The Japan Statistical Society

日本統計学会誌
第35巻，第2号，2006年3月
165頁～180頁

擬似乱数ユーザーの方へ—Mersenne Twister法開発者より

松　本　　眞†

To the users of pseudorandom number generators —from a developer of Mersenne Twister

Makoto Matsumoto†

漸化式を用いて乱数のように「見える」数列を発生し，乱数として利用することを擬似乱数発生という．真乱数発生に比べて高速・低コストであるほか，初期シードと漸化式を記録すれば誰でも再現可能であるため追試を行いやすいなどのメリットがある．

しかし，現在科学技術計算ライブラリなどで通常利用されている擬似乱数の多くに無視できない欠陥があり，擬似乱数ユーザを混乱させている．一方で，これらの欠陥を考慮して筆者と西村拓士が97年に開発し普及が進んでいるMersenne Twister法では，周期が$2^{19937}-1$で623次元空間に均等に分布しているなどの数学的な保障があり，大規模な統計的検定にも合格している上，実際に数兆個単位でシミュレーションに利用されている．

本稿は，ユーザに届きにくい擬似乱数開発者側の情報，すなわち漸化式・既知の欠陥・評価方法などを解説することにより両者の距離を縮め，研究成果を共有し刺激しあうことを目的とする．

Pseudorandom number generation (PRNG) is to generate a sequence of numbers which "mimics" a truly random number sequence, and to use them as random numbers. PRNG is faster and cheaper than physical random number generations, and is easily reproduced by recording the initial seed, which is necessary to check the simulation result later or by another computer.

However, commonly used generators in scientific computing libraries often have some flaws, which confuse the users of PRNG. On the other hand, Mersenne Twister PRNG developed by the author and Takuji Nishimura in '97 and now being widely spread, has mathematical assurance such as the long period $2^{19937}-1$ and 623-dimensionally equidistribution. It also passes stringent statistical tests, and actually used in simulations where trillions of random numbers are consumed.

This manuscript explains the designers' information on PRNG to the users, such as the recursion formulas, known defects, methods of evaluation, which have not reached to the users enough. This intends to tie the users and the designers, so that they can take the benefit of the recent research results and stimulate each other.

Key Words and Phrases: Random number generation, Pseudorandom number generation, Mersenne Twister, MonteCalro methods, Simulation, Defects

はじめに

乱数発生法の研究においては，「統一見解」は得られていない，と筆者は認識している．そのため，それぞれの研究者がドグマをもち，互いにいくらか矛盾する内容を真実として主張し

† 広島大学大学院理学研究科数学専攻（〒739-8526 東広島市鏡山1-3-1　広島大学大学院理学研究科）

NII-Electronic Library Service

The Japan Statistical Society

ているきらいがある．当稿も，一つの視点からみた主張にすぎない．個人的には客観的真実に近づける努力を払って執筆したつもりだが，上記のことを差し引いて読んでいただければ幸いである．

本稿は，西村拓士氏との共同研究である Mersenne Twister 法[20]（MT 法）を紹介することを目的の一つにおいている．用語に厳密な定義を与えると却ってわかりにくくなる恐れがあるので，感覚的に理解できる用語を用いてまず MT 法の紹介をし，それから少し広い視点から擬似乱数全般の解説を試みる．

§1.2 で触れるように，擬似乱数研究の最大の問題は「利用者と研究開発者の間に距離がある」ことと思える．本稿が，この距離を少しでも近づけられたら幸いである．質問メールは常に大歓迎で，できる限り返事をする努力をしているが全てには答え切れていないことをあらかじめ陳謝させていただく．

1. 問題のある生成法と Mersenne Twister 法

MT 法は筆者と西村拓士氏が 98 年に発表した擬似乱数発生法である．状態空間が 19937 ビット，周期が $2^{19937}-1$ であり，通常のパソコンでも毎秒 1000 万個以上の擬似乱数が生成できる高性能高速擬似乱数発生法である．現在，モンテカルロ法用としてはもっとも信頼性の高い擬似乱数の一つとして世界的に普及している．多くの数値計算ライブラリーに組み込まれているほか，MT 法を利用した多種のフリーウェアがインターネット経由でダウンロードできる．

さて最初に強く注意しておきたいのは，現在広く用いられている擬似乱数の中にも質の悪いものが多くあることである．新しく提唱されたものでも，高品質であるとは限らない．一例を挙げよう．多くの研究者が Knuth の名著[7]（97 年第 3 版）を「擬似乱数発生のバイブル」と呼び，「この本に推奨されている擬似乱数を使えば間違いがない」といった紹介をしている．しかし，この本が扱っている生成法の大部分は 70 年代に開発されたもので，それぞれに多少の問題がある．第 3 版で新たに追加された ran array という生成法も，古い擬似乱数「ラグつきフィボナッチ法」の「一部を使う」という改良に過ぎず，(luxury level を低く設定した場合には）偏りが見られる[23]．筆者は Knuth 氏が 96 年に来日した際，第 3 版のドラフトを見て「バイブルと呼ばれるほど影響力のある本が，よい擬似乱数を紹介していないのは問題ではないか」と若気の至りで噛み付いたことがある．氏の返答は「この本をバイブルと呼びたい人は，『旧約聖書』と呼ぶべきであろう．他の巻にはもっと緊急に書き直すべきところがあり，乱数には手がほとんど回せない」とユーモアのある大人の答であった．このころ筆者は MT の論文の草稿を書き上げたところで，そこでは MT の名前は primitive twisted GFSR という長いものであった．氏にこの草稿を渡したところ，氏から「とても面白いが，まだテストが不完全だ．広く普及して徹底的なテストに合格すれば，15 年後に予定している第 4 版で紹介するに値しよう．…ところで，名前が長すぎる．もっと印象に残る名前を考えては」というコメントをいただき，それに応えて「MT」という名を考案した．氏からは他にも有益なコメントをいただいた．ここに謝意を表させていただく．(筆者も[7]はバイブルと呼ぶに値すると考えている.)

さて，以下，問題があるが広く使われている乱数発生法を二例挙げる．

1.1 線形合同法

もっとも標準的な生成法といえる「線形合同法」(Lehmer, '60 ごろ）は，整定数 a, c, N を固定し N を法とした（N で割った余りを取ること，mod N と記す）線形漸化式

$$x_{j+1} := ax_j + c \bmod N \quad (j=0, 1, 2, \ldots) \tag{1}$$

により 0 から $N-1$ までの整数列を生成し，擬似乱整数列として用いるというものである．

NII-Electronic Library Service

The Japan Statistical Society

(初期値として，x_0 をユーザーが与える必要がある.) この数列を実数列と考えて N で割ることにより，[0, 1) 区間に一様分布する擬似乱数として用いることも多い．周期はたかだか N である.

高速生成のためには積と剰余の計算がネックとなる．そのため，N は計算機の 1 ワード長整数の最大値（たいてい 2^{32}）以下のものが多い．しかし，現在のパソコンでは数分で 2^{32} 個の乱数を使ってしまうため，周期が短すぎると言える．また，生成された擬似乱数を高次元空間にプロットすると以下のような格子構造が観察される.

