1 初めに

最近の電子計算機の普及と高性能化により、計算機シミュレーションは工学、理学などの分野のみならず、経済学や社会学など人文科学においても重要な手法になっている。特に、近年注目を集めている、株価予想など数理ファイナンスと呼ばれる分野においては、計算機シミュレーションの技法を利用したアプリケーションソフトウェアも実用化されている。このような計算機シミュレーションにおいては、擬似乱数などを利用した Monte Carlo 法は必要不可欠な道具として利用されている。Monte Carlo 法は古くは Buffon の針の実験として知られているが、現代では、計算機で生成される乱数モドキ、擬似乱数、を用いる数値計算法として利用されている。

擬似乱数は電子計算機の黎明期から研究、実用化されてきた。例えば、最初に擬似乱数を考案したのは、有名な von Neuman で、彼の擬似乱数生成法は、平方採中法、と呼ばれるものであった。この方法は、擬似乱数生成法としてはあまり良くないものであったため現在では利用されないが、その後すぐに、現在でも広く利用されている、線形合同法が考案され、実用に供されている。さらに、符号理論などとの関連から、 shift register sequence (m-sequence などとも呼ばれる) なども古くから研究、実用化されてきた。この他にも、加算生成法などもよく知られている。

また、擬似乱数のランダムネス（一様分布性、無相関性など）について、理論的研究や統計的検定方法などが多くの研究者によって研究されてきた。しかしながら、理論的研究の多くは周期の長さ等についての研究や、一周期全体にわたる一様分布性などの研究である。更に、統計的検定方法のうち、古くから知られている方法は、乱数表（物理的にルーレットなどを利用して得られた「乱数」などの表）の検定方法として考案されたものである。また、近年に計算機上での利用を考慮して考案された、 spectral 検定などは主に線形合同法を対象にしたもので、 m-sequence や加算生成法などの擬似乱数には適用が困難である。更に、この方法は擬似乱数の一周期全体に対して空間内での分布の度合いを検定するものであり、乱数列の時系列としての検定ではない。このように古くから知られている検定方法は、近年の高性能な計算機での大規模なシミュレーションにおけるような擬似乱数の利用法の現状に相応しい検定方法とは考えにくい。

また、例えば従来からの spectral 検定などのような、空間内での一様分布性の研究や検定と、偶然現象などシミュレーションにおける擬似乱数の利用法（時系列や確率過程のシミュレーション）との間の問題意識のズレなども重要な問題と考えられる。例えば、空間内に均等に分布する乱数列が Brown 運動の見本関数のシミュレーションに適当かどうか、という疑問が自然に派生してくるであろう。残念ながら、これらの疑問に理論的に答えることは、現段階では極めて困難である。

擬似乱数は計算機シミュレーションなどに広く利用されるようになってきており、実用上の重要性は増すばかりであるが、一方では、擬似乱数生成法の多くは有限代数や解析学と密接な関係があり、特に、線形合同法、 m-sequence 、加算生成法などは理論的研究対象としても重要である。これらの生成法の研究には未知の理論的な問題が多く存在する。線形合同法や m-sequence、加算生成法に限らず、計算機上でアルゴリズムに基づいて生成される擬似乱数を用いたシミュレーションによって得られる現象については代数学的な説

明が可能なはずであり、説明を可能にするには新しい理論が必要になる可能性は大きいと思われる。本報告では、この点についても簡単に触れたい。

一方で、半導体の熱雑音や放射性物質を利用した、物理乱数発生装置も実用化されており、近年では personal computer に内蔵可能なものも販売されているようであり、他方では、最新の super computer に搭載されるような高価な装置も実用化されている。このような物理乱数と擬似乱数の違いは前者が自然現象を利用しているのに対して、後者は計算機でアルゴリズムに基づいて生成される点にある。前者は自然現象を利用するが為に、真に乱数であるか否か、が検証不可能であり、後者は計算機を利用するために、乱数ではあり得ない。また、前者は自然現象を利用するので生成速度を高速化することが難しいが、後者は計算機が高速化されればそれに従って高速化される。このような違いの他、計算機で計算される、擬似乱数には、初期値が同じであれば、いつでも同じ数列を生成できる、という物理乱数にはない利点がある。即ち、あるシミュレーションにおいて、全体的な計算が正しいか否か、を繰り返し同じ数列を用いて検証できるのである。このようなことを、物理乱数を用いて行うためには、生成された数列全体を補助記憶装置等に記録しておかねばならない。しかしながら、近年の Monte Carlo 法のように、１つの計算で数十億、数百億もの乱数を利用する場合には、それらの数列全体を記録することは現実的ではないであろう。このような理由からも、擬似乱数は近年の計算機シミュレーションには不可欠は道具となっている。

本報告では、擬似乱数の生成法として、理論的研究が比較的容易であり、広く利用されている、線形合同法、 m-sequence 、加算生成法などを取り上げ、確率過程として最も簡単なモデルであり、また最も重要な対象である random walk による統計的検定方法とその結果について考察し、そこから派生する理論的問題について報告する。

2　擬似乱数生成法について

ここでは、数学の話題としてなじみの少ない、擬似乱数生成法について、伏見 [15] 、Knuth [22] などを参考に簡単に振り返ってみることにする。詳しいことは、それらの文献を参照されたい。

擬似乱数は計算機で生成される、乱数モドキ、であり、その生成法のほとんどは代数学的なアルゴリズムによるものである。また、擬似乱数はその性格上、漸化式で計算され、有限な値をとるので必ず繰り返す。この繰り返しの間隔の最小値を周期という。つまり、擬似乱数列を x_i とし、周期を P とすると

$$x_{P+i} = x_i \quad i \geq 0 \ ,$$

となる。以下に重要な生成法の例として、線形合同法、m-sequence, 加算生成法、Cellular Automaton 擬似乱数、について概略を述べる。

2.1　線形合同法

$M > 0$ を十分に大きな整数とする。通常は、計算機で扱える整数の範囲内で最大の素数をとるか、あるいは、2^w をとることが多い。ここで、w は計算機の整数（符号なし整数、つまり、C 言語でいうところの unsigned int のこと）のビット長である。最近の personal computer などでは、 $w = 32$ である。$a > 0, b$ は整数とする。このとき

$$x_n = a\ x_{n-1}\ +\ b\ , \quad (\mathrm{mod}\ M)\ , \quad (n > 0)$$

で整数 $x_n, 0 \leq x_n < M$ を生成する方法を線形合同法という。ここで、初期値 x_0 は適当に与える。この生成法は提案者の名前をとって Lehmer 法とも呼ばれ、広く利用されてきており、現在でも世界中の計算機で利用されている。ここで、$M = 2^w$ とする利点は、掛け算および足し算の結果生じる整数オーバーフロー（計算結果が 2^w 以上になること）は通常、オーバーフローした部分が無視されるため、わざわざ $\pmod M$ を計算しなくてすむ、という点にある。然しながら、システムによっては、整数オーバーフローが禁止されていたり、エラーメッセージを出したりする場合などでは、 $\pmod M$ の計算を注意して行う必要がある。このことは、 M として、最大の素数をとる場合でも同様である。

$(0, 1)$ 上の一様乱数 $y_n, 0 \leq y_n < 1$, を得るには、$y_n = x_n/M$ とすればよい。

性質

- 周期は M を越えることができない。そこで、$x_{n+k} = x_{m+k}, k \geq 0, n \neq m$ となる添え字の組 m, n が必ず存在するが、 $|n - m|$ 最小値を周期とよぶ。線形合同法の場合、1 次式で計算されるので、周期は M 以下である。

- (x_n, x_{n+1}) 平面上の点と考えると、線形合同法の場合、これらの点は幾つかの直線上に並んでしまう。この性質は「粗結晶構造」などと呼ばれ、線形合同法の欠点とされてきた。実際、

$$y = a\ x\ +\ b \quad (\mathrm{mod}\ M)\ ,$$

上に並ぶ。$(x_n, x_{n+1}, \ldots, x_{n+k-1})$ として k 次元空間で考えると $k-1$ 次元超平面上に並ぶ。

- **係数 a の選び方について**
  x_n の周期は上記のように M を越えることができないので、係数 a を適当に選んで、なるべく周期を長く、かつ x_n がなるべくランダムに見えるようにする必要性がある。これについては、以下のような、x_n が最大周期を持つための必要十分条件が知られている。

  1. b が M と互いに素である。
  2. $a-1$ が M を割りきるすべての素数の倍数である。
  3. M が 4 の倍数であれば、 $a-1$ も 4 の倍数である。

- **乗算合同法の場合の注意**
  $b=0$ の場合、特にこの生成法を乗算合同法、と呼ぶことがある。上記のように、$b=0$ の場合には最大周期 M を達成することができない。然し、M が十分大きければ周期は十分大きくなる。また、b の値は数列 x_n のランダムネスには大きな影響を与えないことが知られており、乗算合同法はよく利用されている生成法の一つでもある。以下に、乗算合同法の場合の M , a の選び方について知られている結果を簡単に述べる。

  1. $M=2^e, e \geq 4$ の場合、可能な最大周期は $M/4$ であり、 $a=3 \pmod 8$, または $a=5 \pmod 8$ の場合で、そのとき初期値 x_0 は奇数でなければならない。
  2. $M=p>2$, p が素数の場合、可能な最大周期は $p-1$ であり、初期値 x_0 は $\neq 0$ であり、かつ、 $a^{(p-1)/q} \neq 1, \pmod p$ が、$p-1$ のすべての素因数 q 対して成り立つ場合である。

  このことから、よく利用されている、 $M=2^w$ の乗算合同法の場合、初期値 x_0 は奇数を選ぶことが重要になる。その場合、 x_n はすべて奇数となるが、最大周期は $M/4$ であるので、すべての奇数を取るわけではない。詳しくは、伏見 [15] を参照のこと。

2.2 m-Sequence, またはM系列

前節で説明した、線形合同法はそのアルゴリズム上、最大周期は、法 M を越えることができない。現在の personal conputer では $M = 2^{32}$ をとることが可能であるが、近年のシミュレーションの規模大きさを考えると、必ずしも十分な大きさとは言えない。また、前述したように、線形合同法には「粗結晶構造」という問題もある。

そこで、線形合同法に代わるものとして、 Maximum-length linearly recurring sequence, 略して m-sequence, あるいはM系列、と呼ばれる生成法が考案されている。この生成法は有限代数体 GF(2) とその拡大体 GF(2^p) の理論を基礎としている。そこで、以下に簡単に GF(2) , GF(2^p) について述べる。詳しいことは、伏見 [15] , Lidl and Nierderreiter [26] , などを参照されたい。

2.2.1 Galois 体

まず、 0 と 1 からなる最小の代数的体 GF(2) について考える。代数的体とは、数の集合で、加算、減算、乗算、除算の四則演算が定義され、これらの演算に関して閉じているもののことである。GF(2) の場合、以下のように四則演算が定められる :

$$0+0=0\ ,\ \ 0+1=1\ ,\ \ 1\times 1=1\ ,\ \ 1\times 0=0\ ,\ \ 1/1=1\ .$$

この数体上で多項式を考える。

$$f(x)=\sum_{i=0}^{n}\ c_i\ x^i\ \ ,\ \ c_i\in \mathrm{GF}(2)\ .$$

多項式の演算は通常の実数体の場合と同様であるが、基礎の数体が GF(2) であるので、例えば以下のような計算になることに注意されたい :

$$(1+x)^2=1+2\ x+x^2=1+x^2\ \ .$$

多項式が可約、あるいは既約である、という定義は通常の定義に準ずる。上でみたように、実数体の場合には既約である、$1+x^2$ は GF(2) の上では可約になることが分かる。ある既約な多項式 $f(x)$, $\deg f(x)=p$ が原始多項式であるとは、$1+x^n$ が $f(x)$ で割り切られるような次数 n の最小の値が 2^p-1 であるとことする。

$f(x)$ を GF(2) 上の原始多項式とし、 GF(2) 上の多項式を $f(x)$ で割った余りの多項式の全体を考える :

$$\mathrm{GF}(2^p)=\{g\pmod f : g \text{ は GF(2) の多項式 }\},$$

を考える。ここで $p=\deg f$ である。代数でよく知られているように、GF(2^p) は GF(2) の p 次の拡大体と呼ばれるものである。$f(x)$ が原始多項式であることは $f(x)=0$ の根 σ (GF(2^p) の元) が原始根であることを意味する。つまり、 $\sigma^{2^p-1}=1$ となる。これらのことに関しては Lidl and Niederreiter [26] , 山本 [] などを参照されたい。

