疑似的な一様乱数とベータ分布

中村　永友[1]　　土屋　高宏[2]

要　旨

ベータ分布による新しいタイプの一様乱数の生成方法を提案する．これは，真性一様乱数に似た乱数であるため準一様乱数と呼ぶ．結果としての一様乱数ではなく，統計的性質に基づいて生成されることがこの乱数の主な特徴である．もう 1 つの特性は，*Seeds* としての乱数の集合からほぼ無限の乱数を生成できることである．その数値的特性と優位性を，数値実験と疑似乱数としてのメルセンヌ＝ツイスター法と物理乱数の比較によって検証する．

キーワード：一様分布，準一様乱数，順序統計量，非復元抽出，ベータ分布

1　はじめに

本報告は一様乱数列の持つ統計的な性質を利用した一様乱数の生成法を提案する．数値実験やシミュレーション研究等において乱数を得るための方法として，物理乱数と漸化式タイプの乱数生成法（乗算合同法，メルセンヌ＝ツイスター法など）が主流であるが，第 3 の新たな方法を提案する．

これまで数多く提案されてきた疑似乱数の生成手法は，確率的あるいは統計的な理論や考え方の下で生成されていないことを指摘する．疑似一様乱数は「疑似」と冠しているとおり，ある種を基とする決定論的な方法（漸化式）で得られるものであり，結果的に得られる数列が乱数列であることが適合度検定などで保証されているのである．物理乱数はランダム（確率的変動をする）とされる物理現象から得られ，これも結果として乱数列と見なしているのである．

この第 3 の方法は，ある確率分布にしたがう乱数を 1 組用意し（数百～数千個程度＝n 個：*Seeds*），規格化をして，ここからすべての要素を非復元抽出することで新たな 1 組の一様乱数が得られる．乱数 1 個ずつではなく，n 個同時に一様乱数の集合を生成する方法である．もう少し詳しく説明する．*Seeds* となるある条件に従うベータ乱数を 1 組（n 個）用意し，これを累積したものが一様乱数列になる．さらに *Seeds* から非復元抽出を行い累積すると，また別の 1 組の一様乱数列ができる．この方法により理論上 $n!$ 通りの一様乱数列の集合ができるので，$n=500$ で約 10^{1134} 組の一様乱数集合が作成され，つまり $500\times10^{1134}\approx10^{1137}$ 個の一様乱数が作成可能である．たった 1 組の *Seeds* を用意し，非復元抽出をすることで，膨大な（ほぼ無限の）一様乱数が得られる．このような発想による乱数生成法はこれまでにない新たな方法で，一度に n 個生成されるので高速生成が可能なことが特徴である．

提案手法は統計的な性質を備え，かついわゆるこれまでの疑似乱数とは生成過程が異なるので，この手法で得られた一様乱数を「準一様乱数（quasi-uniform random number）」と呼ぶこととする[(1)]．

以下，第 2 節で関連する定理などを示し，第 3 節で提案手法とアルゴリズムについて，第 4 節は理想的なベータ乱数の生成法について，第 5 節では数値実験の結果を示す．第 6 節では本提案手法とその他の疑似一様乱数の比較を行う．

2　一様乱数生成に関する諸定理

最初にいくつかの定義と定理を示す．

定義 1（順序統計量）　区間［0, 1］の一様乱数 $\{x_1, x_2, ..., x_n\}$ を順序統計量にしたものを $\{x_{(1)}, x_{(2)}, ..., x_{(n)}\}$ と表記する．

定義 2（ベータ分布）　ベータ分布 Beta(α, β) は以下

[1] 札幌学院大学　経済学部；nagatomo@sgu.ac.jp.
[2] 城西大学　理学部；takahiro@josai.ac.jp.

で定義される：

$$f(x|\alpha,\beta)=\frac{x^{\alpha-1}(1-x)^{\beta-1}}{\mathrm{B}(\alpha,\beta)},\ (0<x<1).$$

$\mathrm{B}(\cdot,\cdot)$はベータ関数で，

$$E[X]=\frac{\alpha}{\alpha+\beta},\ V[X]=\frac{\alpha\beta}{(\alpha+\beta)^2(\alpha+\beta+1)}$$

である．

ちなみに，Beta(1, 1)は一様分布となるが，本論文ではこの文脈では使わない．

定理 1（順序統計量の分布） 区間［0, 1］における一様乱数に対する順序統計量 $u=x_{(r)}$ は，次の確率分布にしたがう（David and Nagaraja, 2003）：

$$f(u)=\frac{1}{\mathrm{B}(r,n-r)}u^{r-1}(1-u)^{n-r-1},$$

$0\le u\le 1$ で，

$$E(u)=\frac{r}{n},\ E(u^2)=\frac{r(r+1)}{n(n+1)},\ V(u)=\frac{r(n-r)}{n^2(n+1)}$$