例として，80 年代後半まで ANSI-C 言語の標準推奨擬似乱数であった rand は $a=1103515245$, $c=12345$, $N=2^{31}$ とした線形合同法である．この擬似乱数列の 3 次元空間での分布を見るため,

1．3 つ擬似乱数を発生し，それらを xyz 座標とする点を単位立方体にプロット

2．1. を 2^{31} 回繰り返す

3．単位立方体の原点付近，一辺 0.015 の立方体内にプロットされた点を描画

すると，図 1 のような格子構造が明らかになる．従って，この方法を使った場合，格子構造が見えてくるような多量の（数千万個程度の）擬似乱数の使用は明らかに危険である．また，別種の欠陥が N が 2 のべき乗である場合に起きる．剰余計算が高速になるのでそのような N を選ぶことが多いが，出力が交互に偶数と奇数になる（一般に，2 進数で書いて下 s 桁は周期 2^s で循環する).

現在でも，多くの科学計算ライブラリやシミュレーションの教科書で線形合同法が推奨されている．法 N を 2^{48} に増やした drand48 などの線形合同法もあるが，本質的に同じ問題を持つ上，生成速度がずっと遅い．(ちなみに，最新の BSD-C 言語の rand は 90 年代にラグつきフィボナッチ法 random に置き換わっている．この生成法は高速だが，深刻な欠陥が報告されている [23][31][25].)

MT 法では一周期に渡って見た場合，出力は 32 ビット精度で 623 次元空間に均等に分布している (§4.2.2)．数列の一部分だけを使った場合の分布の保障はないが，MT 法は 10 億個以上の乱数サンプルを用いた多様な統計的検定に全て合格している．現在まで筆者にメールなど

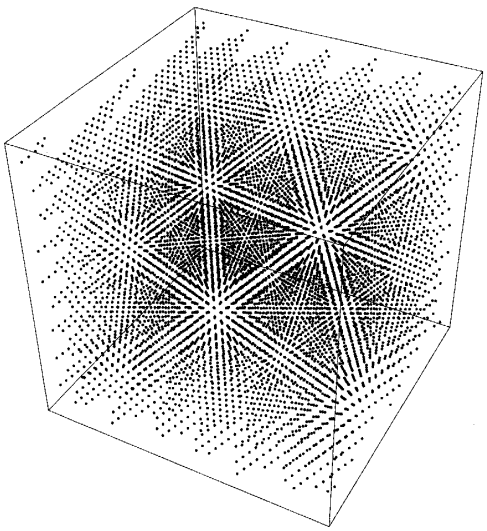

図 1　80 年代まで ANSI 標準 C 言語の擬似乱数 rand により生成された空間内の「ランダム」な点列．ランダムとは言い難い結晶構造が見られる.

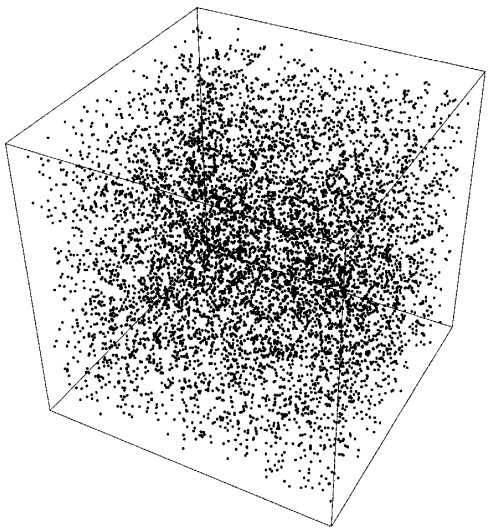

図 2　同じ方法でメルセンヌツイスター法により生成されたランダムな点列.

NII-Electronic Library Service

The Japan Statistical Society

で寄せられた10件ほどの「MT法の統計的偏りの発見」は，全て検定方法の誤りであることが判明している．筆者がさまざまな生成法の検定を行った経験によれば，MT法に限らず一般に次のことが観察される．「一周期でM次元均等分布している擬似乱数は，連続するM個以下の乱数を組にして扱う統計的検定には合格する．」すなわち，M次元までの独立性しか要求しない検定には合格する．従って，623個程度までの多変量のシミュレーションに用いた場合には問題がない（そして，おそらくはもっと高い次元でも問題は生じない）と思われる．（作為的に設計された検定ではこの限りではないが．）

1.2 GFSR法

もう一種，広く用いられかつ問題の大きい生成法に3項GFSR（Generalized Feedbacked Shiftregister）法[9]がある．これは，計算機の1ワードをwビットとしたとき，1ワードを{0, 1}成分のw次元横ベクトルとみなし，ベクトル列$x_0, x_1, x_2, \ldots$を漸化式

$$x_{j+n} := x_{j+m} + x_j \quad (j=0, 1, \ldots) \tag{2}$$

で生成し，ランダムワード列として用いる方法である．ここで，和は成分ごとの2を法とする和である．（計算機用語で言うビットごとの排他的論理和であり，きわめて高速に計算できる．数学用語で言えば，$\mathbb{F}_2$係数のベクトルの和である．ここで，2を法とした加減乗算の与えられた集合{0, 1}を2元体といい$\mathbb{F}_2$であらわす．）$x_0, \ldots, x_{n-1}$を初期値としてユーザーが与える必要がある．定数$n>m>0$をt^n+t^m+1が原始多項式になるように選ぶと，このベクトル列の周期は2^n-1となる．2^n-1が素数になると原始性の判定が簡単になるため，n=89, 127, 521, 607といった値のものが良く用いられている．

この方法は周期が長く，一個の生成に数回のCPU命令しか要せずに高速なので広く用いられて来た．が，ベクトルの成分としてあらわれる0-1の個数のバランスに顕著な偏りがある．どのビットでも同様だが，例えば生成されるワード列の最上位ビットに着目し，それをN個ずつに区切って，N個中に現れる1の個数を数え，二項分布$B(N, 1/2)$からの偏りを見る（これを重み分布検定という）と，$N>n$の時には偏りが大きいことが知られている（$N\leq n$では経験的に偏りは観察されない）．従って，ランダムウォークや相転移など，状態変数に過去の履歴が長く残るマルコフ過程型のシミュレーションでは大きな誤差を生み出すことがある．

この事実は68年Lindholm[10]，75年Fredricsson[5]らにより数学的に解析され警告されていたが，なぜか無視されてきた．1992年にFerrenbergら[4]が（その理由は解析せずに）「Ising模型のシミュレーションでこの種の擬似乱数を使うと相転移温度が狂う」ことを報告し脚光を浴びた．が，これは擬似乱数の専門家にとっては既知の問題であった．このように，「問題点がはっきりしている擬似乱数が繰り返し提案され，いまだに広く用いられている」というところに，現状の問題の根深さがある．

信頼性の高い高速擬似乱数発生法が存在しても，乱数研究の大家の「ドグマ」により無視され普及しない．そして，（専門家にとっては既知の）欠陥を持つ乱数が使われつづけ，ユーザーによりその欠陥が（再）発見されて脚光を浴び，場当たり的な改良により欠陥が縮小される．やがて，計算機の性能の向上により欠陥が再び明らかになる．といった繰り返しが起きる．その結果，欠陥を持つものばかりが広く使われ，物理や統計などの分野でも（その欠陥も含めて）喧伝されてきた，とさえ言える．