2.2.2 原始多項式 $f(x)$ から生成される m-sequence

$$f(x) = \sum_{i=0}^{p} c_i x^i \quad , \quad c_i \in \mathrm{GF}(2) \, , c_0 \neq 0, \quad c_p \neq 0 \, ,$$

を GF(2) 上の原始多項式とする。 $f(x)$ に対して、 GF(2) の元 a_n の列を

$$a_n = \sum_{i=1}^{p} c_i \, a_{n-i} \quad , \quad n \geq p \, ,$$

ここで、初期値 $a_0 \, , \ldots \, , a_{p-1}$ はすべてが 0 、とはならないように任意に定める。GF(2^p) は p 個の 0 , 1 の組の集合と考えられ、上記の数列から p 個取ってきた組 $(a_n, a_{n+1}, \ \ldots \ , a_{n+p-1})$ は GF(2^p) の元である。初期値の組が GF(2^p) の原始根である場合には、

$$\#(\{(a_n, \ \ldots \ , a_{n+p-1}) : n \geq 0\}) = 2^p - 1 \, ,$$

となる。ここで、 $\#(A)$ は集合 A の元の個数を表す。このように定められる GF(2) の元 a_n の列を、Maximum-length linearly recurring sequence , 略して m-sequence , という。また、その作り方から shift register sequence , とも呼ばれる。$f(x)$ を本報告では Jungnickel [21] に従って 、m-sequence a_n の feedback polynomial , と呼ぶことにする。通常は特性多項式 characteristic polynomial という用語が使われる場合が多いが、例えば、Lidl and Nierderreiter [26] と Golomb [17] では呼び方に関して混乱があり、本報告では敢えて特性多項式という名前は避けることにする。この問題については後述することにする。

実際の計算機の上で m-sequence を生成する場合、GF(2) の足し算は、排他論理和 **XOR** で計算できる。つまり、

$$0 \ \mathbf{XOR} \ 0 = 0 \, , \quad 0 \ \mathbf{XOR} \ 1 = 1 \, , \quad 1 \ \mathbf{XOR} \ 1 = 0 \, ,$$

となる。また、実行時間を短くするために、 $f(x)$ として 3 項式が用いられることが多い。$f(x) = x^p + x^q + 1, \, p > q > 0$, その場合、$a_n$ は

$$a_n = a_{n-p} \ \mathbf{XOR} \ a_{n-q} \, , \quad p \geq n \, ,$$

で計算される。原始 3 項式の具体例は、計算機を利用していろいろ求められており、詳細な表が得られている。これについては、伏見 [15] やその他の参考文献を参照されたい。

$f(x)$ が原始多項式であっても、a_n の初期値が GF(2^p) の原始根になっていなければ最大周期を達成できないので、通常は $f(x)$ の次数 p として **Mersenne 数**をとる。つまり、素数 p で $2^p - 1$ がまた、素数となるような p をとる。こうすると、$f(x)$ の根はすべて原始根となるので初期値として $(0, 0, \ \ldots \ , 0)$ 以外の任意の元を取ることができる。また、この場合、任意の初期値から始めても、$\{(a_n, \ \ldots \ , a_{n+p-1}) : n \geq 0\}$ は GF(2^p) から $(0, 0, \ \ldots \ , 0)$ を除いた全体に一致することがわかる。これらのことから、以下の性質が導き出される。

m-sequence の性質

- まず、周期は $2^p - 1$ である。
- k-次均等分布する。つまり、

  $$\frac{1}{2^p-1}\,\#\{(a_n,\ \ldots\ ,a_{n+p-1}) : n \geq 0\ ,\ \ a_n = w_1,\ a_{n+1} = w_2,\ \ldots\ ,a_{n+k-1} = w_k\} = \frac{1}{2^k}\ ,$$

  ここで、$w_1, \ldots, w_k$ は任意の 0 , 1 の並びであり、$0 < k < p$ である。

実際に計算機で m-sequence を生成する場合には、計算機の整数で考え、各 bit 毎に排他論理和で計算する。伏見 [15] では擬似乱数として望ましい、各 bit 間の独立性などを確保するために、初期値 の設定の仕方について詳しい議論を展開しているので参考にされたい。

2.3 加算生成法

m-sequence と類似の漸化式による生成法として、以下の加算生成法が古から知られている。

m-sequence の場合と同じように、 GF(2) 上の原始多項式を考えるのであるが、実際には計算速度を速くするために、原始 3 項式を用いるので、ここでもそれに倣う。

$f(x) = x^p + x^q + 1,\ p > q > 0,$ を GF(2) 上の原始 3 項式とする。整数列 x_n を

$$x_n = x_{n-p}\ +\ x_{n-q}\ ,\quad (\mathrm{mod}\ M)\ ,\ \ n \geq p\ ,$$

で定める。ここで、初期値 $x_0, \ldots, x_{p-1}$ はすべてが 0 、とはならないように任意に与える。また、M は通常 2^w とする。ここで、 w は計算機の 1 語の bit 長である。このようにすると、 $(\mathrm{mod}\ 2^w)$ の計算がオーバーフローにより自動的に行われるので実行速度の面で有利になる。x_n の最下位 bit はその生成法より m-sequence であり、上位 bit は下位 bit からの桁上がりがあるのでよりランダムネスが増すことが期待される。このことについては random walk 検定のところで再度触れることにする。また、周期は $l(2^p-1)$, で $0 < l \leq 2^w$ である。Knuth [22] を参照のこと。また、W.H. Press, S.A. Teukolsky, W.T. Vetterling, and B.P. Flannery [32] では、加算の代わりに減算を用いているが、実際上はほとんど変わりは無いように思われるが、彼らが何故減算を用いるか、については不明である。彼らが引用している Knuth [22] では、加算を用いている。

この生成法は Galois 体 GF(2) およびその拡大体 GF(2^p) の理論に基づくのでは、Galois 環 GR(2^w) の理論に基づくものであるが、この理論は Galois 体の理論にくらべて非常に難しく、ほとんど詳しいことが分かっていないのが現状でのようである。

2.4 Cellular Automaton 擬似乱数

この方法は数式処理ソフトウェア、 Mathematica 、の開発者として名高い S. Wolfram によって提唱された生成法である。その名前のように cellular automaton の手法を用いる。

0 , 1 からなる bit 列 a_n , $-\infty < n < \infty$ を用意し、次世代の bit 列 a'_n を以下のように定める：

$$a'_n = \phi(a_{n-r},\ \ldots\ ,a_{n+r})\ ,\ \ -\infty < n < \infty\ ,$$

ここで、$r > 0$ である。そして次に、新世代の数列 a'_n でもとの数列 a_n を同時に置き換える。実際の計算機では無限列は取り扱えないので、$0 \leq n \leq N$ などのように有限列とし、$a_0 = a_N$ などとして周期的な数列を考える。

S. Wolfram [52] では、いろいろな検定を行い、N が十分大きければ、例えば、 $N > 100$ 、十分によい擬似乱数が得られるとしている。

2.5 具体的な擬似乱数生成法の例

以下に幾つかの擬似乱数生成法の例を挙げるが、これらは単に例として挙げるもので、必ずしも優れた擬似乱数生成法と例示するわけではないことに注意されたい。また、各々の生成法で使用しているパラメーターは比較的小さなものを利用しているが、これは後述するように、パラメーターが小さい場合には、比較的短い random walk の見本関数の汎関数での検定で棄却されやすいからである。

- **(A) M89T38**　m-sequence の例として、原始 3 項式 $f(x) = x^{89} + x^{38} + 1$ に基づくものを挙げる：$a_n \in \mathrm{GF}(2)$ 、つまり、 $a_n = 0, 1$ とし、

$$a_n = a_{n-89} + a_{n-38} \ , \quad n \geq 89 \ .$$

  伏見 [15] などでは、$\deg f(x)$ は 127 以上位が適当としているので、 89 では小さいのであるが、ここでは、単に m-sequence の例として挙げているのであり、この生成法を推薦しているのではないことに注意されたい。伏見 [15] では、$f(x) = x^{521} + x^{32} + 1$ などを挙げている。

- **(B) 加算生成法**　上記の原始 3 項式を用いてもよいが、歴史的によく知られた、以下の生成法を挙げる：原始 3 項式として $f(x) = x^{55} + x^{24} + 1$ を用いる方法で

$$a_n = a_{n-55} + a_{n-24} \pmod{2^w} \ , \quad n \geq 55 \ .$$

  ここで w は 16 , 32 などである。Knuth [22] , W.H. Press, S.A. Teukolsky, W.T. Vetterling, and B.P. Flannery [32] などを参照のこと。

- **(C) Cellular Automaton 擬似乱数** $N = 120$, として、以下のように次世代の bit 列 a'_n を現世代の bit 列 a_n から生成する：

$$a'_n = a_{n-1} \ + \ x_n \ + \ x_{n+1} \ + x_n \ x_{n+1} \pmod{2} \ , \ n = 0, \ \ldots \ , N \ ,$$

  但し、$a_0 = a_N$ として周期的に考えるものとする。その後で、新世代の bit 列 a_n で現世代の bit 列 a_n を置き換える。$N = 120$ であるから、 S. Wolfram [52] によれば十分な列の長さ、と言えよう。

3 Monte Carlo 法

擬似乱数を用いて数値積分の近似値を求めたり、偶然現象のシミュレーションを行うことを一般的に Monte Carlo 法と呼ぶ。 Monte Carlo 法は歴史的には、 Buffon の針の実験に始まるとされる。これは以下のような実験である :

床の上に等間隔 d で平行線を数本引く。長さ d の針を N 本用意して、床の上に散まく。平行線と交差した針の本数 m を数え、$p_N = \frac{m}{N}$ を計算する。N を大きく取ると p_N は $\frac{2}{\pi}$ に近づくことが分かる。このことの理論的な証明については、ベックマン [9] などを参照されたい。

もう少し、数学的な例を考えてみよう。R^2 の単位正方形 $[0,1]\times[0,1]$ を考え、擬似乱数 x_n , y_n , $0 \le x_n,\ y_n < 1$, $n = 1, \dots$, N を生成し、点 (x_n, y_n) を $[0,1]\times[0,1]$ 上に散まき、

$$m_N = \#\{n : 1 \le n \le N,\ \ x_n^2 + y_n^2 \le 1\}$$

を計算する。

ここで、確率変数 X_n , Y_n はそれぞれ区間 $[0,1]$ 上に一様分布し、互いに独立である、と仮定すると、$0 \le a < b \le 1$, $0 \le c < d \le 1$, に対して

$$\Pr(X_n \in [a,b)) = b - a\ ,\ \ \Pr(Y_n \in [c,d)) = d - c\ ,$$

が成り立つ。これより、

$$\Pr(X_n \in [a,b),\ \ Y_n \in [c,d)) = (b-a)\times(d-c)\ ,$$

となることが分かるので、

$$E[\chi_D(X_n, Y_n)] = |D| = \frac{\pi}{4}\ ,$$

となることが、領域 $D = \{(x,y) : 0 \le x, y \le 1, x^2 + y^2 \le 1\}$ を長方形に細分して考え、その細分する細かさを 0 に近づける極限を考えるこより導かれる。これらのことについては測度論における、直積測度の構成、の議論を思い出してほしい。但し、$|D|$ は領域 D の面積を表す。

このような確率論の議論を用いると、上記の擬似乱数の組 (x_n, y_n) が D に落ちる個数 m_N と、散まく点の総数 N の比は、$N \to \infty$ の時、 D の面積 $\frac{\pi}{4}$ に近づく、即ち、

$$\lim_{N\to\infty}\ \frac{m_n}{N}\ =\ \frac{\pi}{4}\ ,$$

となることが確率論において、大数の法則、と知られている定理より分かる。

つまり、 ξ_n, $n = 1, 2$, $\dots$, を互いに独立な確率変数で同じ分布に従うものとし、平均 $E[\xi_n] = \mu$, 分散 $Var(\xi_n) = \sigma^2$, とする。更に、簡単のために、$E[|\xi_n|^3] < \infty$ を仮定すると、

$$\Pr(\lim_{n\to\infty}\ \frac{1}{n}\ \sum_{k=1}^{n}\ \xi_k\ =\ \mu\ ) = 1,$$

となる (大数の強法則)。

上記の例では、$\xi_n = \chi_D(x_n, y_n)$ とおけばよい。但し、$\chi_D(x, y)$ は、もしも $(x, y) \in D$ なら 1 , そうでなければ 0 を値とする関数である。この例から分かることは、擬似乱数の組 (x_n, y_n) を平面上に散まくことにより、平面図形の面積の近似値を求めることができることである。これは、多次元の場合においても同様である。