である．

定理 2（差の分布） 一様分布にしたがう標本の順序統計量の差分 $x_{(k)}-x_{(j)},\ (k>j)$ は，ベータ分布 Beta$(k-j,\ n-k+j+1)$にしたがう（David & Nagaraja, 2003）．ここで，$1\le j<k\le n$ である．

系 1（隣接順序統計量の差の分布） 定理 2 から，i 番目と $i+1$ 番目の順序統計量の差分 $x_{(i+1)}-x_{(i)}$ は，ベータ分布 Beta(1, n)にしたがう．

系 2　ベータ分布 Beta(1, n)の密度関数は次の通りである．

$$f(x|1,n)=\frac{x^0(1-x)^{n-1}}{\mathrm{B}(1,n)}=n(1-x)^{n-1},\ (0<x<1).$$

系 3（ベータ分布にしたがう確率変数の和）　$n+1$ 個のベータ分布 Beta(1, n)にしたがう乱数の和 b は，$b\approx 1$ である．数値的な検証結果を付録 A に示す．

証明　Beta(1, n)の期待値が $\frac{1}{n+1}$ であることから，$n+1$ 個のベータ分布にしたがう乱数の和が 1 になることは次式から明らかである．

$$E\left[\sum_{i=1}^{n+1}X_i\right]=\sum_{i=1}^{n+1}E[X_i]=\sum_{i=1}^{n+1}\frac{1}{n+1}=1.$$

系 4　X が一様乱数列であることの必要十分条件は，X の順序統計量の差がベータ分布であることである．

系 5　一様乱数の順序統計量の差がベータ分布にしたがうことを利用すると，ベータ分布に対する適合度検定で間接的に一様性の検定が可能である．

3　一様乱数の生成アルゴリズムとその変形

3.1　基本アルゴリズム

定理 2 の隣り合う一様乱数の差がベータ分布にしたがうことを利用して，ベータ乱数を使った疑似一様乱数を求めるアルゴリズムを示す．

基本的な考え方は次の通りである：

STEP1　ベータ乱数

$$b_1, b_2, \ldots, b_{n+1} \overset{iid}{\sim} \mathrm{Beta}(1, n)$$

を用意し，$\boldsymbol{B}=\{b_1, b_2, \ldots, b_{n+1}\}$とする．

STEP2　累積値

$$x_1=b_1,\ x_2=x_1+b_2,\ x_3=x_2+b_3,\ \ldots,\ x_n=x_{n-1}+b_n$$

で求められる $\boldsymbol{X}=\{x_1, x_2, \ldots, x_n\}$ が一様乱数列の順序統計量となる．

STEP3　必要な数だけ $\boldsymbol{X}$ から非復元抽出する．

STEP4　さらに多くの乱数が必要なときは，$\boldsymbol{B}$ を非復元抽出して $\boldsymbol{B}'$ を作り，STEP2，STEP3を繰り返す．

図 1：ベータ乱数と一様乱数の順序統計量

ここで注意点を挙げておく．まず，系 3 より，STEP1で任意のベータ乱数の和が $E[\sum b_i]=1$ であること，すなわち $\sum_{i=1}^{n+1}b_i\approx 1$ であること，そして，STEP2では，$x_i=x_{(i)}$ となり，つまり順序統計量になっていることである．

以上を基本として，$\boldsymbol{B}$ の生成方法，$\boldsymbol{B}$ からの抽出方法，$\boldsymbol{X}$ からの抽出方法によって，一様乱数生成のアルゴリズムの変形がいくつか考えられ，以下に示すと共に，その具体的な乱数生成のアルゴリズムを付録に示す．

3.2　一様乱数生成の変形

基本アルゴリズムから変形できる部分を示す．

(a1)STEP1で，$\boldsymbol{B}$の総和が1になるように規格化し，これを$\tilde{\boldsymbol{B}}$とする．

(a2)$\boldsymbol{B}$の代わりに理想的な確率分割点を$\boldsymbol{B}^*$を何らかの方法で求める．

(a3)$\boldsymbol{B}$あるいは$\boldsymbol{B}^*$に$t_i \overset{iid}{\sim} G(0, \tau^2)$を加えて揺動させる．$\sum_{i=1}^{n+1} t_i = 0$，$\tau^2 > 0$ and $\tau^2 \approx 0$.