このため，擬似乱数に対する信用は低く，今日でも「数千万個以上の乱数を使った実験をする場合には，物理乱数を用いねばならない」というような意見を目にする．しかし，前述のように，MT法などの現代的擬似乱数は，十億個程度のサンプルに対する統計的検定をパスしている．そして，核物理シミュレーションなどでは兆単位で使用され成果を挙げている．

NII-Electronic Library Service

The Japan Statistical Society

筆者のグループでは[5]を発展させ，符号理論の MacWillams 恒等式を用いて，$\mathbb{F}_2$ 線形擬似乱数のビットの二項分布からの偏りを精密に計算し，検定に棄却されるサンプルサイズを求めるアルゴリズムを開発実装した（重み齟齬検定[22]）．これによれば 3 項 GFSR 法（2）において，$n=89$ では 10 万個程度を超えたサンプルをとると重み分布検定に棄却されることがわかる．$n=521$ でも，1000 万個程度で棄却される．一方，MT 法では（通常の mt19937 では量的限界によりこの計算が行えなかったが），周期を $2^{521}-1$ とした小規模モデルの mt521 でも，重み分布検定による棄却のためには 10^{156} 程度のサンプルサイズが必要であることが示されている．

重み分布検定は，筆者の経験では GFSR や MT などの $\mathbb{F}_2$ 線形擬似乱数の偏りを検出するにはもっともセンシティブな検定である．連の検定（run-test），区間を等分しての χ^2 検定＋χ^2 値の KS 検定，などの検定の検出力はこの種の生成法に対してはずっと低い．

以下，少し一般的な枠組みから擬似乱数を概観しながら，MT 法開発の背景を述べる．

2. 一般論：擬似乱数

2.1 乱数

統計学において，乱数は古くから利用されている．たとえば，巨大な母集団からランダムにサンプリングを行うには乱数は不可欠である．Pearson の弟子 L.H.C Tippett が 1927 年に世界で最初の乱数表を作った大きな動機は，ランダムサンプリングを効率よく行うことであったらしい．Tippett は多くの紙片に数を書き，箱から無作為に抽出するという実験を繰り返したところ，得られた乱数列には大きな偏りがあることを発見した．偏りがなくなるまで紙片を混ぜるのは大変であり，そのような努力を何度も払う手間を考えれば，表を出版することに意義があると判断したそうである．実際実験してみると，乱数を人力で生成することは想像以上に労力を要する．

近年，コンピュータの発達に伴い，乱数の用途は広がり，かつ大規模化されてきた．統計学と密接なのは，統計的現象のモンテカルロシミュレーションであろう．多数の確率変数を含む複雑な関数の平均，期待値，分散，分布の概形を求めるのに，数値積分を行うことはしばしば不可能に近い．そこで，それぞれの確率変数の実現値を乱数を用いて発生させ，関数に代入することを多数回繰り返し，得られた関数の値の平均，期待値，分散，分布を真の値の近似値とすることを，（数値積分に関する）モンテカルロ法という．

最近は，マルコフ過程モンテカルロ法の利用が急速に広まっている．これは，必ずしも明示的に与えられていない確率分布に従うサンプルを得たい場合に，その分布を定常分布として持つマルコフ過程を設計し，乱数によりマルコフ連鎖を適当なステップ回シミュレーションして最後の状態をサンプルとする方法である．

これらの応用の際，いかにして

- 乱数を発生するか
- その乱数が使用に耐える品質か否かを判定するか

が問題となる．そして，結論から言うと，「腑に落ちる解答は，おそらくは永久に，出そうにない」というのが，乱数研究の端緒から現在まで続いている状況であると言える．その理由は，一言で言えば「乱数の十分実用的と言える定義が存在しない」ことに起因する．

2.2 素朴な定義

乱数に対しては，次の定義が一般的である．

定義 2.1. $X_1, X_2, \ldots$ を，ある同一の分布 $F(x)$ に従う独立な確率変数とする．乱数とは，こ

NII-Electronic Library Service

The Japan Statistical Society

れら確率変数の実現値の列 $x_1, x_2, \ldots$ のことを言う.

本稿では, §5.2 を除き [0, 1) 区間上での一様分布, もしくは有限集合上での一様分布のみを扱う.

たとえば, サイコロを振って出る目を考える. 理想的にはどの目も確率 1/6 であらわれ, また振るごとに出る目は独立であるから, 出た目の列は乱数列である.

しかし, 一つの具体的な「実現値」という概念は, 厳密な定義とは言えない. たとえば, 円周率 $\pi = 3.1415926535\cdots$ の小数部分は, 上の意味で乱数と言えるのか? では, 乱数といえない数列は何なのか? どんな有限数列も, ある確率で実現されるではないか.

擬似乱数の定義となると, さらにあいまいになってくる.

定義 2.2. 擬似乱数発生とは, あたかも乱数のように見える数列を, 決定性のアルゴリズムで生成することを言う.

2.3 物理乱数

擬似乱数はちょっと後回しにする. 乱数発生法としてまず思いつくのは, 物理的に発生される雑音を源泉として乱数を発生させる物理乱数発生器である. しかし, 計算機に一つ余計な回路をつけるコストは低くない. また, このような独立した回路は, 中途半端に故障したときに検知が難しい. 他に, 温度など外部環境への依存性も気になる. 市販されている物理乱数は大抵, 結局擬似乱数と組み合わせる (例えば排他的論理和をとる) ことで環境依存性を排除しているようである.

もっと本質的な問題は,「再現性」がないことである. 核物理のシミュレーションでは数兆個の乱数を使うことも珍しくない. そして, たとえば核実験でセンサーを置く位置を最適化したいときに,「同じ乱数列で」繰り返しシミュレーションを行う必要が生じる. 物理乱数では, 同じ数列を発生するにはそれを全て記録しておくしか方法はない. 他に再現性は, 違う研究グループによる追試の際にも必要となる.

逆に, くじの番号生成や, 暗号乱数の初期値生成など, 再現性が望ましくない応用には物理乱数が適している.

2.4 擬似乱数

より低コストかつ再生容易な生成法として,「漸化式により数列を発生し, それを乱数として用いる」方法が挙げられる. 漸化式と初期値を記録しておけば, 誰でも何回でも同じ数列を発生できる. これを擬似乱数発生という. 分布や周期などの性質が数学的に解析できる点も物理乱数と異なる. 最大の問題は「乱数として使用可能な数列とは何か」という定義の不在である.

2.5 乱数・擬似乱数の定義

実は, 乱数・擬似乱数の定義はいくつも存在する. Kolomogorov 流の定義で言えば,「有限数列が乱数であるとは, その数列よりも短い言葉ではその数列を定義できないこと」である. より厳密には,「普遍的なデジタル計算機であるチューリングマシンにより, その数列の長さより短いプログラムによってはその数列が生成できないこと」, として定義される. このような数列は多数存在することが証明できるが, 計算機による効率的発生は定義から不可能であり,「乱数発生」の立場からは絵に書いた餅である.