次に、近似値の極限値への収束の速さについて考えてみよう。この問題に関しては、確率論でよく知られた、中心極限定理がある：

再び、 ξ_n, $n = 1, 2$, ... , を互いに独立な確率変数で同じ分布に従うものとし、平均 $E[\xi_n] = \mu$, 分散 $Var(\xi_n) = \sigma^2$, とする。更に、簡単のために、$E[|\xi_n|^3] < \infty$ を仮定すると、任意の $-\infty < a < b < \infty$ に対して、

$$\lim_{n\to\infty} \Pr(\ a\ <\ \frac{\sum_{k=1}^{n}\ \xi_k\ -\ n\ \mu}{\sqrt{n}\sigma}\ <\ b\ ) = \int_a^b\ \frac{1}{\sqrt{2\pi}}\ \ \exp\{-\frac{x^2}{2}\}\ dx$$

となる。また、同じことであるが、

$$\lim_{n\to\infty} \Pr(\ a\ <\ \frac{\bar{\xi}\ -\ \mu}{\frac{\sigma}{\sqrt{n}}}\ <\ b\ ) = \int_a^b\ \frac{1}{\sqrt{2\pi}}\ \ \exp\{-\frac{x^2}{2}\}\ dx$$

ここで、 $\bar{\xi}\ =\ \frac{1}{n}\ \sum_{k=1}^{n}\ \xi_k$ 、即ち、 ξ_n の標本平均である。

ここで左辺の積分は、平均 0 、分散 1 、の標準正規分布の積分であるが、この値は、例えば $a = -3.0$, $b = 3.0$ とすると、 0.99 以上の値になることが知られていることに注意する。このことより、この場合、右辺の確率は 0.99 より大きくなり、

$$-3.0\ \frac{\sigma}{\sqrt{n}}\ <\ \bar{\xi}\ -\ \mu\ <\ 3.0\ \frac{\sigma}{\sqrt{n}}\ ,$$

となることが、確率 0.99 以上で期待できることになる。このことから、標本平均 $\frac{1}{n}\ \sum_{k=1}^{n}\ \xi_k$ と平均 μ との誤差は $O(\ \frac{1}{\sqrt{n}})$ 程度の大きさになることが期待される。従って、 Monte Carlo 法で数値積分の近似値を求める場合には、近似の収束の速さは $O(\ \frac{1}{\sqrt{n}})$ 程度となり、これは、通常の数値積分の方法、例えば台形法や Shimpson 法などに比べるとかなり収束が遅い、ことが分かる。然しながら、よく知られているように、通常の数値積分の方法では、積分変数の次元が大きくなると、計算速度が次元の指数関数的に増大し、例えば 20 次元程度で実際上は計算不可能になるのに対して、 Monte Carlo 法では、次元に比例する程度の増大になるので、数百次元以上になっても計算時間が極度に増大しない。最近の数値計算では、変数の次元が大きな場合が多く、そのような場合には Monte Carlo 法が唯一の手段である場合が多いのである。

4 統計的検定について

4.1 擬似乱数に対する統計的検定

前述したように、擬似乱数は計算機で生成する、**乱数モドキ** 、であるので、個々の擬似乱数列 x_n を真の意味で、独立同分布な確率変数列 X_n の代わり、と信ずることはできない。そこで、実際の擬似乱数列がどの程度 **ランダム** であるのか、調べる必要が生じることになる。ところで、擬似乱数の理論は有限代数の理論に基づいていた。特に、線形合同法や m-sequence は理論的な多くの研究がなされている。例えば、第 2.2 節で見た m-sequence の性質などは、周期全体に関するものであった。このように、今までのところ擬似乱数の性質は一周期全体にわたるものがほとんどである。然しながら、擬似乱数の一周期全体を利用すると、擬似乱数の生成の仕方から、ランダムな数の列、と見なすことが出来なくなる。そこで、実際に擬似乱数を利用する場合には、一周期の一部分を利用するように配慮する必要がある。一方では、上で見たように、理論的研究は一周期全体にわたるものがほとんどであり、部分的な数列の性質はほとんど分かっていないのが実情である。

このような実態から、一周期の一部分を利用する実態に即した、擬似乱数のランダムネスの検討が必要になるのは当然といえよう。これを行うのが、統計的検定である。ここでは、 Knuth [22] 、伏見 [15] などに基づいて、擬似乱数の統計的検定方法の古典的なものを簡単に解説する。より詳しいことは上記の参考文献などを参照されたい。

- **ポーカー検定**　これは、その名の通り、ポーカーの手になぞらえて、 5 個の擬似乱数の値を

  1. **AAAAA**　同じ値が 5 回続いて起きる。
  2. **AAAAB**　同じ値が 4 回起き、 1 回は異なる値が起きる。
  3. **AAABB**　同じ値が 3 回、別の値が 2 回起きる。
  4. **AAABC**　同じ値が 3 回起きる。
  5. **AABBC**　同じ値が 2 回、別の値が 2 回起きる。

  などのように分類し、それぞれの出現頻度を計算し、理論分布と比較するものである。実際に比較するときには、後述する、適合度の検定、例えば χ^2 検定、などを利用する。ここで、同じ値、としたのは実際には生成された擬似乱数の上位の数 bit などに注目した値、のことである。線形合同法などでは全く同じ値が続いて起きることはあり得ない。

- **連の検定**　擬似乱数の値が $x_k < x_{k+1}$ となる場合、上昇連、とよび、 $x_k > x_{k+1}$ となる場合、下降連、と呼ぶ。この検定は擬似乱数列の中の上昇連、下降連の長さを計算し、理論分布と比較するものである。

- **衝突検定**　簡単のために、 1 次元の場合に考える。区間 $[0, M]$ を幾つかの小区間に分ける。ここで、M は擬似乱数の最大値とする。擬似乱数列のうち、同じ小区間に入るものがある場合、衝突が起きた、と考える。分割数が、擬似乱数の個数に比べ

て、はるかに大きい場合、衝突の起きる確率は小さいことが期待されるが、実際に観測される衝突の回数を理論分布と比較する検定である。

これらの他にも、数多くの統計的検定が提案され、実際に利用されているが、ここでは、それらの多くは、確率過程や偶然現象などのシミュレーションにおける擬似乱数の利用を考慮したものとは異なるもの、と思われることを注意するにとどめることにする。即ち、確率過程のシミュレーションなどでは一つの統計量、例えば原点へ戻ってくるまでの時間等、を計算するのに、長さの長い、また多くの場合、長さの決まった、擬似乱数の列が必要であるが、上記の検定では比較的短い数列を対象にしていたり、長さを固定した数列を対象に出来ない、などの点が指摘されよう。このようなことから、確率過程や偶然現象のシミュレーションなどへの擬似乱数の利用を踏まえた、統計的検定を考えるのは当然の成り行き、と言えよう。

なお、擬似乱数の検定として、スペクトル検定、が有名であり、強力な検定として知られているが、この検定は、主に線形合同法の『粗結晶構造』を検出するもので、 m-sequence やその他の擬似乱数生成法の検定としては有用でない、と思われるので本報告では詳しく述べないことにする。 Knuth [22] 、伏見 [15] などを参照されたい。

4.2 適合度の検定

前節で述べた、擬似乱数に対する統計的検定を行う場合、理論分布との比較をする必要があり、適合度の検定、と呼ばれる統計的検定が利用される。ここでは、適合度の検定の中で、擬似乱数の検定に比較的よく利用される、χ^2 検定と Kolmogorov - Smirnov 検定について、簡単に述べることにする。適合度の検定としての χ^2 検定はいろいろな統計の教科書に載っているのでそれらを参照されたい。また、 Kolmogorov - Smirnov 検定は教科書的な統計の本にはあまり見かけないが、例えば、ホーエル [19] 、伏見 [15] 、Knuth [22] などを参照されたい。いずれの検定も、ノンパラメトリック検定、と呼ばれる、分布の型を仮定しない検定方法の例である。ノンパラメトリック検定は分布の型を仮定する検定理論に比べて難しい、とされているようである。

4.2.1 χ^2 検定

離散的な値 c_k , $k = 1, \ \ldots \ , K$, をとる確率分布を考える。観測値 x_i , $i = 1, \ \ldots \ , N$, の各値に対する観測度数 o_k , $k = 1, \ \ldots \ , K$

$$o_k \ = \ \#\{i : x_i = c_k \ , \ i = 1, \ldots , N \ \} \ ,$$

を考える。χ^2 検定はこの観測度数が理論分布 f_k , $k = 1, \ \ldots \ , K$ に従う分布と見なせるかどうかを統計的に検定するものである。具体的には、次の統計量を計算する :

$$\chi_0^2 \ = \ \sum_{k=1}^{K} \ \frac{(o_k - N \ f_k)^2}{N \ f_k} \ .$$

この統計量 χ_0^2 は漸近的に、自由度 $(K-1)$ の χ^2 分布に従うことが知られている（ホーエル [19] 参照)。漸近理論に基づくため、例えば各期待度数 $N\ f_k$ が 5 以上（或いは 10 以上、という説もある）あれば十分な近似が得られる、というような議論がある。また、離散分布でなく連続分布に基づく場合には、可能な値の範囲を適当に分割し、各小区間に入る観測値の観測度数と、期待度数を考える必要がある。

この検定はまた、分割表などに用いて、独立性の検定などにも広く利用されている。

4.2.2 Kolmogorov-Smirnov 検定

前節で述べた χ^2 検定が基本的に離散分布に関する理論に基づいた、漸近理論であったのに対して、 Kolmogorov - Smirnov 検定は連続分布に基づいた検定である。

まず、観測値 $x_i\ , i = 1,\ \ldots\ , N$ に基づいて経験分布 $F_N(x)$ を求める：

$$F_N(x)\ =\ \frac{1}{N}\ \#(\{x_i : x_i \leq x\ , 1 \leq i \leq N\ \})\ ,\quad -\infty < x < \infty\ .$$

次に、観測値が理論分布関数 $F(x)$ を持つ母集団からの標本と見なせるかどうかを以下の統計量を基に検定する：

$$K_N^+\ =\ \sqrt{N}\ \sup_{-\infty<x<\infty}(F_N(x)\ -\ F(x))\ ,$$

$$K_N^-\ =\ \sqrt{N}\ \sup_{-\infty<x<\infty}(F(x)\ -\ F_N(x))\ .$$

この統計量の理論分布は random walk の破産問題を解くことによって求めることができる（例えば Feller [13] を参照のこと）が、通常は分布表から自由度 N の分布の tail probability を求めることになる。上記の Kolmogorov - Smirnov 統計量 $K_N^+\ ,\ K_N^-$ はまた次のように計算することもできる：

観測値 $x_i\ , i = 1,\ \ldots\ , N$ を大きさの順に並べ換えたものを $x_{(i)}\ , i = 1,\ \ldots\ , N$ とする。

$$K_N^+\ =\ \sqrt{N}\ \sup_{i=1,\ \ldots\ ,N}\left(\frac{i}{N}\ -\ F(x_{(i)})\right)\ ,$$

$$K_N^-\ =\ \sqrt{N}\ \sup_{i=1,\ \ldots\ ,N}\left(F(x_{(i)})\ -\ \frac{i-1}{N}\right)\ .$$

これは、観測分布関数と理論分布関数の誤差は、各 $x_{(i)}$ の前後で最大になる、ことから導かれる。

χ^2 検定にしても、 Kolmogorov - Smirnov 検定にしても、一度に大量の標本を検定するよりは、小さなグループに分けて検定し、それらの検定結果を再度まとめて検定をする、という検定の積み重ねが、部分的な変動を捕らえやすいことが知られている (cf. Knuth [22]) 。

5 1-dimensional simple symmetric random walk

本報告では、確率過程や偶然現象のシミュレーションに対する擬似乱数の利用という状況に適した統計的検定の 1 方法として random walk の見本関数の汎関数を用いた検定法を提案する。この方法では、最も単純かつ基本的な random walk である 1-dimensional simple symmetric random walk を考え、その見本関数の汎関数を考察する。以下、簡単に random walk と呼ぶことにする。Random walk の見本関数については多くの研究がなされ、詳しい結果が知られているが、ここでは、 Feller [12] を参考に知られている結果をまとめることにする。