(a2) は第4節で示すような方法で求めることができる．(a3) は確率分布として $G(0, \tau^2)$ をどのように選ぶかで，種々の方法が考えられる．

4 *Seeds* の作り方

では，準一様乱数を作るための基本となるベータ乱数について考える．何の工夫もなくベータ分布Beta(1, n)にしたがう乱数を$n+1$個生成しても良いが，何らかの方法で「理想的なベータ乱数」を1組作ることを考える．以下に3種類の方法を示す．

4.1 数値的に求める(1)

第1の方法として，ベータ乱数を生成し，順序統計量$x_{(i)}$にして，その期待値$E[x_{(i)}]$を求める方法である．例えば$n=10$で，1,000,000,000回の実験で求めると，表1の左列となる．これらの和は1.0である．10,000回程度で求めるとちょうど1にはならないので，理想的なものとするには規格化が必要である．

4.2 数値的に求める(2)

前項では$E[x_{(i)}]$で求めたが，一見同じように見えるが，結果が異なる第2の方法を提示する．それは，一様乱数を1組生成し，その順序統計量の差分を求め，その期待値をとる方法である．つまり，$u_i \overset{iid}{\sim} U(0, 1)$, ($i=1, ..., n$) を生成し，$u_{(i)}$ として，$x_{(i)} = u_{(i+1)} - u_{(i)}$ の期待値 $E[x_{(i)}]$ を求める．$n=10$で，1,000,000,000回の実験で求めると表1の右列となる．このように，前項との結果とは異なり，微妙に異なっている．これらの値は生成過程から，総和は必ず1である．

表1：理想的な Seeds を数値的に求める

ベータ乱数による	一様乱数による
0.009009286018075020	0.008264594112927058
0.018821170725493540	0.017355802920068130
0.029602956065012787	0.027456824718271274
0.041583508257186360	0.03882044326216984
0.055081955412787355	0.05180707052473218
0.070572666156743930	0.06695811429431960
0.088796486666656110	0.08514012096877491
0.11101957508487831	0.10786675719546632
0.13969611032342796	0.13816953152927391
0.18066327295457618	0.18362475628522062
0.25515265143355930	0.27453598418918680

4.3 理論解法

何らかの理論により *Seeds* 求めることが考えられ，1つのアイディアとして，密度関数の確率分割する$n+1$個の点を求めることができる．つまり，

$$\int_{x_{(i)}}^{x_{(i+1)}} \mathrm{Beta}(1, n)dx$$

の確率を$\sum x_{(i)} = 1$という制約条件の下で求めることが可能である．この解を得るための自由度が大きいため，各区間の確率をだいたい$\frac{1}{n+2}$にするなどの別の条件を加える必要がある．これについては今後の検討課題とする．

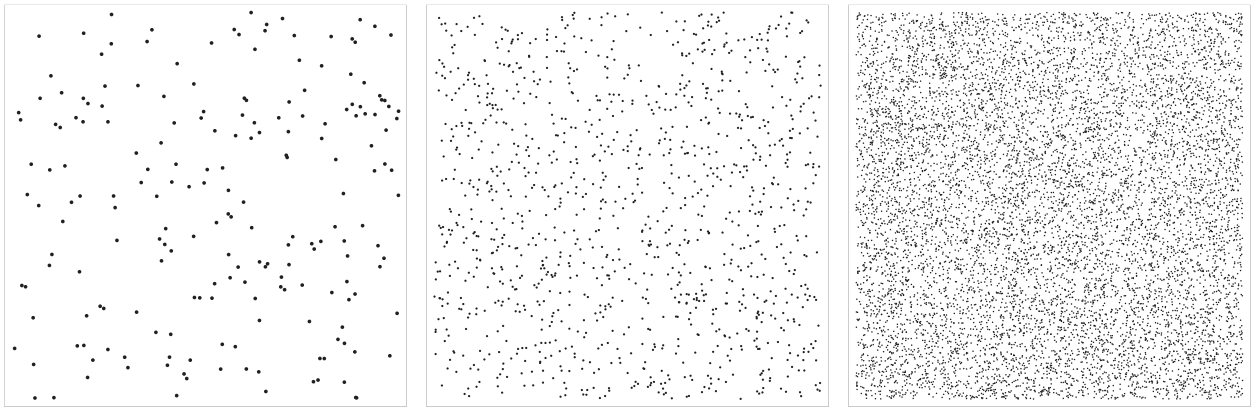
5 数値実験

準一様乱数の生成結果を示す．図2は，(a2) の方法で得られた準一様乱数の散布図である．図3は図2 (b) のx座標値の順序統計量の差のヒストグラムである．実線はベータ分布 Beta(1, 1200)の密度関数である．

図3が非常にきれいなベータ分布をしている理由は明らかで，それは理想的なベータ乱数そのものだからである．逆に言えば，図3から図2ができているのである．

6 議論

乱数の利用者は何らかの保証のある乱数列を使っている．つまり，乱数列を生成する機構の一様乱数性が保証されていればよいのである．その保証とは，疑似乱数であれば，その生成機構や周期性が数理的に確認できること，物理乱数であればその物理的背景や雑音を出す素子の性質が明確であることであろう．いずれの場合も，生成された乱数列を仮説検定（L'Ecuyer & Simard, 2007の TestU01など）によって検証することができる．以下では提案手法である準一様乱数，疑似乱数，Gould (1992) で引用された乱数の比較を行いつつ，乱数について議論する．