そこで, 暗号乱数の分野では次の定義を用いる.「ある数列が計算量的に安全な擬似乱数であるとは, それを生成するのには多項式時間しかかからないが, その数列から次に現れる数を

NII-Electronic Library Service

The Japan Statistical Society

推測することは多項式時間ではできないこと.」生成する側には，初期値と漸化式が与えられ，効率的に数列を生成できる．擬似乱数を解読する側には，漸化式と今までに生成された数列のみが与えられる．そこから初期値を求めたり，これから生成される数列に対する情報を得たりすることが現実的な計算時間では不可能，というのが計算量的擬似乱数の定義である．統計の視点から見ても「計算機を用いてどのような統計的検定を行っても，棄却するのに非現実的に長い時間がかかる」ことを意味しており，申しぶんない．

問題は，そのような数列が存在するかどうかである．この種の数列として最初に提案され実用に供されているものに，Blum-Blum-Shub（BBS）と言われる生成法がある[1]．もしこの擬似乱数を多項式時間で見破る方法があれば，「(ある種の）合成数の素因数分解」という，いままで多項式時間では不可能と信じられてきた計算が多項式時間で可能となることが証明されている．いままで数学者が挑戦してできなかったのだからそれはありそうにない，というのが安全性の根拠である．

現存の「計算量的に安全な擬似乱数」はすべて，「この擬似乱数が見破られれば，多項式時間では計算不可能と信じられているある既存の問題が多項式時間で解けてしまう」ということに根拠を持っている．万一その既存の問題の高速な解法が見つかればその安全性を失ってしまう．(逆手にとれば，この種の擬似乱数の統計的偏りを発見すれば，画期的な素因数分解法などが得られることになる．実は，これは筆者が密かに狙ってきた「青春の夢」である.）さらに，この種の擬似乱数の周期や分布の保障は一般には難しい．そのため，計算量的に安全ではないが周期の長い擬似乱数と組み合わせて用いることも多い．

暗号用の乱数を発生するには，上のようなものの他に，フィルター方式と呼ばれる「計算量的に安全ではないが周期や分布の保障のある擬似乱数を，逆変換が困難と思える一方向性の関数によって変換する方式」がある．こちらの方は，その安全性の理論的保証がない．しかし，一方向性関数の選択の自由度が広く高速なため，生成速度が重視される用途では，フィルター方式の方が自然な選択であるとも思える．例えば，'05 年に ECRYPT が行ったストリーム暗号乱数の公募[3]には，BBS のような計算量的に安全な擬似乱数の応募はなかった．(筆者-萩田-西村-斉藤は「MT＋フィルター」型の暗号乱数生成法を公募した．論文はプレプリントアーカイブ[24]で見られる.）

なお，モンテカルロ法用擬似乱数においては，計算量的安全性は必ずしも要求されない．むしろ，周期や分布にある程度の保障があることが望まれる．

3. 擬似乱数生成法

このような定義不在の状況下で，さまざまな漸化式による数列が擬似乱数として用いられている．

3.1 オートマトン＝漸化式

定義 3.1. S を（メモリのとりうる状態すべてからなる）有限集合とし，$f: S \to S$ を次状態を決める関数とする．$s_0 \in S$ を初期状態とし，単位時刻ごとに状態を $s_0, s_1 := f(s_0)$, $s_2 := f(s_1), \ldots$ と遷移させていく．出力の集合を O とし，$o: S \to O$ を出力関数とする．状態 s_i での出力を $o(s_i)$ で与えることにより，出力列

$$o(s_0), o(s_1), o(s_2), \ldots$$

を得る．このようなシステムを（無入力有限状態）オートマトンという．

メモリが有限で，入力の行われないデジタルコンピュータはこのようなシステムである．出

NII-Electronic Library Service

The Japan Statistical Society

力列はいずれは周期的になり，その周期はSの元の数$\#(S)$を超えない．

言い換えると，周期的にならない数列を生成するアルゴリズムにおいては，生成にしたがって占有しているメモリが増えていく．時々「周期が無限の擬似乱数発生アルゴリズム」を提唱する論文を目にするが，実際に計算機に実装する際には使用メモリに上限が設けられる．どうせ上限が設けられるのなら，それを確定した上でその範囲で周期を最大化したアルゴリズムのほうが乱数性が良いことが多い．

「実装した場合理論と異なる」という問題は，「カオス理論や確率論を用いた」という売り込みの擬似乱数にしばしば起きる．連続無限の状態空間をもつ力学系を元に，「カオス的力学系であるから良い擬似乱数だ」という根拠で提案され流布している擬似乱数にranlux[11]という生成法があるが，初期値に対する（カオス理論的解析からすればありえないはずの）依存性が観測される[25]．その原因の一つは，丸め誤差にある．計算機内部では連続無限は離散化されて実現されており，そこで生じる丸め誤差がカオス理論的解析とは異なった現象を起こす．実際，先に見た線形合同法も，（$\{0, 1, \ldots, M\}$を正規化して）連続な$[0, 1)$区間上の力学系として見ればカオス的であるが，離散集合上の力学系としては先に見たような格子構造を持つ．

そもそも，擬似乱数発生法に用いられてきた漸化式の多くは連続関数から程遠いものであり，ある意味「カオス」以上の非連続力学系となっている．カオス的であるだけでは，擬似乱数の品質をほとんど保証しない．

同様に，「初期値を$[0, 1)$区間からランダムに選べば確率1で良い擬似乱数を発生する」といった擬似乱数も，トートロジーのきらいがある．というのは，もし$[0, 1)$区間から一様に一個乱数を取ってくることができるのであれば，その小数展開を計算するだけでもいくらでも真乱数を得ることができるからである．そして離散化に伴う誤差のため，これは実現できない．

デジタル計算機における擬似乱数発生は，確率という解析的な現象を扱うものであるにも関わらず，むしろ離散的集合を離散のまま取り扱い周期や分布を最適化する「離散最適化」が有効な分野と思える．

3.2 線形合同法再び

先に見た線形合同法は，オートマトンの定義3.1で，Nを十分大きな数とし，Nを法とする剰余の集合$\mathbb{Z}/N$を状態集合Sとし，次状態関数を

$$f(s)=as+c \mod N$$

としたものである．

現在モンテカルロ法における主流は，線形合同法を含んだより広いクラスの「線形生成法」，すなわち何らかの意味で線形な次状態関数を用いた生成法である．次状態関数として，二次関数や分数関数などの非線形写像を用いたものも提唱され使われてはいるが，周期の計算が難しかったり，分布に関する保障がなかったり，生成速度が遅かったりする．ちなみに，先にあげたBBSは二次関数を用いている．

そもそも，Von Neumannが40年代に提唱した最初の擬似乱数発生法の次状態関数は，「自乗して10進展開し中位の桁を数桁取り出す」という非線形なものであった．この方法を使うと，初期値によっては周期が非常に短くなることがある［7, §3.1］．線形合同法はこのような経験を重ねて選び出された漸化式であり，一時代を築いた．線形合同法は（良いパラメータを選べば）優れた生成法であったが，大規模利用が進むにつれ§1.1で見たような問題が深刻化した．