5.1 Random walk の定義

まず、以下のような独立同分布確率変数列 X_n , $n = 1, 2, \ldots$ を考える :

$$\Pr(X_n = 1) \;=\; \Pr(X_n = -1) \;=\; \frac{1}{2}\;,\quad n = 1, 2, \ldots\;.$$

次に、確率変数 S_n を以下のように定義する :

$$S_0 = 0\;,\quad S_n \;=\; \sum_{k=1}^{n} X_k \;\;,\;\; n \;>\; 0\;.$$

この S_n , $n \geq 0$ を原点から出発する 1-dimensional simple symmetric random walk, 単に random walk , と呼ぶことにする。そして、点列 (k, S_k) , $k = 0, 1, 2, \ldots$ を折れ線で結んだグラフを、 random walk の見本関数 (sample path) と呼ぶ。また、 n を時間と見なし、時刻 N までで考えるとき、N を見本関数の **長さ** とも呼ぶことにする。

Random walk や random walk の見本関数の性質を調べる時、重要な手段となる、反射原理、について以下に述べることとする。

5.2 反射原理 reflection principle

これは、random walk が値 +1 , -1 を確率 $\frac{1}{2}$ で取る独立同分布な確率変数列の部分和であることから導かれる性質であり、単純な結果であるが、random walk の見本関数の性質を導く上で、重要な働きをするものである。random walk の見本関数のグラフのことを、折れ線、と呼ぶことにする。

定理（反射原理）　$a > 0, 0 < m < n$ とする。点 (m, a) から点 (n, b) に至る折れ線の中、横軸に到達する折れ線の本数は、点 $(m, -a)$ から点 (n, b) に至る折れ線の本数に等しい。

証明　点 (m, a) から出発する折れ線が初めて横軸に到達する時刻を k, $m < k < n$ とする。点 (m, a) から点 $(k, 0)$ に至るまでの部分を、横軸に関して対称に折り返すと、点 $(m, -a)$ から点 $(k, 0)$ に至る折れ線ができる。また、明らかにこの対応は 1 対 1 対応である。このことより、点 (m, a) から点 (n, b) に至る折れ線の中、横軸に到達する折れ線の本数は、点

$(m, -a)$ から点 (n, b) に至る折れ線の本数に等しいことが分かる。（証明終）

この反射原理から導かれる基本定理を述べるために、また、見本関数の汎関数の確率分布を述べるために、幾つかの記号を準備する。

まず、整数 n , r に対して、

$$p_{n,r} \;=\; \Pr(S_n = r) \;=\; \left(\begin{array}{c} n \\ \frac{n+r}{2} \end{array} \right) \, 2^{-n} \;\; ,$$

とおく。ここで 2 項係数 $\left(\begin{array}{c} n \\ \frac{n+r}{2} \end{array} \right)$ は $\frac{n+r}{2}$ が整数にならない場合は 0 とするものとする。

$S_n = 0$ となるのは n が偶数の場合に限られる。そこで、

$$u_{2k} \;=\; \Pr(S_{2k} = 0) \;=\; \left(\begin{array}{c} 2k \\ k \end{array} \right) \, 2^{-2k} \;,$$

とおくことにする。有名は Stirling の公式 :

$$n! \;\sim\; \sqrt{2\pi} \; n^{n+\frac{1}{2}} \; e^{-n} \;, \quad \text{as } k \;\rightarrow\; \infty \;,$$

を用いると、

$$u_{2k} \;\sim\; \frac{1}{\sqrt{\pi k}} \;, \quad \text{as } k \;\rightarrow\; \infty \;,$$

が分かることに注意しておく。次に、基本的な定理を述べる。

定理

$$\Pr(S_1 \neq 0, \; \ldots \; , \; S_{2n} \neq 0) \;=\; \Pr(S_{2n} \;=\; 0) \;=\; u_{2n} \;.$$

$$\Pr(S_1 > 0, \; \ldots \; , \; S_{2n} > 0) \;=\; \frac{1}{2} \, u_{2n} \;.$$

証明　まず、 2 番目の式から示す。

$$\Pr(S_1 > 0, \; \ldots \; , \; S_{2n} > 0) \;=\; \sum r = 1^{\infty} \; \Pr(S_1 > 0, \; \ldots \; , \; S_{2n-1} > 0, \; S_{2n} = 2r) \;.$$

ここで、点 $(1, 1)$ から点 $(2n, r)$ への見本関数で、横軸に到達するものの数は反射原理より、点 $(1, -1)$ から点 $(2n, r)$ への見本関数の数に等しいことに注意すると、

$$\Pr(S_1 > 0, \; \ldots \; , \; S_{2n-1} > 0, \; S_{2n} = 2r) \;=\; \frac{1}{2} \; (p_{2n-1,2r-1} - p_{2n-1,2r+1})$$

となることが分かる。これを r について可能な範囲で和をとれば、第 2 の結果が得られる。第 1 の結果は

$$\Pr(S_1 \neq 0, \; \ldots \; , \; S_{2n} \neq 0) \;=\; \Pr(S_1 > 0, \; \ldots \; , \; S_{2n} > 0) \;+\; \Pr(S_1 < 0, \; \ldots \; , \; S_{2n} < 0) \;,$$

であることに注意し、 X_k の分布の対称性から

$$\Pr(S_1 > 0, \; \ldots \; , \; S_{2n} > 0) \;=\; \frac{1}{2} \, u_{2n} \;=\; \Pr(S_1 < 0, \; \ldots \; , \; S_{2n} < 0) \;,$$

より導かれる。(証明終)

5.3 見本関数の汎関数

まず、簡単な汎関数から考えていくことにする。以下では、長さ $2n$ の見本関数 (k, S_k), $k = 0, 1, 2, \ \ldots \ , 2n$ を考えることにする。

5.3.1 Last visit time

見本関数 (k, S_k), $k = 0, 1, 2, \ \ldots \ , 2n$ に対して、原点への **last visit time** LV_{2n} を

$$LV_{2n} \ = \ \max\{2\,k : S_{2k} \ = \ 0 \ , \ \ 0 < k \leq n \ ,$$

で定義する。この last visit time の理論分布について以下の結果が知られている :

定理　Arc Sine Law

$$\alpha_{2k,2n} \ = \ \Pr(LV_{2n} \ = \ 2\,k) \ = \ u_{2k}u_{2n-2k} \ , \ \ k = 0, 1, \ldots, n \ .$$

証明　$LV_{2n} = 2k$ は $S_{2k} = 0$, $S_{2k+1} \neq 0$, ..., $S_{2n} \neq 0$ と同じであることと、時刻 $2k$ までの見本関数の挙動とそれ以降の見本関数の挙動は独立である (X_j の独立性から導かれる) ことから、

$$\alpha_{2k,2n} \ = \ u_{2k}u_{2n-2k} \ ,$$

となる。(証明終)

注意　ここで、この分布は左右対称であることを注意しておく。実際、

$$\alpha_{2k,2n} \ = \ u_{2k}u_{2n-2k} \ = \ u_{2n-2k}u_{2n-(2n-2k)} \ = \ u_{2k}u_{2n-2k} \ = \ \alpha_{2n-2k,2n} \ .$$

この分布の左右対称性は、独立確率変数列 X_n の対称性 (± 1 を確率 $\frac{1}{2}$ で取ること) に関係することを後述する。

5.3.2 滞在時間　Sojourn time

見本関数 (k, S_k) , $k = 0, 1, 2, \ \ldots \ , 2n$ が正の部分に滞在する滞在時間 **sojourn time** SJ_{2n} を以下のように定義する。

$$I(x) \ = \ \begin{cases} 1 & \text{if} \ \ x \geq 0 \ , \\ 0 & \text{その他} \ \ , \end{cases}$$

と置き、

$$SJ_{2n} \ = \ \sum_{k=1}^{2n} \ I(S_{k-1} + S_k) \ ,$$

で、 sojourn time SJ_{2n} を定義する。 random walk の見本関数は奇数時間では、奇数の値しか取れないことに注意すると、 SJ_{2n} は次の式でも計算できることが分かる。

$$SJ_{2n} \ = \ 2 \ \sum_{k=1}^{n} \ I(S_{2k-1}) \ .$$

Sojourn time の理論分布は以下の定理で与えられる。

定理（ Arc Sine Law ）

$$\alpha_{2k,2n} \;=\; \Pr(SJ_{2n} \;=\; 2\,k\;) \;\;,\quad k = 0, 1, \ldots, n\;.$$

証明　まず、 $b_{2k,2n} = \Pr(SJ_{2n} \;=\; 2k)$ と置き、帰納法を用いる。

$$b_{2k,2\nu} \;=\; \alpha_{2k,2\nu}\;,\;\; k \;=\; 0, 1, \ldots, \nu\;,\;\; \nu < n\;,$$

と仮定する。

$SJ_{2n} = 2k$ とし、原点への first return time が $2r$ であるとする。つまり、

$$S_1 \neq 0, S_2 \neq 0,\; \ldots\; , S_{2r-1} \neq 0, S_{2r} = 0\;,$$

と仮定する。まず、もし、$S_1 > 0, S_2 > 0,\; \ldots\; S_{2r-1} > 0$ なら $r < k$ でなければならない。この場合、残りの $2n - 2r$ 時間で正の部分に $2k - 2r$ 時間滞在することになるから、その確率は $f_{2r} = \Pr(S_1 \neq 0, S_2 \neq 0, \ldots, S_{2r-1} \neq 0, S_{2r} = 0)$ と置くと、

$$\frac{1}{2}\;\sum_{r=1}^{k}\;f_{2r}\;b_{2k-2r,2n-2r}\;,$$

また逆に、$S_1 < 0, S_2 < 0,\; \ldots\; S_{2r-1} < 0$ なら $n - r \leq k$ であり、残りの $2n - 2r$ 時間内に正の部分に $2k$ 時間滞在する必要がある。そこで、その確率は、

$$\frac{1}{2}\;\sum_{r=1}^{n-k}\;f_{2r}\;b_{2k,2n-2r}\;,$$

となる。これらをまとめると、

$$b_{2k,2n} \;=\; \frac{1}{2}\;\sum_{r=1}^{k}\;f_{2r}\;b_{2k-2r,2n-2r} \;+\; \frac{1}{2}\;\sum_{r=1}^{n-k}\;f_{2r}\;b_{2k,2n-2r}\;,$$

となる。従って、仮定より

$$b_{2k,2n} \;=\; \frac{1}{2}\;u_{2n-2k}\;\sum_{r=1}^{k}\;f_{2r}\;u_{2k-2r} \;+\; \frac{1}{2}\;u_{2k}\;\sum_{r=1}^{n-k}\;f_{2r}\;u_{2n-2k-2r}\;,$$

が得られる。ここで、

$$u_{2k} \;=\; \sum_{r=1}^{k}\;f_{2r}\;u_{2k-2r}\;,$$

であることに注意すると、 $b_{2k,2n} \;=\; \alpha_{2k,2n}$ を得る。(証明終)

注意　$u_{2k} \;\sim\; \frac{1}{\sqrt{\pi k}}$ であることに注意すると、

$$\alpha_{2k,2n} \;\sim\; \frac{1}{n}\;\frac{1}{\pi\sqrt{x_k(1-x_k)}}\;,\quad \frac{k}{n}\;,$$

となることが分かる。このことより、$0 < x < 1$ に対して、

$$\sum_{k<xn} \alpha_{2k,2n} \sim \frac{2}{\pi} \arcsin \sqrt{x} ,$$

が十分大きな n に対して成り立つことが分かる。これが、 **Arc Sine Law** の名前の由来である。

5.3.3 最大値 Maximum

$MX_n = \max\{S_k : k = 0, 1, \ldots, n\}$ とおく。この最大値の理論分布は以下で与えられる。

定理

$$\Pr(MX_n = r) = p_{n,r} + p_{n,r+1} \quad r \geq 0 .$$

証明　まず、定理の式の中の $p_{n,r}$, $p_{n,r+1}$ は同時にはどちらか一方だけが正の値を取り、他方は 0 となることに注意する。

$k \leq r$ とする。原点から出発して点 (n, k) への見本関数の中、直線 $x = r$ に到達する見本関数の数は、反射原理より、原点から出発して点 $(n, 2r - k)$ に至る見本関数の数に等しいことが分かる。このことより、

$$\Pr(MX_n \geq r \ , \ S_n = k) = \Pr(S_n = 2r - k) = p_{n,2r-k} .$$

従って、

$$\Pr(MX_n = r \ , \ S_n = k) = p_{n,2r-k} - p_{n,2r+2-k} .$$

これを $k \leq r$ に関して足すと、

$$\Pr(MX_n = r) = p_{n,r} + p_{n,r+1} ,$$

となる。(証明終)