$(a)n = 200$　　$(b)n = 1200$　　$(c)n = 10000$

図 2：理想的なベータ乱数から作った一様乱数を 2 次元で表現

（a3）により得られた一様乱数列を 2 組用意し，(x_i, y_i) として組み合わせて打点した散布図．左から n＝200，1200，10000である．

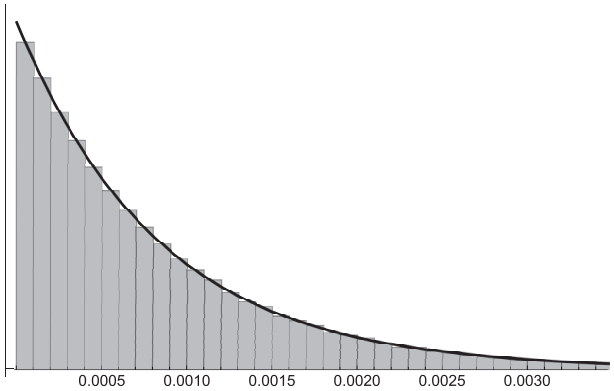


図 3：提案手法の順序統計量の差のヒストグラム

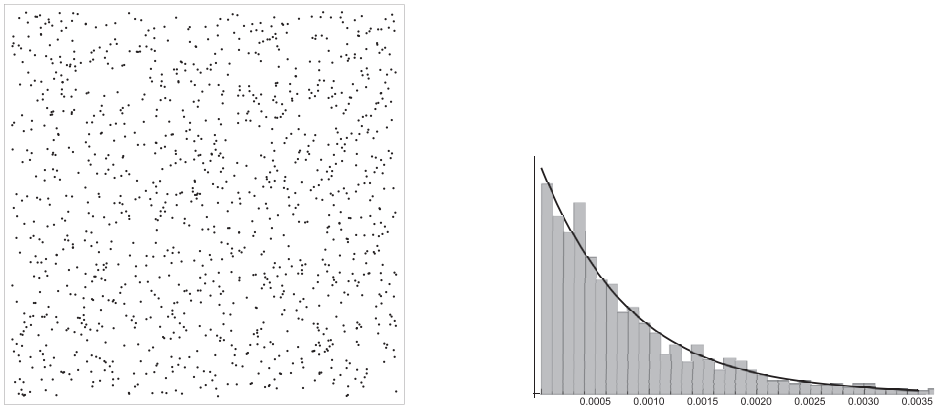
6.1　乱数の比較

ここでは，提案手法の準一様乱数，従来の疑似乱数（メルセンヌ＝ツイスター法），物理乱数（統計数理研究所，1998），Gould（1992）の例示との比較をする．

図 2 は本提案手法である．図 3 は図 2(b) の x 座標の順序統計量の差の分布である．実線はベータ分布の曲線である．非常にきれいにヒストグラムにあてはまっていることが確認できる．これは当然で，理想的なベータ乱数だからである（ベータ分布にあてはまるように作っている）．

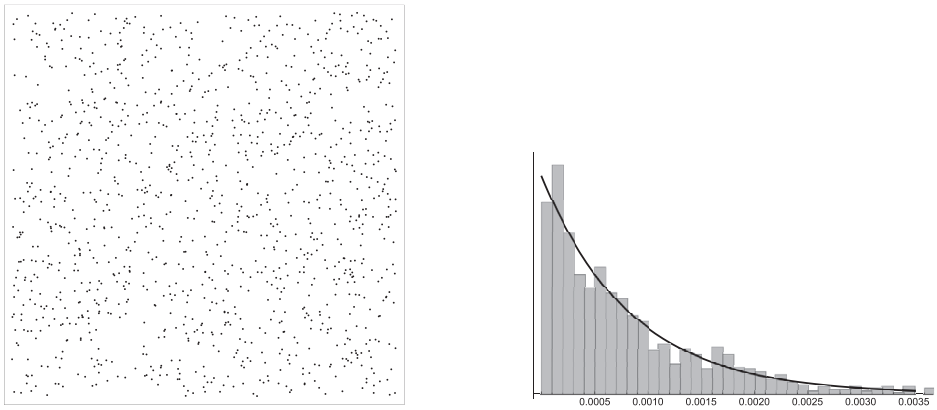
図 4(a) はメルセンヌ＝ツイスター法による疑似一様乱数である．2400個の疑似乱数を生成し，2 個ずつ組み合わせて散布図に示した．(x_i, y_i)，$(i=1, ..., 1200)$ である．図 4(b) は x 座標の順序統計量の差の分布である．実線はベータ分布の曲線である．提案手法に比べて多少凸凹しているが，全体としてベータ分布のあてはまりは悪くはない．

