Shiryaev]. Translated from the 1987 Russian original by A. B. Sossinsky. Mathematics and its Applications (Soviet Series), 27. Kluwer Academic Publishers Group, Dordrecht, (1993) xxvi+275 pp.

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

[27] 小谷眞一, 測度と確率，岩波書店，(2005).

[28] M. Li and P. Vitányi, *An introduction to Kolmogorov complexity and its applications*, 2nd ed. Graduate Texts in Computer Science. Springer-Verlag, New York, (1997).

[29] M. Luby, *Pseudorandomness and cryptographic applications*, Princeton Computer Science Notes, Princeton University Press, (1996).

[30] G. Maruyama, On an asymptotic property of a gap sequence, *Kôdai Math. Sem. Rep.*, **2** (1950), 31–32.

[31] P. Martin-Löf, The definition of random sequences, *Inform. Control*, **9** (1966), 602–619.

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

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

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

[35] K.R. Parthasarathy, *Probability measure on metric spaces*, Academic Press, (1967).

[36] H. Pfeiffer, *Monte Carlo methods with few random bits*, Dissertation zur Erlangung des akademischen Grades doctor rerum naturalium (Dr. rer. nat.), vorgelegt dem Rat der Fakultät für Mathematik und Informatik der Friedrich-Schiller-Universität Jena, 2005.

[37] Ya.G. Sinai, *Probability Theory — An Introductory Course*, Springer Textbook, 1992. (邦訳) シナイ確率論入門コース (森真訳)，シュプリンガー・フェアラーク東京，(1995).

[38] I.M. Sobol', *A Primer for the Monte Carlo Method*, CRC Press, (1994).

[39] H. Steinhaus, Sur la probabilité de la convergence de séries, *Studia Math.*, **2** (1930), 21–39.

[40] D.R. Stinson, *Cryptography (Theory and practice)*, CRC Press, (1995).

[41] 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.

[42] 杉田洋，無限次元準モンテカルロ法，数理解析研究所講究録 **850**，確率数値解析における諸問題，(1993).

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

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

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

[46] 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.

[47] 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.

[48] H. Sugita, Security of Pseudo-random Generator and Monte-Carlo Method, *Monte Carlo Methods and Appl.*, **10-3**, VSP, (2004), 609–615.

[49] H. Sugita, *Monte Carlo method, random number, and pseudorandom number*, MSJ Memoirs **25** (2011), xiv+133 pp.

[50] 杉田洋，確率と乱数，数学書房選書 4，数学書房，(2014).

[51] 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://www.math.sci.osaka-u.ac.jp/~sugita/mathematics.html`.

[52] H. Sugita and S. Takanobu, Random Weyl sampling for robust numerical integration of complicated functions, *Monte Carlo Methods and Appl.*, **6-1**, VSP, (2000), 27–48.

[53] 高橋正子，計算論 (計算可能性とラムダ計算)，第 1 章，近代科学社，(1991).

[54] 高橋陽一郎+志賀浩二，確率論をめぐって，日本評論社，(1992).

[55] 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.

[56] 津田孝夫，モンテカルロ法とシミュレーション〈改訂版〉，培風館，(1977).

[57] A. Turing, On Computable Numbers, With an Application to the Entscheidungsproblem, *Proceedings of the London Mathematical Society, Series 2*, Volume 42 (1936). Eprint. Reprinted in The Undecidable pp.115–154. `http://www.abelard.org/turpap2/tp2-ie.asp`

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

[59] 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).

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

[61] K. Yasutomi, A direct proof of dependence vanishing theorem for sequences generated by Weyl transformation, *J. Math. Kyoto Univ.*, **43-3** (2003), 599–607.

[62] K. Yasutomi, A dependence vanishing theorem for sequences generated by Weyl transformation, *J. Math. Kyoto Univ.*, **44-2** (2004), 365–380.

[63] 安富健児，Private communication.

[64] A.K. Zvonkin and L.A. Levin, The complexity of finite objects and the development of the concepts of information and randomness by means of the theory of algorithms, (Russian), *Uspehi Mat. Nauk*, **25-6(156)** (1970), 85–127, (English translation by S.M.Rudolfer) *Russian Math. Surveys*, **25-6** (1970), 83–124.

索 引

記号

* 本書の最新版は
http://www.math.sci.osaka-u.ac.jp/~sugita/mcm.html に掲載.