3.3 有限体の利用

§1.2で触れたとおり$\mathbb{F}_2=\{0, 1\}$を2元体，すなわちmod 2で加減乗算を行う集合とする．

NII-Electronic Library Service

The Japan Statistical Society

オートマトンの定義 3.1 において，状態集合 S を d 次元ベクトルの空間 $\mathbb{F}_2^d$ とし，f として $\mathbb{F}_2$-線形写像を用いるのが $\mathbb{F}_2$ 線形生成法である．

MT 法はこのクラスに属する生成法であり，漸化式

$$x_{j+n} := x_{j+m} + x_{j+1}B + x_jC \quad (j=0, 1, \ldots)$$

によって 32 ビット整数の列を生成する．ここで，$x_0, x_1, \ldots$ は $\mathbb{F}_2$ 係数，すなわち 0, 1 を成分とする 32 次元の横ベクトルである．B, C は $\mathbb{F}_2$ 係数の (32×32)行列であり，横ベクトルとの積が計算機 CPU のビット演算だけで高速に計算できるよう形を工夫したものである．$n>m>0$ は整定数である．

こう見ると MT 法は，通常の n 階線形漸化式において，整数の代わりに $\mathbb{F}_2$ ベクトルを用い，係数を行列に置き換えたものに過ぎない．$B=O$（零行列），$C=I$（単位行列）とすると §1.2 の GFSR に一致する．GFSR の漸化式に行列を導入したのが筆者-栗田の Twisted GFSR[18][19]であり，C を非正則な行列にすることで周期を大きな素数にする改良を行ったものが MT 法である．

線形合同法のような大きな法 N を使う生成法に対し，GFSR 法や MT 法の一つの利点は「ラウンドロビン」と呼ばれる計算機実装テクニックを使うことで，漸化式の階数（あるいは周期）によらず一定の速度で生成ができる点（[7, P. 28 Algorithm A] 参照），および高次元均等分布性（§4.2.2）を改善しやすい点にある．実際には MT 法では上の x_j をそのまま使うのではなく，Tempering と呼ばれる $\mathbb{F}_2$ 線形変換を施して上位ビットの均等分布性を改善したものを出力している．

皮肉なことに，欠陥のある 3 項 GFSR 法は広く普及したが，その後開発された問題のない $\mathbb{F}_2$ 線形生成法はなかなか広く普及しなかった．その大きな原因の一つは，90 年代に Knuth や Marsaglia のような乱数研究の大家が $\mathbb{F}_2$ 線形生成法に否定的な見解を述べていたことにある．（彼らは 3 項 GFSR 法を念頭においていたようである．）2003 年になって Marsaglia はこれを覆し，XORSHIFT という $\mathbb{F}_2$ 線形生成法を開発・提唱した[26]．ただし，この生成法は高次元均等分布性を調べていないため，乱数性に問題があることが Panneton-L'Ecuyer により指摘され，改良版が提案されている[29]．

4. 擬似乱数の評価法

上で見たように，擬似乱数発生の立場から見たとき，「乱数性」の真に実用的な定義はいまだ存在しない．したがって，擬似乱数の乱数性を肯定的に評価する方法も存在しない．

4.1 統計的検定

否定的に評価する方法はいろいろ存在する．「数列が乱数列である」ということを帰無仮説として，数列に対しある統計量を計算し，それが理論上の分布から離れすぎていないか，あるいは近すぎないか，を計り確率値の形で評価を行うのが統計的検定である[7, §3.3]．

統計的検定は物理乱数を含む全ての乱数発生法に適用できる．また，十分な規模と種類の検定に合格することは，擬似乱数発生法が満たすべき最低限必要なことである．しかし，どのような統計量に注目するべきかは指導原理がない．多くの検定が提唱され行われているが，それぞれの検定が何をとらえているかも，検定間の関係もいまひとつはっきりしない．

網羅的な統計的検定のパッケージとして Marsaglia の diehard[12]が知られているが，L'Ecuyer-Simard の検定[8]がより大規模化されておりまた新しい種類の検定も追加されている．周期の短い（$\sim 2^{32}$）生成法はほとんど後者の検定で棄却される．なお，MT 法はこれら全ての検定を通っている．

NII-Electronic Library Service

The Japan Statistical Society

統計分野の人には言うまでもないが，有意水準5%での棄却は20回に1回程度見られる．しかし，最近でも時々，メールにて「MT法をテストしたところ，5%での棄却が100回に5回も見られた」というような報告を受ける．基礎的な統計教育の必要性を感じる瞬間である．

擬似乱数の偏りが統計的検定で検出される場合，サンプル数を増大させるに従って確率値は極端なものになっていく．例えば，有意水準0.1%以下での棄却が何度でも繰り返しおこる．サンプル数を10倍にすれば，もっと極端な確率値が観測される．このようなことが起きれば，その擬似乱数は棄却されると言える．（この点は，サンプル数を増加させるのが困難な通常の統計的検定とは性質がやや異なる．）

なお，統計的検定を非常に大規模に行ったり，χ^2値の列にKS検定を適用するなどの2重検定[7, §3.3.1]を行うと，近似誤差（例えばχ^2分布の近似公式による誤差）が蓄積されて罪のない擬似乱数を棄却してしまうこともしばしばある．また，整数の実数化に伴う丸め誤差により棄却してしまうこともある．統計的検定で疑わしい結果を得たときには，性質の異なる複数の擬似乱数を検定して，問題があるのが擬似乱数の方なのか，検定方法の方なのかをはっきりさせておく必要がある．（筆者はMT法に対するこの種の間違った棄却の報告をいくつも受けている．）

4.2 理論的評価

実際に統計的検定を行うと，有意水準1%での棄却が10回に1回くらい観察されたりして，疑わしいのかそうでないのかわからずに延々と実験を繰り返す羽目に陥ることがしばしばある．

これに対し，理論的評価は非実験的な評価法である．擬似乱数を生成する漸化式によっては，生成数列が数学的な構造を有する．その構造について，ある数値的指標を導入して「望ましさ」を計るのが擬似乱数の理論的評価である．従って，漸化式のタイプごとに理論的評価の方法は異なるのが普通である．

理論的評価は，乱数性の評価として妥当かどうかの判断が難しく，わかりにくい．評価を理解するのに，その数学的意味を理解する必要性がある．反面，非常に非乱数性の検出力が高いテスト方法である．また，漸化式のパラメータ選択の際に，複数のパラメータ間の優劣をつけることができる．そのため，擬似乱数を開発する際は，まず漸化式を考案し，その漸化式に適した理論的評価により最適に近いパラメータを選び，それが生成する擬似乱数に対しさまざまな統計的検定を行って問題がないことを確認して開発を終える，というステップを踏むのが普通である．

理論的評価を用いれば擬似乱数に対する肯定的評価が可能だが，その評価指標値を最適化することが「乱数性」に関する最適化になっているかどうかはわからないことを注意しておく．例えば，周期は長いほど望ましいが，周期が長いものをとれば乱数性がいいというわけではない．

4.2.1 周期

代表的な理論的評価指標は，周期である．MT法は$2^{19937}-1$の周期を持つ．現在のBSD-C言語の標準擬似乱数randomは周期が約2^{63}である．市販の数表処理ソフトやフォートランに含まれている標準擬似乱数には，周期が2^{32}程度の線形合同法が多い．

周期は長いに越した事はないが，どのくらい必要かは定かでない．理論的根拠に乏しいが，シミュレーションで使う予定の乱数の個数の3乗以上の周期が望ましいとされている．これは，ある種の統計的検定を線形・非線形生成法に対して行うと，周期の3乗根くらいのサンプルサイズから偏りが検出されるという経験から導かれた目安である．

周期は状態集合の元の数$\#(S)$以下になるが，これらが一致したときその擬似乱数発生法は最大周期性を持つという．慣習的な理由により，周期が$\#(S)-1$のときに極大周期性を持つ

NII-Electronic Library Service

The Japan Statistical Society

という.