図 5 は統計数理研究所のデータライブラリにある物理乱数である．最初のファイルの先頭から1200組（2400個）取り出して，散布図と順序統計量の差の分布を示した．



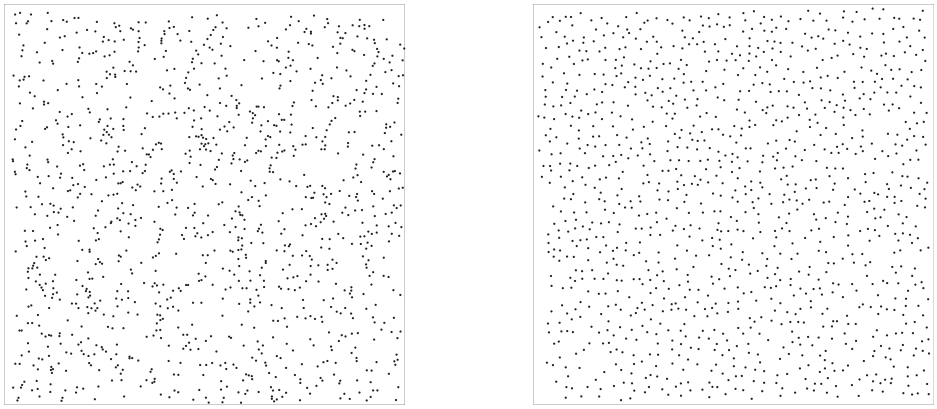
(a) 散布図　　$(b)x$ の順序統計量の差の分布

図 4：疑似一様乱数（メルセンヌ＝ツイスター法）による生成（n＝1200）



(a) 散布図　　(b)x の順序統計量の差の分布

図 5：物理乱数



(a)Stars　　(b)Worms

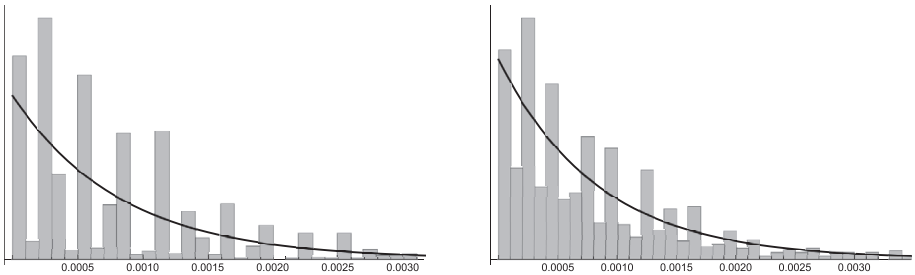
図 6：Gould の stars（左）と worms（右）

この図は Gould（1992）の266-267ページに示されていて，物理学者 Ed Purcell によるコンピュータプログラムによる例示である．左は「星」というタイトルで，夜空の星はランダムであるということを前提に，それを模したものである．このようなランダムであるものは意図せず何らかのパターン（星座のようなものや何らかの形など）が見えてくるのが特徴である，と記述している（心理学ではパレイドリア現象と呼ばれる）．また右は「虫」というタイトルがついている．これはある種の虫の幼虫が自分のテリトリ（抑制ゾーン）を作る状況を模したものである．人間の見た目では，こちらの方が正方形を埋め尽くすランダムなパターンと思いがちであるが，しかしこれは，一定の間隔の中でランダムに打点している．

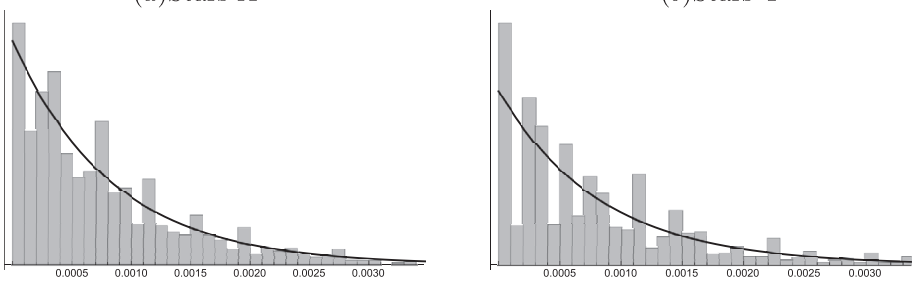
図 6 は Gould（1992）での例示である（詳細は図の注釈を参照）．図 6 は Ed Purcell のプログラム（Gould，1992）により生成された乱数で，左はいわゆる乱数（疑似乱数）であり，右はある虫のテリトリを模したものである．乱数について詳細を知らない一般人は，右側を一様乱数だと思うだろうが，乱数を扱ったことのある研究者などは左側がそれだと選択するであろう．図 7 は図 6 の各座標値の順序統計量の差のヒストグラムである．これらは全体としてかなり凸凹があるが，ベータ分布の傾向があるという感じである．