5.3.4 Hamming weight

この汎関数は random walk の見本関数の性質と言うよりは、むしろ符号理論で重要な汎関数を random walk に持ち込んだものである。

$$HW_n = \frac{1}{2} \sum_{k=1}^{n} (X_k + 1) ,$$

で Hamming weight HW_n を定義する。ここで、X_k は random walk を構成する時に使う、独立同分布確率変数列である。$X_k = \pm 1$ であるので、 $\frac{1}{2}(X_k + 1)$ は 0, 1 の値に等しいことに注意する。後に、擬似乱数に対する統計的検定をおこなう場合にはこの汎関数は、擬似乱数の最上位 bit の中の 1 の個数、として計算される。この Hamming weight の理論分布は明らかに 2 項分布に従うことが分かる。

$$\Pr(HW_n = k) = \left(\begin{array}{c} n \\ k \end{array} \right) 2^{-n} .$$

5.3.5 汎関数間の依存性

後の節で見るように、random walk の見本関数の汎関数を用いて擬似乱数の randomness の統計的検定を行うのであるが、その際、検定に用いる汎関数同志が互いに従属している場合には、例えば、sojourn time と last visit time との間に強い従属性がある場合には、それらの汎関数を用いて 2 通りの検定を行うことはあまり意味がないであろう。そこで、各汎関数間の依存性を調べることが重要になってくる。この問題に関しては Takashima [40] で sojourn time と last visit time との依存性が調べられ、石川、高嶋 [49] ではその他の汎関数について調べられている。詳しくはこれらを参照されたい。ここでは、簡単に、sojourn time と last visit time との同時分布に関する結果について述べるにとどめる。

定理

$$\Pr(SJ_{2n} = 2j, LV_{2n} = 2k) \;=\; \frac{I(j+k-n) + I(k-j)}{2(k+1)}\;\; u_{2k}\; u_{2n-2k}\;,\;\; 0 \le j,k \le n\;.$$

証明　まず、有名な Equidistribution theorem に注意する。

$$\Pr(SJ_{2n} = 2j, S_{2n} = 0) \;=\; \frac{1}{n+1}\; u_{2n}\;,\;\; 0 \le j \le n\;.$$

この結果を用いると

$$\begin{aligned} & \Pr(SJ_{2n} = 2j, LV_{2n} = 2k, S_i > 0\ 2k < i \le 2n) \\ = \;& \Pr(SJ_{2n} = 2j + 2k - 2n, S_{2k} = 0)\frac{1}{2}u_{2n-2k} \\ = \;& \frac{u_{2k}u_{2n-2k}}{2(k+1)}\;,\;\; j \ge n-k\;, \end{aligned}$$

が得られる。同様に

$$\begin{aligned} & \Pr(SJ_{2n} = 2j, LV_{2n} = 2k, S_i < 0\ 2k < i \le 2n) \\ = \;& \Pr(SJ_{2n} = 2j, S_{2k} = 0)\frac{1}{2}u_{2n-2k} \\ = \;& \frac{u_{2k}u_{2n-2k}}{2(k+1)}\;,\;\; j \le k\;. \end{aligned}$$

これらをまとめると、定理の結果が得られる。（証明終）

この結果より、sojourn time と last visit time は互いに独立ではないが、共分散は 0 であることが分かる。何故なら、同時分布が左右対称であるから。

6 Random walk 検定について

前節で述べた、 random walk の見本関数の汎関数を利用する、擬似乱数に対する統計的検定について述べる。

基本的に random walk のどの見本関数を用いても、検定の仕方は同様であるので、sojourn time を用いた検定について述べることにする。

1. まず、擬似乱数を初期化する。m-sequence や加算生成法、cellular automaton 擬似乱数などでは、複数の初期値を用意しなければならないので、別の擬似乱数生成法、例えば線形合同法など、を利用して初期化する。

2. 以下の step を M 回繰り返す:

   (a) 擬似乱数 x_i, $i = 1, 2, \ldots, 2L$ を生成し、random walk の見本関数

   $$s_0 = 0, \quad s_n \;=\; \sum_{k=1}^{n} x_k^* \;\;, \quad n = 1, 2, \ldots, 2L,$$

   を構成する。ここで、 x^* は整数 x の最上位 bit を表す。

   (b) 見本関数に対して、汎関数 sojourn time SJ_{2L}^j を計算する :

   $$SJ_{2L}^j \;=\; 2 \sum_{k=1}^{L} \; I(x_{2k-1}^*) \;\; j = 1, 2, \ldots, M \;.$$

   さらに、これらの観測値の経験分布を求める :

   $$o_i \;=\; \#\{j : SJ_{2L}^j = 2\,i \;,\; j = 1, 2, \ldots, M \;\; i = 0, 1, \ldots, L \;,$$

3. Step 2 で求めた、観測度数 o_i, $i = 0, 1, \ldots, L$ に対して sojourn time の理論分布に関する適合度検定を χ^2 検定を用いておこない、 χ^2 統計量

   $$\chi_0^2 \;=\; \sum_{i=0}^{L} \; \frac{(o_i - e_i)^2}{e_i} \;.$$

   ここで、 e_i は sojourn time の値が $2\,i$ になる期待度数である。これは、第 節で求めた確率 に M を掛けたものである。

4. 上記の Step 2 , 3 を例えば 30 回繰り返し、得られる χ^2 統計量を Kolmogorov - Smirnov 検定を用いて、自由度 L の χ^2 分布に関する適合度の検定をおこなう。

   $$K_{30}^{+} \;=\; \sqrt{30} \; \sup j = 1^{30} \; \left(\frac{j}{30} - F(\chi_{(j)}) \right) \;,$$

   $$K_{30}^{-} \;=\; \sqrt{30} \; \sup j = 1^{30} \; \left(F(\chi_{(j)}) - \frac{j-1}{30} \right) \;.$$

   ここで、 $\chi_{(j)}$ は 30 個の χ^2 統計量を大きさの順に並べ直したものである。

5. 上記の Step 4 を例えば 100 回繰り返し、得られる Kolmogorov - Smirnov 統計量 K_{30}^{+}, K_{30}^{-} で、99 % 以上の範囲入る個数と 95 - 99 % の範囲入る個数を求める。

100 回の Kolmogorov - Smirnov 検定の結果の中、 95 - 99 % , 99 % 以上の各範囲に入る個数が期待される個数よりかなり大きければこの擬似乱数は sojourn time 検定で棄却される、とする。sojourn time は見本関数の長さが偶数の場合、偶数の値しか取らないので χ^2 検定の自由度は L となることに注意する。他の汎関数を用いる場合は、自由度は異なる。また、 maximum などを用いる場合には、 maximum が大きな値をとる確率は非常に小さくなるので、いくつかの観測度数を合併して期待度数が十分大きくなるようにする必要が生じることにも注意する必要がある。

7 Random walk 検定の結果について

前説で述べた random walk の見本関数の汎関数を用いる統計的検定の結果を表 1、表 2、表 3 に載せる。これらの表から分かることは、第 2.5 節に挙げた擬似乱数生成法はいずれも、random walk 検定に失敗している、という事実である。

特に、 sojourn time による検定について考えてみよう。図 1 は、 m-sequence **M89T38** を用いて sojourn の経験分布と理論分布をグラフにしたものである。長さ 300 の見本関数を 100,000 本シミュレーションして sojourn time の観測度数を折れ線で表したものである。この図より明らかに、経験分布の密度関数は左右対称でなく、理論分布の密度関数に比べて、平均 150 より小さな部分（図中央より左側）では観測度数が理論分布の密度関数より小さな値をとり、逆に、平均より大きな部分（右側）では理論分布の密度関数より大きな値を取っている。

また、 Cellular automaton 擬似乱数の検定結果も興味深いものがある。Wolfram [52] によれば、配列の長さが 100 以上あれば十分に' 良い' 擬似乱数が得られる、としているが、本報告の検定結果は random walk などの確率過程のシミュレ-ションには適していない、という結果を示している、と言ってよいと思われる。[52] をうけて、 Engel [11] のような教科書的な文献にもこの方法は紹介されいるので注意が必要である。

本来、 m-sequence では、0 , 1 の出現頻度は $\frac{1}{2}$ にほとんど等しく、sojourn time の観測度数が左右対称になることが期待されるはずであるが、実際のシミュレーションでは図 1 、図 2 などの用にように左右非対称になる。この現象を調べるために、次節以降で Hamming weight と sojourn time の関係について考えてみる。

7.1 Sojourn time の漸近分布

前節で述べた、 m-sequence による sojourn time 検定の結果について考察するために、本節では、今まで考えてきた、 simple symmetric random walk だけでなく、より一般的な random walk を考える。本節の詳しい内容等については、 Andersen [1] , [2] , Spitzer [34] , [35] 等を参照されたい。

まず、独立確率変数列 (通常、確率論では triangular array と呼ばれる) を考える :

$$X_1^{(n)}, X_2^{(n)}, \ \ldots \ , X_n^{(n)} \ , \ n > 0 \ .$$

各 $X_1^{(n)}, X_2^{(n)}, \ \ldots \ , X_n^{(n)}$ は同じ分布 (必ずしも対称とは限らない) に従い、必ずしも整数値をとるとは限らないものとする。また、異なる n に関して確率変数列の分布は変化し得るものとする。この独立確率変数列に対して、 random walk $S_k^{(n)}$ を

$$S_k^{(n)} \ = \ \sum_{k=1}^{n} \ X_k^{(n)} \ ,$$

で定義する。

Andersen [1] , [2] はこのような一般的な random walk に対して、 sojourn time の漸近分布と $S_n^{(n)} > 0$ となる確率の極限との関係を与えている。

定理

$$\lim_{n\to\infty} \Pr(S_n^{(n)} > 0) = \alpha \ ,$$

とおく。このとき、

$$\lim_{n\to\infty} \Pr(SJ^{(n)} < x\, n) \ = \ \frac{\sin \alpha\pi}{\pi} \ \int_0^x \ t^{\alpha-1} \ (1-t)^{-\alpha} \ dt \ , 0 < x < 1 \ .$$

ここで、 $SJ^{(n)}$ は $S_n^{(n)}$ についての sojourn time である。またこの結果の右辺に現れる積分は、 β 分布と呼ばれる確率分布の場合である。

ここで、前節で扱った simple symmetric random walk の場合に戻ると X_n の確率分布の対称性、即ち、

$$\Pr(X_n \ = \ 1) \ = \ \Pr(X_n \ = \ -1) \ = \ \frac{1}{2} \ , \quad n = 1, 2, \ \ldots \quad ,$$

より、明らかに $\alpha = \frac{1}{2}$ である。従って、理論分布の密度関数のグラフは左右対称なグラフになり、 α の値が $\frac{1}{2}$ より大きい場合には、密度関数はグラフの中心より左側で $\alpha = \frac{1}{2}$ の場合より下がり、右側で上がる、ことになる。図 1 、図 2 のシミュレーションの結果のグラフは同様の傾向を示していることに注意する。

さて、これまでの simple symmetric random walk の場合に戻って Andersen の結果をもとに、検定結果を考えてみよう。

7.2 m-sequence の Hamming weight について

Lindholm [27] , Jordan and Wood [20] では、 m-sequence の Hamming weight の分布を feedback polynoimal（ [27] , [20] では characteristic polynomial と呼んでいる）で決定している。以下に簡単に彼らの結果を述べる。

$a_n, n = 1, 2, \ldots, m$ を 0 , 1 からなる長さ m の数列とする。この数列の Hamming weight HW_m は次のように定義される。

$$HW_m \ = \ \sum_{i=1}^{m} \ a_i \ \ .$$

$f(x)$ を Galois 体 GF(2) の原始多項式とし、 a_n は $f(x)$ を feedback polynomial とする m-sequence とする。

$$A_m^{(j)}(f) \ = \ \{h : j \text{ 項式}, \ h(0) = 1, \ \deg h < m, \ f \text{ は } h \text{ を割り切る} \}, \ \ m \geq 3 \ ,$$

と置く。また、$B_m^{(j)}(f) \ = \ \#(A_m^{(j)}(f), m \geq 3$ とする。ここで、 $\#(A)$ は集合 A の元の個数である。

Jordan and Wood [20] では、 Hamming weight の分布について次のような結果を得ている：

定理

$$\Pr(HW_m = k) = 2^{-m} \left(\begin{array}{c} m \\ k \end{array} \right) \frac{2^p}{2^p - 1} (1 + \sum_{j=1}^{m} B_m^{(j)}(f)\ F_m^k(j)),$$

ここで、 $F_m^k(j)$ は 2 項係数で決まり、 $f(x)$ には無関係な値である。

一方で、 Lindholm [27] では、 Hamming weight の 3 次の moment が次の値で決定されることを示した。

$$m\, B_m^{(3)}(f)\ -\ \sum_{h \in A_m^{(3)}(f)} \deg h\ .$$

これらの結果より、例えば $m > 4p$ の場合、

$$f(x)\ , f^2(x)\ , f^4(x)\ \in\ A_m^{(3)}(f),$$

であるから、 3 次の moment は 0 でないことが分かる。即ち、 Hamming weight の分布は平均の回りに偏っていることが分かる。従って、例えば $m > 2p$ の時、

$$\Pr(HW_{2m} > m) \neq \frac{1}{2}\ ,$$

となることが分かり、 random walk の場合に翻訳すると

$$\Pr(S_{2m} > 0) \neq \frac{1}{2}\ ,$$

となる。図 1 などのシミュレーションでは、この確率は $> \frac{1}{2}$ であることが分かる。このことから、 sojourn time の経験分布の密度関数は Arc Sine Law からの偏りを示すことになる。

表 1: (A) M89T38 , 100 samples.