4.2.2 高次元均等分布性

シミュレーションにおいて，一組の操作にちょうど M 個の乱数が消費され，それを繰り返すと仮定する（例えば，M 個の確率変数を持つ関数を計算する場合など）．すると，オートマトン（定義 3.1）による擬似乱数発生法は，「初期値 s_0 から長さ M の数列

$$o(s_0), o(f(s_0)), \ldots, o(f^{M-1}(s_0))$$

を生成する

$$g : S \to O^M \qquad (3)$$

なる関数 g」であると考えることができる．ここに O は一個の出力がとりうる値の集合である．以下，S, O は有限集合と仮定する．

擬似乱数発生法が最大周期性（または極大周期性）をもち，かつ初期値 s_0 を S から一様にランダムにサンプルすれば（g による像である）出力列が O^M 上に一様に分布するとき，その擬似乱数発生法は M 次元均等分布性を持つという．

最大周期性より，一周期に渡って擬似乱数を発生すると全ての状態がちょうど一回ずつ実現される．従って，一周期に渡って出力を生成したとき，連続する M 個の出力の組（周期と同じ個数存在する）は O^M の全ての元をちょうど同数ずつ実現する．このような M の最大値を，その擬似乱数発生法の「均等分布の次元」という．M は大きい程よいが，f の全射性が必要条件なので

$$\#(O)^M \leq \#(S), \quad M \leq \lfloor \log \#(S) / \log \#(O) \rfloor$$

という自明な上限を持つ．等号が成立するとき，「最適な均等分布の次元を持つ」という．MT 法はこの性質をもち，$M=623$ 次元均等分布している．

この性質は「一周期に渡ってみれば，一様に M 次元空間の中に分布している」「初期値をランダムに選べば，一様に M 次元空間の中に分布している」といったことを保障する．しかし，通常実際に使われるのは一周期のごく一部である．また，「あまりにもバランスが良すぎるのではないか」という心配を聞くこともある．実際，乱数性の立場からはこの性質が何を保障しているのかははっきりしない．しかし，§1.1 で触れたように経験上次のことが観察される：「一周期での M 次元均等分布の保障された数列のごく一部を使った場合，連続する M 個以下の乱数の組に対する統計的検定はパスする.」

統計では多変量を扱うモンテカルロ法が良く使われる．上の経験則から言えば，変数の数に見合った高次元均等分布性をもつ擬似乱数発生法を用いることがより安全である．また，ランダムウォークをシミュレーションする場合には，「現在の位置」は今までに生成された擬似乱数の総和である．このためステップ数が上の M を超えると，統計的に有意な誤差を生じることがある（3 項 GFSR 法で顕著）．ランダムウォークに限らずマルコフ過程をシミュレーションする場合には，過去の擬似乱数出力列全てが現在の状態に影響を与えるため，高次元均等分布性をもつ擬似乱数が望ましい．（かといって，「高次元均等分布性の低い擬似乱数を用いると必ず結果が狂ってくる」というわけではない．経験的には，特に問題があるものを使わない限り大抵正しい結果が得られる．金融工学などでは 1000 個を超える確率変数がからんだ関数をしばしば取り扱うが，そのために 1000 次元以上の均等分布性をもつ擬似乱数を用いなければならないということはない.）

均等分布の次元の計算方法であるが，極大（ないし最大）周期の $\mathbb{F}_2$ 線形生成法の場合（す

NII-Electronic Library Service

The Japan Statistical Society

なわち状態集合 S，出力集合 O が $\mathbb{F}_2$ 線形空間で，次状態関数 f，出力関数 o が $\mathbb{F}_2$ 線形写像の場合）線形代数で行える．この仮定のもとで，M 次元均等分布性は（3）の出力列関数 g の全射性と同値となり，行列のランク計算に帰着される．（MT 法のように巨大な状態空間を持つ場合は計算時間がかかるため，冪級数格子による高速算法を使った．）なお，非線形生成法の場合は次元計算は大抵困難である．

さて，多くの場合 O は計算機の 1 ワード整数の集合，例えば 32 ビット整数の全体である．モンテカルロ法ではこれを $[0, 1)$ 区間に正規化することが多い．すると，実数の値としては，上位ビットは下位ビットよりも重要である．これを反映して，次の「v-ビット精度での均等分布の次元」がしばしば評価指標として用いられる．

定義 4.1. O の上位 v ビットをとる写像を $\mathrm{tr}_v : O \to \{0, 1\}^v$ とする．

$$\mathrm{tr}_v(o(s_0)), \mathrm{tr}_v(o(f(s_0))), \mathrm{tr}_v(o(f^{M-1}(s_0)))$$

が上の意味で M 次元均等分布しているとき，擬似乱数発生法は v ビット精度で M 次元均等分布しているという．そのような M の最大値を $k(v)$ であらわし，「v ビット精度での均等分布の次元」という．

再び全射性が必要条件なので，ちょっと計算すると次の上限を得る．

$$k(v) \leq \lfloor \dim(S)/v \rfloor \quad v = 1, 2, \ldots,$$

この上限が全ての v（例えば O が 32 ビット整数なら $v = 1, 2, \ldots, 32$）で達成されているとき，最適均等分布性を持つという．

MT 法は最適均等分布性を持っていない．$v = 1, 2, 4, 8, 16, 32$ では $k(v)$ はこの上限を満たしているが，例えば $v = 3, 5$ では上限の 93%程度を満たしているだけである．最適均等分布性を持つように出力関数を $\mathbb{F}_2$ 線形な範囲で調整することは可能だが，生成に時間がかかるようになる上，本当にそれが必要なことかが筆者には疑わしい．例えば $k(3)$ の実現値は 6240 次元であり，上限値は 6645 次元である．この差を（作為的でない）統計的検定によって検知するのは不可能に近いことに思える．

上位ビットだけに着目するのも不自然である．ある種の応用では下位ビットも重要になる．（MT 法では，下位ビットも良い高次元均等分布性を持っていることを確認してある．）また，何らかの調整により上限を達成しても，例えば一個おきに擬似乱数を使った場合には最適性は保障されない．さらに，同じ時間をかけるなら線形関数で変換するよりは，$\mathbb{F}_2$ 線形でない関数で出力を変換（フィルター方式）したほうがより安全な擬似乱数が生成できると考えられる．

