2つの優れた乗算合同法乱数：開示

中澤 直也[1]/ 中澤 宏[2]

1. 現在最優秀の大周期乗算合同 (MC) 法乱数生成機構#001

幸運にも我々の最初の発見となって、現在まで最も優れた統計的性質で用い得る大周期 $T \approx 2^{53}$ を誇る MC 乱数生成機構#001 を以下開示します。

(構成) 奇素数の積の法 d を持つ MC 生成機構 (d, z) です。

$d = p_1 p_2 = 18055400005099021 \approx 2^{54.00}$

SG 素数の部分法 $p_1 = 134265023 \approx 2^{27.00}$

SG 素数の部分法 $p_2 = 134475827 \approx 2^{27.00}$

p_1 の原始根乗数 $19061252 \approx 2^{24.18}$

p_2 の原始根乗数 $77600525 \approx 2^{26.21}$

孫子の定理による法 d での乗数 $z \equiv 7759097958782935 \approx 2^{52.78}$

使用可能最小公倍数周期 $T = 4513849934089543 \approx 2^{52.00}$

同じ法で使用可能周期と検定結果も共有する乗数は:

$$\begin{aligned} z^{-1} &\equiv 8723774547862110 \approx 2^{52.95} \\ -z &\equiv 10296302046316086 \approx 2^{53.19} \\ -z^{-1} &\equiv 9331625457236911 \approx 2^{53.05} \end{aligned}$$

(検定結果 1) (p, z^i) の $1 \le i \le 11$ に対する[3]一般化 2 次検定値

1.08678338　1.23476055　1.09373237　1.14778981　1.13682785

1.16390618　1.09784908　1.21656428　1.52552804　1.34934813

7.69460527

(検定結果 2) 3-6 次正単体基準スペクトル検定評価値

[1]573-0081 枚方市釈尊寺町 28-18-103 枚方乱数工房
[2]573-0081 枚方市釈尊寺町 28-18-103 枚方乱数工房
[3]後の記述との関係で z^{11} までの結果を示します。

1.13600074　1.04031015　1.10996227　1.21389160

(検定結果 3) 3-6 次正単体基準稜検定評価値

(3 次最大最小稜検定値)　0.78489424　1.18938572

(4 次最大最小稜検定値)　0.73780699　1.17913686

(5 次最大最小稜検定値)　0.83524952　1.20173353

(6 次最大最小稜検定値)　0.71002135　1.20574247

検定評価値は優れています。特に最大稜検定値は優秀です。詳記すれば、これは法 d で乗数 z が作る (d,z) 格子が正 (単体) 格子に近い事の検定、(d,z) 格子の最短の l 個の (1 次独立な) 基ベクトルのそれぞれに可能な符号 $\pm$ を付けて考えた 2^l の組み合わせ[4]で作られる単体の最大稜 2^l 個の最小のものの長さ L_l、で正 l 格子の最短稜の長さを割ったものです。難しいですが、正 l 格子の最隣接格子間隔 ${L_l}^*$ に対して次最隣接格子間隔は $2^{1/2}$ 倍になります。(d,z) 格子で ${L_l}^*$ から増大変形した長さ L_l が $2^{1/2}$ 倍に届かない、${L_l}^*/L_l < 2^{-1/2} \approx 0.7071$ となる、即ちそれよりも最大稜検定値が大きいという事は L_l が正 l 格子の『次最隣接格子間隔』よりも小さく、この (d,z) 格子は正 l 格子からの小さい変形だとする事ができるのです。使える周期は長大で、高速なパソコン上で専一に生成させても 2 年以上かかります。[5] ここではこの開示で可能になった『孫子の定理を用いた乱数生成計算と関連した意味』に立ち入るため、節を新しくして進みます。

2. 孫子の定理と 2 素数の積の法の MC 乱数計算方法

孫子の定理は法が 2 つ以上の互いに素な部分法の積の場合に理論的に明快な構造理解を与えます。[6] そればかりではなく、孫子の定理は計算量と計算過程についても重要です。ただそれは法が具体的にどのような『互いに素な部分法』から成るか、を指定しないと議論ができません。我々が#001 を発見し、詳しい記述理解を求める現時点は新しい機会です。この#001 の場合の上で議論します。難しいが読解に御尽力下さい。