Path length L	Path number M	K_{30}^{+}		K_{30}^{-}	
		95∼99%	99% ∼	95∼99%	99% ∼
Hamming weight test					
160	50,000	0	0	0	100
200	50,000	0	0	0	100
Maximum test					
160	50,000	0	0	0	100
200	50,000	0	0	0	100
Sojourn time test					
160	50,000	0	0	1	99
200	50,000	0	0	0	100

表 2: (B) 加算生成法 , 100 samples.

Path length L	Path number M	K_{30}^{+}		K_{30}^{-}	
		95∼99%	99% ∼	95∼99%	99% ∼
Hamming weight test					
200	200,000	0	0	24	21
200	300,000	0	0	25	44
Maximum test					
200	200,000	1	0	13	5
200	300,000	0	0	12	12
Sojourn time test					
200	200,000	2	0	4	1
200	300,000	0	0	12	1
Last visit time test					
200	200,000	2	1	12	3
200	300,000	1	1	12	3

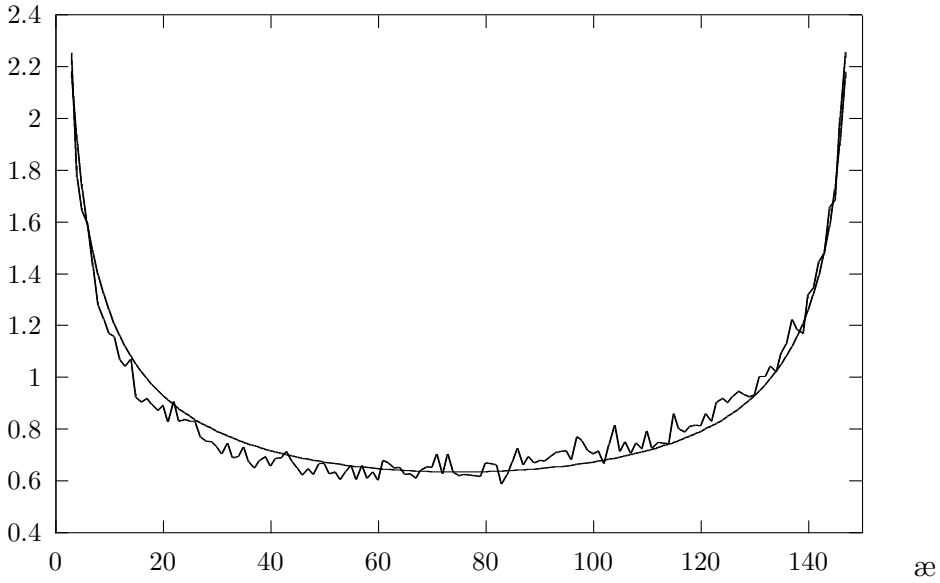

図 1: **M89T38**, Sojourn time test, 2L = 300

8 多数項の原始多項式による m-sequence について

前節で、原始 3 項式を feedback polynomial とする m-sequence は、見本関数の長さが原始 3 項式の次数よりある程度長い場合の random walk 検定において棄却されることを見てきた。伏見 [15]、[16] では、多数項を持つ原始多項式を feedback polynomial とする m-sequence の高速生成法を提案している。ここでは、簡単にそれを振り返り、彼の方法は random walk 検定の立場からは、原始 3 項式を feedback polynomial とする m-sequence と同様の欠点を持つことを述べる。

$g(x) = x^p + x^q + 1$, $p > q > 0$ を原始 3 項式とする。$G(x) = g(x^3)$ と置くと、$G(x)$ は既約ではないが、p 次の因数 $f(x)$ を持つことが知られている。$f(x)$ は必ずしも多数の項を持つとは限らないが、一般的には多くの項を持つことが期待できる。また、$f(x)$ が原始多項式になるかどうか一般的には分からないが、例えば、p が Mersenne 数の場合には、$f(x)$ は原始多項式となることが分かる。そこで、$f(x)$ を feedback polynomial とする m-sequence a_n を考える。$f(x)$ が $G(x)$ の因数であることから、

$$a_n \;=\; a_{n-3p} \;+\; a_{n-3q}\;,\quad n \geq 3p\quad ,$$

が成り立つことが分かる。この関係を用いると a_n を高速に生成することができる。ただし、$a_0, a_1, \ldots, a_{3p-1}$ は $f(x)$ を feedback polynomial に持つように決めなければならない。

以上が、伏見 [15] による、多数項原始多項式による m-sequence の高速生成法の概略である。

本報告では、例として $g(x) = x^{31} + x^3 + 1$ を用いた（ **M93T9** ）。この原始 3 項式の次数 31 はかなり小さいように思われるが、これを選んだのは第 2.5 で挙げた m-sequence **M89T38** の feedback polynomial と同じような次数を $G(x)$ が持つように考慮した結果である。

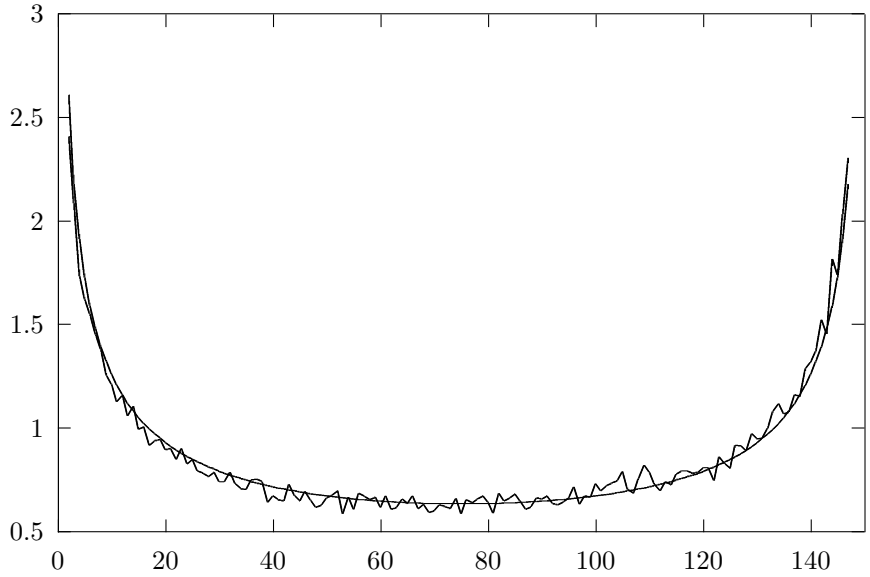

図 2: **M93T9**, Sojourn time test, 2L = 300

図 2 は図 1 と同じ条件で sojourn time 検定を行った結果である。この図 2 によると **M93T9** は sojourn time 検定に関して、 **M89T38** と類似の偏りを示していることが分かる。これは以下のようなことから了解されるであろう。

前節で述べたように、 m-sequence の Hamming weight の分布は feedback polynomial によって決定され、その 3 次の moment は $A_m^{(3)}(f)$ で決まる。**M93T9** が原始 3 項式に基づく **M89T38** と同様の偏りを示すことは、$A_m^{(3)}(G)$ は 3 項式の場合と類似の構成を持つのではないか、という推測出来るであろう。そこで、実際に数式処理ソフトウェアを用いて、 **M93T9** の feedback polynomial $f(x)$ について $A_m^{(3)}(f)$ と $A_m^{(3)}(G)$ を求めてみた。その結果、

$$A_m^{(3)}(f) \;=\; A_m^{(3)}(G)\ ,$$

が $p \le m \le 4\,p$ で成り立つことが確かめられた。このことは、伏見 [15] による、多数項を持つ原始多項式による、 m-sequence の高速生成法は、 sojourn time 検定、Hamming weight 検定に関して原始 3 項式による m-sequence と同様の欠点を持つことを意味する。このことは、単に上記の例 **M93T9** だけでなく、より高次の原始 3 項式の場合にも検証された。詳しくは Takashima and Ueda [48] を参照されたい。

この議論を更に押し進めて、以下のような予想を得る：

予想　あまり大きくない j , m に対して $A_m^{(j)}(f) = A_m^{(j)}(G)$.

ここで注意しなければならないのは、この予想は、すべての j , m に対して成り立つわけではない、ことである。このことは、原始多項式の倍数に関する一般論より分かる。しかしながら、数式処理ソフトウェアを用いることにより、ある程度の範囲では成り立っていることが確認されている（ Takashima [38] 参照)。さらに Munemasa [31] は次のような結果を得ている。

表 3: (C) Cellular Automaton , 100 samples.

Path length L	Path number M	K_{30}^{+}		K_{30}^{-}	
		95∼99%	99% ∼	95∼99%	99% ∼
Hamming weight test					
240	50,000	0	0	0	100
Maximum test					
240	50,000	0	0	0	100
Sojourn time test					
240	50,000	0	0	0	100
Last visit time test					
240	50,000	0	0	0	100

定理　f を 3 項式（必ずしも原始 3 項式であることを仮定しない）とする。このとき、$A_{2p+1}^{(3)}(f) = \{f, f^2\}$ 。

これらのことを考慮しながら、図 、図 などの結果を検討すると、sojourn time の極限分布は β 分布に従う、とする Andersen [1] , [2] の定理から予想される α の値と実際のシミュレーションから推定されるパラメーターとの間に微妙な食い違いが見られることが分かる。これに関しては、シミュレーションに用いた m-sequence の **独立性**に問題があるのではないか、と疑われる。

9 反転 m-sequence について

第 2.2 節で簡単に述べたように、Lidl and Niederreiter [26] と Golomb [17] とでは、特性多項式 characteristic polynomial の定義に違いが見られる。従来、この特性多項式の定義の違いはあまり重要視されてこなかったように見受けられるが、本当に重要な差異がないのかどうか、本節で考えてみたい。

以下、GF(2) 上で考え、まず、反転多項式 reciprocal polynomial の定義を述べる。

$$f(x) \;=\; \sum_{k=0}^{p} \; c_k \; x^k \;, \quad c_0 = c_p = 1 \;, \quad c_k \in \mathrm{GF}(2) \;,$$

に対して、 $f(x)$ の reciprocal polynomial $f^*(x)$ を

$$f^*(x) \;=\; \sum_{k=0}^{p} \; c_k \; x^{p-k} \;=\; x^p \; f(\frac{1}{x}) \;,$$