線形合同法においては，高次元均等分布性に対応するものとしてスペクトル検定があるが，ここでは説明を割愛する（[7, §3.3.4] に解説がある）．

4.2.3 重み齟齬検定

擬似乱数列の特定のビットに着目して 0, 1 の擬似乱ビット列を得る．このビット列のもっとも素朴な統計的量は全体に占める 1 の個数であろうし，もっとも素朴な統計的検定はそれが二項分布からどれだけずれているかを見るものであろう．この検定を重み分布検定という．さて，F2 線形生成法の場合には，ある条件のもと，出力列にあらわれる 1 の個数の分布を正確に計算することができる．この分布を求めて二項分布からの乖離を定量的に求めるのが §1.2 でも触れた重み齟齬検定である [22]．

この検定は，$\mathbb{F}_2$ 線形生成法に固有の理論的評価法でありながら，他の理論的評価法と異なり統計的な意味が明白である．また，その擬似乱数が何個くらいの使用に耐えるのかの目安を与えることができる利点がある．

NII-Electronic Library Service

The Japan Statistical Society

5. 利用上の注意

5.1 初期化の問題

擬似乱数発生法を用いる際には，最初に初期状態を一つ選ぶ操作が必要である．これを初期化とよび，その際に与えるパラメータを初期シードという．

多くの擬似乱数で，初期シードは一個の 32 ビット整数である．このシードのサイズは，今日では多くの応用で小さすぎる．N 個の元からなる集合から，一様ランダムに重複を許して元を $\sqrt{N}$ 回抽出すると，重複が起こる確率は 1/2 を大きく超えてしまう（誕生日の衝突になぞり birthday paradox と呼ばれる）．従って，2^{16} 回程度初期化を行うと，どれか二つの初期化が全く同一の乱数列を生成する公算が高い．この点を考慮して，2002 年に初期化ルーチンを改善した MT 法 mt19937ar.c では，任意長の配列を初期シードとして与えることができるようになっている．

なお，高性能と信じられている多くの擬似乱数発生法において，初期値を系統的に（例えば 0, 100, 200, …などと）与えると，生成される数列の間に相関パターンが観察される．筆者はこの事実を和田維作氏[31]から教わったが，その後筆者-和田-蔵本-芦原により，GNU Scientific Library にある 58 種の擬似乱数発生法のうち，45 種がこのような欠陥を持つことがわかった[25]．これらは主に初期シードを初期状態へと変換する関数の選択の不注意に起因するものであり，必ずしも乱数発生法の漸化式の問題ではない．mt19937ar.c では非線形初期化関数を導入しており，この種の現象は観察されていない．

一般的に言えば，擬似乱数はなるべく初期化しないほうが良いことを注意しておく．長周期擬似乱数は巨大な状態空間をもっており，そのなかで初期シードから初期化ルーチンによって生成される状態はきわめて少なく，それらが何か望ましくない構造を持っている可能性が潜んでいるためである．MT 法でも，初期状態中のビットに 0 があまりにも多い場合には，数万回の生成の間乱数性が損なわれることが確認されている．(mt19937ar.c による初期化ではこのような初期状態はまず生成されない．) 速度を若干犠牲にして，この点を改良しさらに高次元均等分布性を最適にした擬似乱数発生法 WELL が，Panneton-L'Ecuyer-筆者によって得られている[30]．

5.2 さまざまな分布に従う乱数生成

統計などでの応用では，しばしば正規分布などの非一様分布に従う擬似乱数が必要となる．通常，(生成のしやすさから) 一様分布乱数をまず生成し，それを所望の分布に変換するのが普通である．分布関数の逆関数が高速に計算できる場合には，一様分布乱数を逆関数で変換すればよい．例えば，[0, 1) 上の一様分布に従う確率変数 X に対し，$-\log(1-X)$ を計算すれば指数分布が得られる．(余談だが，「線形合同法を MT 法に変えたらプログラムが動かなくなった」という質問メールのほとんどがこれにまつわるものである．上の式と同じ分布を持つ $-\log X$ で指数分布を得ている場合，$X=0$ となるとエラーが生じる．ある種の線形合同法では 0 が決して出力されないのでこの問題が生じないが，擬似乱数を MT 法に取り替えると 0 は正しい確率である $2^{-32} \fallingdotseq 43$ 億分の 1 程度の確率で出力されエラーが生じる．)

非一様分布への変換方法について筆者は詳しくないので，包括的な文献として[2]，日本語の文献として[6]を挙げるに留める．ここでも注意しておきたいのは，非一様分布乱数を一つ作るために多数の一様乱数が使われる場合が多いことである．典型的なのは，(あまり効率的ではないが) 標準正規分布乱数を発生させるのに一様乱数を N 個生成し $x_1, \ldots, x_N$ として，

NII-Electronic Library Service

The Japan Statistical Society

$$(x_1, \ldots, x_N - N/2) \Big/ \sqrt{\frac{N}{12}}$$

を求めて（近似値として）使用するというものである．このような場合，前述の通り元の擬似乱数はN次元以上の均等分布をしていることが望ましい．$N=12$がしばしば用いられているが，尖度が小さくなるなど近似誤差が大きい．一方，正規分布の精度を良くしようと思ってNを大きくすると，乱数によっては逆に分布に偏りが現れることがある．実際，BSD-C言語の現標準乱数randomで$N=34$として上の和の正規分布からの偏りを調べると，830万個程度のサンプル数を超えると有意水準1%未満で棄却される[23]．

統計分野の研究者から「非一様分布擬似乱数を生成する場合，非一様分布の近似精度を良くすると乱数性が悪くなる．この両者は両立しえない」という発言を聞いたことがあるが，上のような経験からであろうと思われる．MT法のような高い次元での均等分布性を持つ擬似乱数に関しては，このような説は当たらないと思われる．

5.3 並列計算機での利用

近年，計算機を多数並列に走らせての大規模シミュレーションが盛んである．並列化による高速化のためには，それぞれの計算機に擬似乱数発生法を実装するのが普通である．（並列計算機のアーキテクチャに依存しないシミュレーションを行うため，プロセスごとや素粒子ごとに擬似乱数発生法を実装することもある．）この際，それらの間の相関が問題になる．通常，同一の漸化式を用いて初期シードのみを変えて用いることが多いが，系統的に初期シードを与えると（例えば計算機に順番に振られたID番号をシードに用いたりすると）擬似乱数によっては§5.1で触れた相関が生じることがある．また，線形合同法などの状態空間が小さい擬似乱数発生法を多数の計算機で用いた場合，初期シードが異なっていても，birthday paradoxにより状態の衝突が（時間差つきで）生じ，ある計算機の使っている擬似乱数列が別の計算機の擬似乱数列をずらしたものになる可能性がある．これらを避けるためには，状態空間が十分大きな擬似乱数発生法を用い，初期化関数に非線形関数を用いるなどの工夫が必要である．MT法はこの意味で並列計算にも適している．（難点は，占有するメモリ空間が32ビット長624ワードと他の方法より大きい程度である．）