次に，各一様乱数のデータセットの一様乱数性を検証する．単純に順序統計量の差の分布をとり，ベータ分布に対する適合度検定を行った結果を図 8 に示す．いずれも 1 組の乱数の集合から，リサンプリングしたデータに対する検定結果である．横軸はデータセットからのリサンプリング数 n，縦軸は適合度検定の p 値である．指定した n に対して10,000回リサンプリングして，p 値を求め，その平均を求めている．ここでは重複せずにリサンプリングしている．

図 8(a) の準一様乱数は，n を大きくすればするほどベータ分布をしていることがわかる．(b) は p 値が0.5付近で分布していている．(c) はサンプリング数が400以上で p 値が下がる傾向が見られる．(d) と (e) は Ed Purcell によるコンピュータプログラムで生成された一様乱数とみられるものの結果であるが，n が100前後以上で p 値はほぼ 0 であるので，ベータ分布



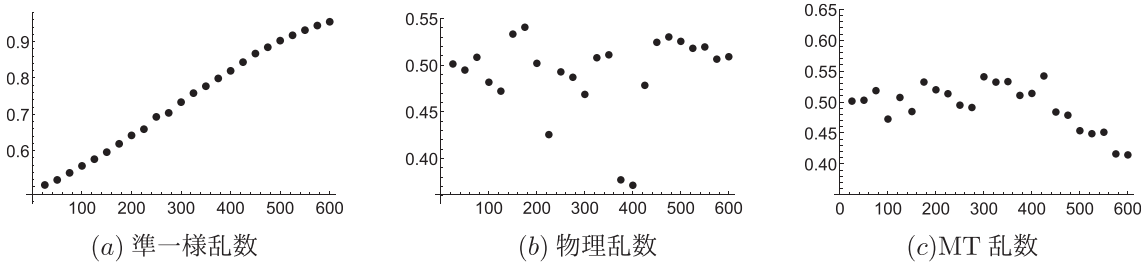
(a)Stars-X　(b)Stars-Y



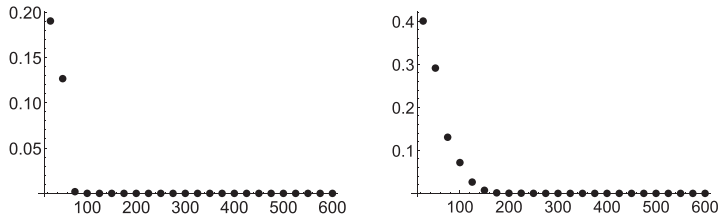
(c)Worms-X　(d)Worms-Y

図 7：stars と worms の各座標軸データの順序統計量の差の分布

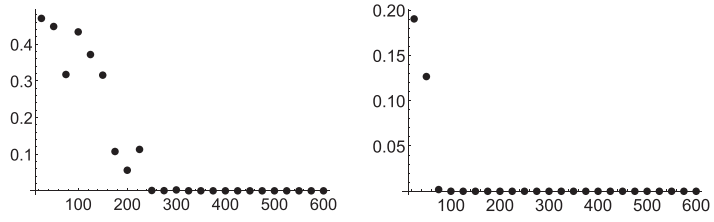
Stars と Worms のデータを各座標軸のデータごとに 1 次元データに落として，それを順序統計量に直し，隣り合うデータの差分を集計したヒストグラム．順序統計量の差はベータ分布にしたがうので，ベータ分布の密度関数を同時に表示している．この図は物理学者 Ed Purcell によるコンピュータプログラムによる出力である．



(a) 準一様乱数　(b) 物理乱数　(c)MT 乱数



(d)Stars-X　(e)Stars-Y



(f)Worms-X　(g)Worms-Y

図 8：各一様乱数からサンプリングした適合度検定の結果

横軸がサンプリングしたデータ数 n，縦軸が適合度検定の結果の p 値である．n を固定して，一様乱数のセットからサンプリングして，適合度検定から p 値を求め，これを10,000回行った平均を示している．

とは見なせない，という結果である．(*f*) と (*g*) も Ed Purcell によるコンピュータプログラムによるものであるが，(*f*) は *n* が200程度それ以上ではそう見なせず，(*g*) は *n* が50程度以上では全くそう見なせない．

以上を概観すると，本提案手法が最も一様乱数性が高いことがわかる．図 8(*b*) と (*c*) からは，物理乱数が理想的な乱数と考えれば，メルセンヌ＝ツイスター法による疑似一様乱数はこれに近い性質を持っていることがわかる．一方，Ed Purcell によるコンピュータプログラムはその中身がわからないため，順序統計量の差がベータ分布にしたがうという視点からは，いずれも一様乱数性がかなり低いことが確認できる．