で定義する。 m-sequence a_n

$$a_n \;=\; \sum_{k=1}^{p} \; c_k \; a_{n-k} \;, \quad n \geq p \;,$$

を考えよう。ここで、注意すべきことは、 Golomb [17] では $f(x)$ を a_n の characteristic polynomial と呼び、Lidl and Niederreiter [26] では $f^*(x)$ を a_n の characteristic polynomial と呼んでいることである。そこで、 a_n の他に a_n^* を次のように定めよう。

$$a_n^* \;=\; \sum_{k=0}^{p-1} \; c_k \; a_{n-p+k}^* \;, \quad n \geq p \;,$$

即ち、 a_n^* は $f^*(x)$ を feedback polynomial とする m-sequence である。この m-sequence a_n^* と a_n との間には次のような関係があることが容易に分かる。

$$\begin{array}{ccccccccc} a_0 & a_1 & a_2 & a_3 & \cdots & a_{n-3} & a_{n-2} & a_{n-1} & a_n \\ \Updownarrow & \Updownarrow & \Updownarrow & \Updownarrow & \cdots & \Updownarrow & \Updownarrow & \Updownarrow & \Updownarrow \\ a_n^* & a_{n-1}^* & a_{n-2}^* & a_{n-3}^* & \cdots & a_3^* & a_2^* & a_1^* & a_0^* \end{array}$$

勿論、$p < n$ とし、初期値 $a_0^*, \ldots, a_{p-1}^*$ は $a_n, \ldots, a_{n-p+1}$ に取る。つまり、数列の長さを決めたとき a_n^* は a_n を逆の向きに見ていった数列である。この事実と次のような、代数学でよく知られているように、$f(x)$ と $f^*(x)$ はよく似た性質を持っていることから、今まで characteristic polynomial の定義における違いにあまり注意が払われなかったようである。例えば、 $f(x)$ が既約であることと、 $f^*(x)$ が既約である、ことは同値である。また、$f(x)$ が原始多項式であることと、 $f^*(x)$ が原始多項式である、ことは同値である、等々。

以下では、random walk 検定の観点から、 a_n と a_n^* の性質を比較してみよう。そのために以下のような、原始 5 項式を考えた。ここで、原始 3 項式でなく、原始 5 項式を考える理由は、すでに見てきたように、原始 3 項式による m-sequence は random walk 検定に関して良くない結果をもたらすので、 a_n と a_n^* の性質の比較には適していないと、判断されるからである。

表 4: **P5** and **R5**, 100 samples

Test type	Path length	N (×1000)	**P5** K_{30}^-		**R5** K_{30}^-	
			95%∼99%	99%∼	95%∼99%	99% ∼
HW	160	100	14	1	14	3
	240	100	9	3	11	5
MX	160	100	25	10	24	46
	240	100	14	6	3	96
SJ	160	100	0	1	35	28
	240	100	6	1	30	19
$\mathrm{FP}_{r=12}$	160	100	8	1	8	1
$r=20$	240	100	6	2	2	0

表 5: **P7** and **R7**, 100 samples

Test type	Path length	N (×1000)	**P7** K_{30}^-		**R7** K_{30}^-	
			95%∼99%	99%∼	95%∼99%	99%∼
HW	160	100	15	7	15	9
	240	100	12	2	11	4
MX	160	100	10	4	22	29
	240	100	7	1	19	34
SJ	160	100	2	2	10	2
	240	100	9	0	7	3
$\mathrm{FP}_{r=12}$	160	100	7	1	1	1
$r=20$	240	100	5	2	5	1

P5: $x^{61}+x^5+x^2+x+1$, **R5:** $x^{61}+x^{60}+x^{59}+x^{56}+1$,

P7: $x^{61}+x^7+x^4+x+1$, **R7:** $x^{61}+x^{60}+x^{57}+x^{54}+1$,

P8: $x^{61}+x^8+x^7+x^2+1$, **R8:** $x^{61}+x^{59}+x^{54}+x^{53}+1$,

P10: $x^{61}+x^{10}+x^6+x^5+1$, **R10:** $x^{61}+x^{56}+x^{55}+x^{51}+1$,

P47: $x^{61}+x^{47}+x^{20}+x^3+1$, **R47:** $x^{61}+x^{58}+x^{41}+x^{14}+1$,

P53: $x^{61}+x^{53}+x^{23}+x^3+1$, **R53:** $x^{61}+x^{58}+x^{38}+x^8+1$.

これらの結果より、 random walk 検定の観点からは、 m-sequence a_n と a_n^* とは必ずしも同様の性質を持つ、とは言えないことが分かる。より強く、幾つかの $f(x)$ の場合では、明

表 6: **P8** and **R8**, 100 samples

Test type	Path length	N (×1000)	**P8** K_{30}^{-}		**R8** K_{30}^{-}	
			95%∼99%	99%∼	95%∼99%	99%∼
HW	160	100	20	6	13	7
	240	100	10	2	7	6
MX	160	100	10	1	25	12
	240	100	5	1	10	3
SJ	160	100	0	1	11	1
	240	100	6	1	7	2
$\mathrm{FP}_{r\,=\,12}$	160	100	4	4	6	0
$_{r\,=\,20}$	240	100	4	0	3	0

表 7: **P10** and **R10**, 100 samples

Test type	Path length	N (×1000)	**P10** K_{30}^{-}		**R10** K_{30}^{-}	
			95%∼99%	99%∼	95%∼99%	99%∼
HW	160	100	12	6	16	6
	240	100	11	5	11	8
MX	160	100	20	10	23	28
	240	100	16	5	23	27
SJ	160	100	7	1	10	3
	240	100	7	1	8	5
$\mathrm{FP}_{r\,=\,12}$	160	100	4	3	3	4
$_{r\,=\,20}$	240	100	7	3	5	0

表 8: **P47** and **R47**, 100 samples

Test type	Path length	N (×1000)	**P47** K_{30}^{-}		**R47** K_{30}^{-}	
			95%∼99%	99%∼	95%∼99%	99%∼
HW	160	100	16	8	15	10
	240	100	12	1	11	5
MX	160	100	10	4	10	3
	240	100	4	4	6	5
SJ	160	100	6	1	3	3
	240	100	5	2	4	4
$\mathrm{FP}_{r\,=\,12}$	160	100	5	2	3	1
$_{r\,=\,20}$	240	100	3	2	3	0

表 9: **P53** and **R53**, 100 samples

Test type	Path length	N (×1000)	**P53** K_{30}^{-}		**R53** K_{30}^{-}	
			95%∼99%	99%∼	95%∼99%	99%∼
HW	160	100	15	6	16	13
	240	100	10	2	10	8
MX	160	100	10	3	12	2
	240	100	7	1	9	2
SJ	160	100	3	2	8	0
	240	100	6	1	5	2
$\mathrm{FP}_{r\,=\,12}$	160	100	2	0	5	1
$_{r\,=\,20}$	240	100	2	1	3	1

らかに異なる性質を示している。このことは、m-sequence の性質は Galois 体 GF(2^p) の性質でもあることを考えると、代数的に $f(x)$ とその reciprocal polynomial $f^*(x)$ との間には明らかな性質の違いがあることが分かる。

10 Hybrid 擬似乱数について

ここまで、主に m-sequence を中心に考えてきたが、m-sequence は random walk 検定に関してあまり芳しい成績を収めなかった。m-sequence に関しては、 k-次均等分布性など多くの研究がなされているが、それらは、一周期全体で平面（高次元空間も含む）上に一様に分布するか、という問題である。これに対して random walk 検定で目指したものは、非常に長い数列から計算される統計量をどの程度よく近似できるか、という問題である。この 2 つの問題はかなり異なる様相を持つことが分かる。それは単に m-sequence に関する random walk 検定の結果だけでなく、線形合同法の結果と比較することにより、より理解できるであろう。本報告では詳しく述べなかったが、線形合同法は高次元空間での均等分布性に関してはあまり芳しい成績は収めることができない。これは、その生成法（線形関数を利用する）による。ところが、 m-sequence と異なり、 random walk 検定では良い成績を示す。線形合同法の中でも、悪名高い RANDU でも sojourn time 検定や last visit time 検定にパスする。このように、擬似乱数は「真の乱数」ではないために、得手不得手があることが理解されるであろう。

そこで本節では、線形合同法と、それらとは異なる性格を持つように見える m-sequence , 加算生成法、などとを組み合わせて、上記の 2 つの問題、即ち、高次元空間での均等分布性と時系列のシミュレーション、に適した擬似乱数が生成できないか、という問題を考えることにする。

このようなアプローチは既に、 L'Ecuyer [25], Sobol [33], Wichiman and Hill [51] などによって試みられている。彼らは、例えば $[0, 1)$ 上に一様分布する、2 つの異なる線形合同法 x_n, y_n, を用いて、

$$z_n \;=\; x_n \;+\; y_n \qquad (\mathrm{mod}\ 1)\ ,$$

により、新しい擬似乱数を生成する方法を研究している。この方法では、係数の異なる線形合同法を利用するため、各々の係数を十分に吟味して選択しないと、望んだ性質が得られない。しかしながら、係数の良い組み合わせを見つけることは難しい問題である。

そこで、本報告では、線形合同法同志の組み合わせではなく、線形合同法と m-sequence , 加算生成法など、性質の異なる擬似乱数を組み合わせる（合同加算を用いる）ことを考える。以下に、簡単な例を挙げる。なお、これらの例は単に例として挙げるだけであり、これらを推薦するわけではないことを断っておく。

D：線形合同法と m-sequence の組み合せ

$$x_n = 16807 x_{n-1} \qquad (\mathrm{mod}\ 2^{31} - 1), \qquad n > 0,$$

$$y_n = y_{n-89}\ \mathbf{XOR}\ y_{n-38}, \quad n \geq 89,\ y_n : 31 - \mathrm{bit}\ \text{整数},$$

とし、

$$z_n = \frac{x_n}{2^{31} - 1} + \frac{y_n}{2^{31}} \qquad (\mathrm{mod}\ 1),$$

を考える。ここで、 x_n は 32 bit 整数を使う計算機で整数オーバーフローを利用せずに計算可能となるように生成することが可能である（ [25] 参照)。

E：線形合同法と m-sequence の組み合せ

$$x_n = 1664525x_{n-1} \pmod{2^{32}}, \quad n > 0,$$

$$y_n = y_{n-89} \ \mathbf{XOR} \ y_{n-38}, \quad n \geq 89, \quad y_n : 32 - \text{bit 整数},$$

に対して、

$$z'_n = x_n + y_n \pmod{2^{32}}, \quad z_n = z'_n/2^{32},$$

とおく。上記の方法との違いは、x_n を 32 bit 整数を用いて計算するため x_n と y_n の和を取る時に、通常の整数の和を取ればよく、 [**D**] におけるように、浮動小数点演算を行わなくてよいことにある。このため、生成速度が速い。

F：線形合同法と加算生成法の組み合せ

$$x_n = 1664525x_{n-1} \pmod{2^{32}}, \quad n > 0,$$

$$y_n = y_{n-55} \ + \ y_{n-24}, \qquad (\text{mod } 2^{32}) \ \ n \geq 55,$$

に対して、

$$z'_n = x_n + y_n \pmod{2^{32}}, \quad z_n = z'_n/2^{32}.$$

とおく。この計算でも [**E**] と同様に浮動小数点数の和を使わずに、整数和で計算できる。

上記のように、線形合同法と m-sequence 、加算生成法などとの組み合せにより、新しい擬似乱数を生成する方法を、本報告では hybrid 擬似乱数生成法、と呼ぶことにする。

ここで例示したように、２つの擬似乱数の合同和を考えることは、確率論では古くから研究されている。例えば、 [4], [6], [7] などを参照されたい。そこでは、以下のような結果を得ている。

X, Y を $[0, 1)$ 上に値を取る独立な確率変数とする。必ずしも一様分布とは限らないものとする。

$$Z_n \ = \ X_n \ + \ Y_n \qquad (\text{mod } 1) \ ,$$

とおくと、 Z_n は $[0, 1)$ 上、 X_n , Y_n より、一様分布により近い分布を持つ。

このように、これらの研究は一様分布性に関したものであるが、上記に例示した擬似乱数列はいずれも、一様分布性は十分に検討されており、問題ないものと考えられている。一方で、 random walk 検定のような、確率過程のシミュレーションに関する randomness の改善に関しての研究はほとんど無い、というのが実情である。そこで、統計的に random walk 検定をする必要性が生じることになる。

10.1 Hybrid 法に対する random walk 検定

本節では hybrid 擬似乱数に対する random walk 検定の結果について述べる。それらは表 10 、表 11 、表 12 に示される。

これらの検定結果は明らかに、基になっている、 m-sequence や加算生成法に比べて著しい改善を示している。これは、 random walk のシミュレーションに不適当な m-sequence や加算生成法に、線形合同法を組み合せた効果、と考えられる。

一方で、線形合同法や m-sequence は代数的に比較的簡単な構造を持つため多くの研究があり、周期の長さや均等分布性などの性質が分かっている。しかしながら、線形合同法の代数的構造と、 m-sequence の代数的構造は非常に異なるため、この両者を合同和で結びつけた、 hybrid 擬似乱数の代数的構造は全く分かっていない。例えば、周期についても全く分かっていない。しかしながら、 m-sequence の feedback polynomial を、前記の例のように Mersenne 数を次数に持つものを取れば、その m-sequence の周期は $2^p - 1$ であり、素数となるため、これを基にする hybrid 法の周期は $2^p - 1$ より短くはならない。そこで、前記の例などより、より次数の高い原始多項式を用いれば、周期の長さは十分なものが得られるであろう。

しかしながら、線形合同法と加算生成法の組み合せについては、全く理論的研究は無い。実際、加算生成法自体、理論的研究が非常に困難である。

ここで、この hybrid 法の利点について少し述べることにする。上記のように理論的には全く分からない hybrid 法であるが、この方法は、既存の m-sequence や加算生成法から、簡単に (ソフトウェア的に) 作り出すことが可能である。一方、 m-sequence や加算生成法を用いた擬似乱数生成ルーチンや、擬似乱数発生装置 (計算機に組み込みのものや、拡張ボードなどの形のもの) は世の中に広く流布している。これらの既存の計算機資産をプログラムの上で線形合同法と組み合せることにより、 random walk のシミュレーションなどにより適した、擬似乱数にする可能性を hybrid 擬似乱数生成法は与えている、と考えられる。

表 10: [D], 100 samples.