より抜本的な解決法として，計算機ごとに割り振られる擬似乱数発生法の漸化式の型を全部異なるものにしてしまえばよいが，並列度が高いとそれは難しい．そのため，「同一の型の漸化式を用いるが，IDごとにパラメータは変える」パラメトリゼーションという手法が用いられる（SPRNG[13]など）．筆者-西村は，ID番号を与えると，それを漸化式の中に組み込んだ小規模モデルのMT法の（高次元均等分布性の優れた）パラメータを発生させるDynamic Creatorというプログラムを開発し配布している[21]．ID番号が異なると漸化式の特性多項式が互いに素であることが証明されている点でSPRNGより良い独立性の保障を持っており，近年利用が広がりつつある．

6. ホームページと文献

MT法に関する一次情報は，めったに更新されないが[14]で見ることができる．ボランティアによるさまざまな言語への翻訳，非一様分布を生成するパッケージなどへのリンクもある．また，インターネットで検索してもいろいろ見つかる．（英語のページが多いので英語で検索すると良い．）

既存の擬似乱数の問題点に関しては和田維作氏のホームページ[31]に紹介がある．

擬似乱数の文献としては，日本語では[6]，英語では[7]，[27]を挙げておく．より新しい生

NII-Electronic Library Service

The Japan Statistical Society

成法を含んだ解説は，L'Ecuyer 氏のホームページ[8]からダウンロードできる．

本稿が想定する主な読者は「擬似乱数を統計などに利用する人」である．今まで筆者がユーザーから（対話や電子メールで）受けた質問・議論を元に，ユーザーが興味を持ちそうな話題と陥りやすい罠を中心に執筆した．筆者に質問や議論を交わしてくれた方々に感謝する．MT 法そのものの日本語による解説は[15][16]，重み齟齬検定については[17]にある．最近，簡明な MT の紹介記事[28]が二宮祥一氏により執筆された．

Acknowledgment 筆者は，マルコフ過程モンテカルロ法を「マルコフ過程のシミュレーションを行うもの」と全く誤解して執筆していた．この間違いのほか，多くの有益な御指摘をいただいた査読者に，心から謝意を述べたい．

参 考 文 献

[1] L. Blum, M. Blum, and M. Shub. A simple unpredictable pseudorandom number generator.SIAM J. Comput. 15 (1986), 364–383.
[2] Luc Deveroye, Nonuniform random variate generation. Springer-Verlag, 1986.
[3] ECRYPT Stream Cihper Project, http://www.ecrypt.eu.org/stream/http://eprint.iacr.org/2005/165
[4] Ferrenberg, A. M., Landau, D. P., and Wong, Y. J. (1992) Monte Carlo simulations: hiddenerrors from 'good' random number generators. Phys. Rev. Lett. 69 3382–3384.
[5] Fredricsson, S. A. (1975) Pseudo-randomness properties of binary shift register sequences.IEEE Trans. Inform. Theory **IT-21**, 115–120.
[6] 伏見正則，乱数，東京大学出版会，1989.
[7] Knuth, D. E. The Art of Computer Programming. Vol. 2. Seminumerical Algorithms 3rdEd. Addison-Wesley, 1997.
[8] L'Ecuyer, P. http://www.iro.umontreal.ca/~lecuyer/
[9] Lewis, T. G., and Payne, W. H. Generalized feedback shift register pseudorandom numberalgorithms. *J. ACM* 20, 3 (1973), 456–468.
[10] Lindholm, J. H. (1968) An analysis of the pseudo-randomness properties of subsequences of long *m*-sequences. IEEE Trans. Inform. Theory **IT-14**, 569–576.
[11] Lüscher, M. A portable high-quality random number generator for lattice field theory simulations,Computer Physics Communications, 79 (1994) 100–110.
[12] Marsaglia, G. http://stat.fsu.edu/pub/diehard/
[13] Mascagni, M. The Scalable Parallel Random Number Generators Library (SPRNG), http://sprng.cs.fsu.edu/
[14] 松本 眞：Mersenne Twister のホームページ http://www.math.sci.hiroshima-u.ac.jp/~m-mat/MT/mt.html
[15] 松本 眞「メルセンヌ・ツイスター／高次元均等分布擬似乱数発生法」数学セミナー 1998 年 3 月号, 27–33.
[16] 松本 眞「コイン投げで一儲けする方法?疑似乱数研究の現状」情報処理 Vol. 39 No. 11 (1998) 1166–1170.
[17] 松本 眞「コイン投げで儲ける：疑似乱数と符号理論」数理科学 2002 年 12 月号 8–13
[18] Matsumoto, M. and Kurita, Y. "Twisted GFSR Generators," ACM Transactions on Modeling and Computer Simulation **2** (1992), 179–194.
[19] Matsumoto, M. and Kurita, Y. "Twisted GFSR Generators II," ACM Transactions on Modeling and Computer Simulation **4** (1994), 254–266.
[20] Matsumoto, M. and Nishimura, T. "Mersenne Twister: a 623-dimensionally equidistributed uniform pseudo-random number generator" ACM Trans. on Modeling and Computer Simulation **8** (1998), 3–30. http://www.math.sci.hiroshima-u.ac.jp/ m-mat/MT/DC/dc.html
[21] Matsumoto, M. and Nishimura, T. "Dynamic Creation of Pseudorandom number generator," 56–69 in: Monte Carlo and Quasi-Monte Carlo Methods 1998, Ed. H. Niederreiter and J. Spanier, Springer 2000.
[22] Matsumoto, M. and Nishimura, T. "A Nonempirical Test on the Weight of Pseudorandom Number Generators" 381–395 in: Monte Carlo and Quasi-Monte Carlo methods 2000, Springer-Verlag 2002.
[23] Matsumoto, M. and Nishimura, T. "Sum-discrepancy test on pseudorandom number generators" Mathematics and Computers in Simulation, Vol. 62 (2003), pp 431–442.

NII-Electronic Library Service

The Japan Statistical Society

[24] Matsumoto, M., Nishimura, T., Hagita, M. and Saito, M. "Cryptographic Mersenne Twister and Fubuki Stream Cipher," Cryptology ePrint Archive, http://eprint.iacr.org/2005/165.
[25] Matsumoto, M., Wada, I., Kuramoto, A. and Ashihara, H. "Common Defects in Initialization of Pseudorandom Number Generators" submitted.
[26] Marsaglia, G. "Xorshift RNGs." Journal of Statistical Software 8, 1-6. See http://www.jstatsoft.org/v08/i14/xorshift.pdf
[27] Niederreiter, H. Random Number Generation and Quasi-Monte Carlo Methods. SIAM, 1992.
[28] 二宮祥一「松本眞さんの疑似乱数発生法について」数学通信 10 巻 2 号 (2005), 59-63, 日本数学会.
[29] Panneton, F. and L'Ecuyer, P. "On the Xorshift Random Number Generators", 2004, submitted. See http://www.iro.umontreal.ca/~lecuyer/papers.html
[30] Panneton, F., L'Ecuyer, P. and Matsumoto, M. "Improved Long-Period Generators Based on Linear Reccurences Modulo 2" To Appear in ACM Transactions on Mathematical Software.
[31] 和田維作のホームページ http://www001.upp.so-net.ne.jp/isaku/index.html

NII-Electronic Library Service