6.2 優れた疑似乱数との比較

準一様乱数では理論上 *Seeds* として *n* 個のベータ乱数列を用意すると，*n*! 通りの一様乱数列ができることになる．メルセンヌ＝ツイスター法の最大周期は $2^{19937}-1$ であり，これはだいたい 10^{6001} である．この最大周期以上の準一様乱数を得るためには，$n>2080$ 程度であればよいことになる．本報告では示していないが，(a3) のアルゴリズムであれば，せいぜい $n=500$ 程度であればほぼ無限に一様乱数を生成することができる．

6.3 準一様乱数生成に乱数を使用するとは？

準一様乱数を得るために，物理乱数や疑似一様乱数を使うことになる．つまり，ベータ乱数の生成，非復元抽出をするときに何らかの一様乱数を使わざるを得ない．ある意味矛盾している．しかしながら，統計的に保証された乱数が超大量に生成されることを考えれば，無から有は生じないので，その過程で物理乱数や疑似一様乱数を使うことは，ある意味仕方のないことである．

6.4 円周率やネイピア数の乱数列性

円周率 π やネイピア数 *e* は無理数として知られているが，乱数列か否かという問は昔からある．現在の結論としては，統計的には一様乱数列と見なせないことはない．これらが無理数であることは背理法などで証明可能であるが，これらが真性の「乱数的」であるかということの証明は未解決である．仮に仮説検定によって既知の桁数までが乱数列として否定できなければ，乱数としての利用は可能であろう．2020年段階で円周率は小数点以下50兆桁まで計算されているので，これらを何らかの形式で記録しおいて逐次利用が可能であろう．しかし，その「記録」をどのようにするかが大きな問題である．乱数としての数字が記録された乱数表を持つことは，電子計算機でメモリや記録媒体での占有領域の問題から，可能な限り小さいものが最善と考えられてきた．この理由から π や *e* は乱数としては使いづらいこともあり，漸化式による疑似一様乱数生成法が最も用いられている．本報告の提案法と例えばメルセンヌ＝ツイスター法のメモリの総占有領域と比較では，*Seeds* となる種のベータ乱数列の保持，それらをメモリにロードする一定の記録する領域，主に非復元抽出などのプログラムソースコードなど，メモリ等を使用するため，これまでの慣習から外れるものである．しかし，本提案手法は，一様乱数生成・取得法の第 3 の方法であり，統計的に一様乱数性の性質を備えた方法であることが確率論・統計学としては興味深い．

7 おわりに

提案手法の特色を改めてまとめると，(1) 一様乱数を 1 個ずつ生成するのではなく，一様乱数の集合（一様乱数列）として生成すること，(2) これを生成するための種となる集合 *Seeds* を用意することで，理論上 *n*! 通りの一様乱数列が生成できること（*n* は種の要素の個数），(3) 乱数生成の基本アルゴリズムを基礎として，その生成法について数種類の変形が考えられることである．この考え方はこれまでにない新たな方法である．*Seeds* となる数列の集合に関する理論や，詰めなければならない点は多々あり，これらは今後の研究課題である．

謝辞

本研究は2020年度札幌学院大学研究奨励金 SGU-BG2020-01の助成を受けた．

参考文献

[1] Dabid, H. A. and Nagaraja, H. N. (2003). Order Statistics, Third Edition, Wiley.
[2] 情報・システム研究機構統計数理研究所. 物理乱数ライブラリ, https://www.ism.ac.jp/ismlib/jpn/ismlib/data.html.
[3] L'Ecuyer Pierre and Simard Richard (2007). TestU01: A C library for empirical testing of random number generators, ACM Transactions on

Mathematical Software, 33, August 2007 Article No.: 22, https://doi.org/10.1145/1268776.1268777, http://simul.iro.umontreal.ca/testu01/tu01.html.
[4] Stephen Jay Gould (1992). Bully for Brontosaurus: Reflections in Natural History, W. W. Norton & Company.（広野喜幸, 松本文雄, 石橋百枝（訳）(1995). がんばれカミナリ竜〈上・下〉進化生物学と去りゆく生きものたち, 早川書房.）

注釈

(1) Low-Discrepancy Sequence（LDS）という高次元の数値積分に有利で, 準モンテカルロ法に用いられる数列のことを, かつて準乱数と呼んでいた. しかし, 現在これは超一様分布列と呼ぶのが一般的である.

付録

A　系 3 の数値的検証

数値実験で系 3 を確認する. Beta(1, n)にしたがう乱数を $n+1$ 個生成してその和 b を求めることを 1,000,000回繰り返してその平均 $E[b]$ を以下に示す.

n	$E[b]$
10	1.00036
20	0.999918
50	1.00014
100	1.00022
200	0.999913
500	1.00001
1000	0.999999
2000	1.00001
5000	1.00001

このように, 系 3 が正しいことが示された.