Path length L	Path number M	K_{30}^{+}		K_{30}^{-}	
		95∼99%	99% ∼	95∼99%	99% ∼
Hamming weight test					
160	50,000	4	2	1	0
160	50,000	2	2	4	1
200	50,000	0	0	3	2
200	50,000	1	0	3	0
Maximum test					
160	50,000	3	0	2	0
160	50,000	3	1	2	0
160	50,000	3	3	2	0
200	50,000	6	0	5	2
200	50,000	3	3	9	2
Sojourn time test					
160	50,000	3	1	3	1
200	50,000	3	0	3	0
Last visit time test					
160	50,000	5	2	1	0
200	50,000	6	2	8	5

表 11: [E], 100 samples.

Path length L	Path number M	K_{30}^{+}		K_{30}^{-}	
		95∼99%	99% ∼	95∼99%	99% ∼
Hamming weight test					
160	50,000	3	0	4	2
160	50,000	4	1	5	1
Maximum test					
160	50,000	4	0	3	3
160	50,000	4	0	3	1
Sojourn time test					
160	50,000	1	5	3	3
200	50,000	4	0	2	0
Last visit time test					
160	50,000	4	2	3	0

表 12: [F], 100 samples.

Path length L	Path number M	K_{30}^{+}		K_{30}^{-}	
		95∼99%	99% ∼	95∼99%	99% ∼
Hamming weight test					
200	100,000	4	1	5	1
200	200,000	5	2	2	3
200	300,000	6	0	4	1
300	50,000	7	3	2	0
Maximum test					
200	100,000	1	2	5	2
200	200,000	8	0	3	0
200	300,000	2	4	1	1
300	50,000	3	0	3	0
Sojourn time test					
200	100,000	3	0	3	0
200	200,000	4	1	3	2
200	300,000	4	0	9	1
300	50,000	5	0	2	3
Last visit time test					
200	100,000	2	1	5	1
200	200,000	2	0	5	1
200	300,000	2	0	6	1
300	50,000	5	2	6	1

参考文献

[1] E.S. Andersen : On the fluctuations of sums of random variables, *Math. Scand.*, **1**, 1953, 265 - 285.

[2] E.S. Andersen : On the fluctuations of sums of random variables II, *Math. Scand.*, **2**, 1954, 195 - 223.

[3] M.V. Antipov : The system of pseudorandom number generators on the personal computers, *preprint, AN USSR, Siberian Division, Comp. Centre; N 910*, Novosibirsk, 1990. (in Russian)

[4] M.V. Antipov : Sequences of Numbers for the Monte Carlo Methods, *Monte Carlo Methods and Appl.*, Vol. 2, N 3, 1996, 219 - 236.

[5] M.V. Antipov : The restriction principle and foundation of mathematics, *preprint, RAN, Inst. of Num. Math. and Math. Geoph.; N 1100*, Novosibirsk, 1997. (in Russian)

[6] M.V. Antipov : Multiple congruent convolutions and estimates in $L^{\infty}([0,1]^n)$, in *the Proceedings of Third St.Peterburg Workshop on Simulations*, Saint Peterburg University Press, 1998, 290 - 296.

[7] M.V. Antipov and G.A. Mihailov : On the improvement in random number generators by using a modulo 1 sum, *Russ. J. Numer. Anal. Math. Modelling*, **11**, N. 2, 1996, 93 - 111.

[8] P. Bratley, B.L. Fox and L.E. Schrage : *A Guide to Simulation*, 2nd ed., Springer, 1987.

[9] ベックマン：π の歴史 , 蒼樹書房 .

[10] N. Bouleau and D. Lépingle : *Numerical methods for stochastic processes*, (1994) Wiley.

[11] A. Engel : *Exploring mathematics with your computer*, Mtahematical Associations of America, 1993.

[12] W. Feller : *An Introduction to Probability Theory and Its Applications*, **Vol. 1**, 3rd. ed. Wiley, 1968.

[13] W. Feller : *An Introduction to Probability Theory and Its Applications*, **Vol. 2**, Wiley, 1971.

[14] G.S. Fishman : *Monte Carlo, Concepts, Algorithms, and Applications*, Springer, 1995.

[15] 伏見正則：乱数 , 東京大学出版会 , 1989.

[16] M.Fushimi : Random number generation with the recursion $X_t = X_{t-3p} \oplus X_{t-3q}$, *J. of Comp. and Appl. Math.*, **31**, (1990), 105 - 118.

[17] S.W. Golomb : *Shift register sequences*, rev. ed., Aegean Park Press, 1982.

[18] J.H. Halton : On the efficiency of certain quasi-random sequences of points in evalueating multi-dimensional integrals, *Numer. Math.*, **2**, 1960, 84 - 90.

[19] ホーエル：入門数理統計学 , （浅井、村上訳） , 培風館 .

[20] H.F. Jordan and D.C.M. Wood : On the Distribution of Sums of Succesive Bits of Shift-Register Sequences, *IEEE Trans. Comp.*, **C-22**, (1973) 400 - 408.

[21] D. Jungnickel : *Finite fields*, (1993) Wissenschaftsverlag.

[22] D.E. Knuth : *The Art of Computer Programming*, **Vol. 2**, *Semi-numerical Algorithms*, 3rd ed. Addison-Wesley, 1997.

[23] A.N. Kolmogorov : *The information theory and the theory of algorithms*, Nauka, 1987. (in Russian)

[24] Y.Kurita : On biases of weight distribution of L-tuples of M-sequences, *RIMS Kokyuroku, Kyoto Univ.*, **498**, 153 -171 (in Japanese).

[25] P. L'Ecuyer : Efficient and portable combined pseudorandom number generators, *Commun.ACM*, 1986.

[26] R. Lidl and H. Niederreiter : *Introduction to finite fields and their applications*, Cambridge Univ. Press, 1986.

[27] J.H. Lindholm : An analysis of the pseudo-randomness properties of subsequences of long m-sequences, *IEEE Trans. Inform. Theory*, **IT-14**, (1968) 569 - 576.

[28] G. Marsaglia : Random numbers fall mainly in the planes, *Proc. Natl. Acad. Sci. USA*, **61**, 1968, 25 - 28.

[29] M. Matsumoto and T. Nishimura : Mersenne twister: A 623-dimensionally equidistributed uniform pseudorandom number generator, *ACM Trans. on Modelling and Computer Simulations*, **8**(1), 1998, 3 - 30.

[30] G.A. Mikhailov and M.V. Antipov : Estimations of non-uniformity for distributions of the congruent sums of random values, *Doklady RAN*, **347**, N. 1, 1996, 23 - 26. (in Russian)

[31] A. Munemasa : Orthogonal arrays, primitive trinomials, and shift-register sequences, *Finite Fields and Their Applications*, **4**, 1998, 252 - 260.

H. Niederreiter : A statistical analysis of generalized feedback shift register pseudo-random number generators. *SIAM J. Sci. Statist. Comp.*, **8**, 1987, 1035-1051.

[32] W.H. Press, S.A. Teukolsky, W.T. Vetterling, and B.P. Flannery : *Numerical Recipies in C*, Cambridge Univ. Press, 1988.

[33] I.M. Sobol : *Numerical Monte Carlo Methods*, Nauka, 1973. (in Russian)

[34] F. Spitzer : A combinatorial lemma and its application to probability theory, *Trans. Am. Math. Soc.*, **82**, 1956, 323 - 339.

[35] F. Spitzer : *Principles of Random Walk*, Springer, 1964.

[36] K. Takashima : Sojourn time test for maximum-length linearly recurring sequences with characteristic primitive trinomials, *Journal of Japanese Society of Computational Statistics*, **7**, 1994, 77 - 87.

[37] K. Takashima : Sojourn time test for m-sequences with characteristic pentanomials, *Journal of Japanese Society of Computational Statistics*, **8**, 1995, 37 - 46.

[38] K. Takashima : On the number of multiples of certain primitive polynomials over GF(2), *Monte Carlo Methods and Applications*, **2**, 1996, 15 - 24.

[39] K. Takashima : On Hamming weight test and sojourn time test of m-sequences, *Monte Carlo Methods and Applications*, **2**, 1996, 331 - 340.

[40] K. Takashima : Last visit time tests for pseudorandom numbers, *Journal of Japanese Society of Computational Statistics*, **9**, 1996, 1 - 14.

[41] K. Takashima : M系列による滞在時間とハミング重みのシミュレーションからの予想, **統計数理**, **44**, 1996, 99 - 104. (原著論文)

[42] K. Takashima : Random walk tests of additive number generators, "*Proceedings of the Workshop on Turbulent Diffusion and Related Problems in Stochastic Numerics*" (eds. S. Ogawa and K. Sabelfeld), Inst. Stat. Math., 1997, 55 - 65.

[43] K. Takashima : Random walk tests of reciprocal m-sequences, *Monte Carlo Methods and Simulations*, **3**, 1997,

[44] K. Takashima : Random walk tests of pseudorandom number generations by cellular automata, in *the Proceedings of Third St.Peterburg Workshop on Simulations*, Saint Peterburg University Press, 1998, 302 - 305.

[45] K. Takashima : Hybrid pseudo-random number generation, *Monte Carlo Methods and Applications*, **6**, no.1, 2000, 49 - 59.

[46] K. Takashima : Random walk tests and Pseudorandom number generators, in Stochastic Analysis and Applications, Ed. Y. J. Cho, Nova Science Publ., Huntington, New York, 2000, 59 – 65.

[47] K. Takashima : A note on hybrid pseudorandom number generations, in Proceedings of Sixth International Conference of Computer Data Analysis and Modeling, Eds. Aivazian, Kharin, and Rieder, Belarusian Stae Univ. vol.3, 2001, 98 – 103.

[48] K. Takashima and S. Ueda : Sojourn time test of m-sequences by Fushimi's fast generation methods, in *Probability Theory and Mathematical Statistics, World Scientific Publishing*, S. Watanabe, M. Fukushima, Yu.V. Prohorov, and A.N. Shiryaev (eds.) 1996, 471 - 477.

[49] 石川利久、高嶋恵三：1 次元ランダムウォークの汎関数の同時分布について , 大阪教育大学紀要 III , 第 45 巻 , 1996 , 9 - 18.

[50] K.D. Tocher : The application of automatic computers to sampling experiments, *J. Royal Statist. Soc.*, ser. B, **16**, N 1, 1954, 39 - 61.

[51] B.A. Wichmann and I.D. Hill : An efficient and portable pseudo-random number generator, *Appl.Stat.*, **31, N2**, 199 - 190, 1982.

[52] S. Wolfram : Random Sequence Generation by Cellular Automata, *Adv. Appl. Math.*, **7**, 1986, 123 - 169.