B　提案アルゴリズムの詳細

まず, 基本アルゴリズムを示し, 次に提案する各種アルゴリズムを示す.

(a1) アルゴリズム 1

STEP1　ベータ分布 Beta(1, n)にしたがう乱数を $n+1$ 個生成する. $\{x_1, x_2, ..., x_{n+1}\}$ とする.

STEP2　x_i の累積値を求める. $y_1=x_1, y_2=y_1+x_2, ..., y_{n+1}=y_n+x_{n+1}$

STEP3　$\{y_1, y_2, ..., y_{n+1}\}$ が $(0, 1+\varepsilon)$ 区間の一様乱数となる. $\varepsilon=y_{n+1}-1$ である.

アルゴリズム 1 によってできた $\{y_1, y_2, ..., y_{n+1}\}$ は順序統計量となっていて, $y_{i+1}-y_i$ がベータ分布にしたがう. しかし, $\{x_1, x_2, ..., x_{n+1}\}$ は乱数であるため, その総和 y_{n+1} は理論上は 1, つまり $E(y_{n+1})=1$ であるが, 実際には $y_{n+1}\approx 1$ である. つまり, y_{n+1} が 1 を超えることもあれば, そうならないこともあるのである. これを克服するために, 微調整を行ったのが次のアルゴリズムである.

(a1′) アルゴリズム 1′

STEP1　ベータ分布 Beta(1, n)にしたがう乱数を $n+1$ 個生成する. $\{x_1', x_2', ..., x_{n+1}'\}$ とする. $s=\sum_{i=1}^{n+1} x_i'$ を求めて, $xk=x'k/s$ を計算して, $\boldsymbol{X}=\{x_1, ..., x_n\}$ とする.

STEP1′　$\boldsymbol{X}$ から非復元抽出をして $\boldsymbol{X}^*=\{x_1^*, x_2^*, ..., x_{n+1}^*\}$ とする.

STEP2　x_i^* の累積値を求める. $y_1^*=x_1^*, y_2^*=y_1^*+x_2^*, ..., y_{n+1}^*=y_n^*+x_{n+1}^*$

STEP3　$\{y_1^*, y_2^*, ..., y_n^*\}$ が (0, 1) 区間の一様乱数となる.

STEP4　STEP3で n 個の一様乱数ができるが, それ以上の個数が必要であれば, STEP1′～STEP3を実行すれば, さらに n 個の一様乱数が得られる.

STEP1で規格化していることが重要である.

(a3) アルゴリズム 3（理想的なベータ乱数からの生成＋揺動）

STEP1　理想的なベータ分布 Beta(1, n)にしたがう数列を $n+1$ 個用意し, $\boldsymbol{X}=\{x_1, ..., x_n\}$ とする. これはすでに $\sum_{i=1}^{n+1} x_i=1$ となっている.

STEP1′　$\boldsymbol{X}$ から非復元抽出と揺動因子 t_i を加える. $X'=\{x_1', x_2', ..., x_{n+1}'\}=X^*+T=\{x_1^*, x_2^*, ..., x_{n+1}^*\}+\{t_1, t_2, ..., t_n, 0\}$. ここで, $\sum t_i=0$ である.

STEP2　x_i^* の累積値を求める. $y_1^*=x_1', y_2^*=y_1^*+x_2', ..., y_n^*+1=y_n^*+x_{n+1}'$

STEP3　$\{y_1^*, y_2^*, ..., y_n^*\}$ が区間 [0, 1] の一様乱数となる.

STEP4　STEP3で n 個の一様乱数ができる. STEP1′～STEP3を実行すれば, さらに n 個の一様乱数が得られる.

The Proceedings of the Research Institute of Sapporo Gakuin University Vol.8, 57–65 (2021)

Resembling a Truly-Uniform Random Number Generator via a Beta Distribution

Nagatomo NAKAMURA[1]
and
Takahiro TSUCHIYA[2]

Abstract

We propose a new type of the uniform random number generation method via a beta distribution. We call it a quasi-uniform random number because it is made up of the statistical properties of true random numbers. Another property is that almost infinite random numbers can be obtained from a typical set of random numbers as seeds. Its superiority and numerical characteristics were verified by numerical experiments and comparison with the Mersenne-Twister method as a pseudo-random number and physical random numbers.

Keywords: Beta Distribution, Order Statistics, Quasi-Random number, Sampling without Replication, Uniform Distribution.

[1] Department of Economics, Sapporo Gakuiun University; nagatomo@sgu.ac.jp.
[2] Department of Mathematics, Josai University; takahiro@josai.ac.jp.