孫子の定理の結論から再出発します。法 $d = p_1 p_2$ の乗算合同法生成機構

[4]全体の符号は任意だから、意味のある異なる組み合わせは 2^{l-1} 個です。

[5]この長大さはこの生成機構を任意の多次元空間の上の各点に幾何学的隣接格子点の乱数と無相関と見える様分布させて同程度の使い得る周期 T に亙るシミュレーションを可能にします。

[6]例えば Euler の関数の乗法性の明解な理解。

(d, z, n) の整数乱数列 $\{x_0, x_1, x_2, \cdots\}$ は次の形です:

$$\{x_k \equiv \{nz^k \pmod d\} |\ k = 0, 1, 2, \cdots, \quad 0 < x_k < d\}.$$

諸変数と定数を列の整数番号 $k = 0, 1, 2, \cdots$ に対して次の様に定義します。

$$\begin{gathered}
z_1 :\equiv z \pmod{p_1}, \quad z_2 :\equiv z \pmod{p_2}, \\
n_1 :\equiv n \pmod{p_1}, \quad n_2 :\equiv n \pmod{p_2}, \\
{x^{(1)}}_k :\equiv n_1 {z_1}^k \pmod{p_1}, \quad {x^{(2)}}_k :\equiv n_2 {z_2}^k \pmod{p_2}, \\
D_1 :\equiv {p_2}^{-1} \pmod{p_1} \equiv {p_1}^{-1} {p_2}^{-1} / {p_1}^{-1} \pmod{p_1}, \\
D_2 :\equiv {p_1}^{-1} \pmod{p_2} \equiv {p_1}^{-1} {p_2}^{-1} / {p_2}^{-1} \pmod{p_2}.
\end{gathered}$$

孫子の定理は MC 整数乱数 $\{x_k |\ k = 0, 1, 2, \cdots\}$ が $\{{x^{(1)}}_k |\ k = 0, 1, 2, \cdots\}$ 部分列と $\{{x^{(2)}}_k |\ k = 0, 1, 2, \cdots\}$ 部分列との shuffling(トランプカードの混ぜ合わせ) の形で得られる、と示します:

$$\{x_k :\equiv (p_2 D_1) {x^{(1)}}_k + (p_1 D_2) {x^{(2)}}_k \pmod d |\ k = 0, 1, 2, \cdots\}.$$

実際この式を法 p_1 で見れば列 $\{{x^{(1)}}_k \equiv n_1 {z_1}^k \pmod{p_1} |\ k = 0, 1, 2, \cdots\}$ である、法 p_2 で見れば列 $\{{x^{(2)}}_k \equiv n_2 {z_2}^k \pmod{p_2} |\ k = 0, 1, 2, \cdots\}$ である事は一目瞭然です。ここで指摘したいのは、この形が明らかにする、私達が漸く発見した孫子の定理の数値的魔術です。

補題．(2 つの互いに素な部分法による孫子縮小)

法 $d = ef$ は 2 つの互いに素な部分法 $e > 0$ と $f > 0$ の積であるとする。任意の整数 $A > 0$ に対して次が成り立つ。

$$\mathrm{mod}(eA, d) = \mathrm{mod}(eA, ef) = \mathrm{mod}(e \bmod (A, f), ef) = e \bmod (A, f).$$

(証明) A の f による除法等式 $A = qf + r$、q は商、$r = \mathrm{mod}(A, f) < f$ は余り、を取ると

$$\mathrm{mod}(eA, d = ef) = \mathrm{mod}(efq + er, ef) = \mathrm{mod}(er, ef) = er = e \bmod (A, f)$$

が成り立ちます。これは tautology かと見えますが、A の値の $\mathrm{mod}(eA, d)$ の計算は $\mathrm{mod}(A, f) < f$ と縮小された答を取ってよいと示します。 ■

これによて今や我々は#001 の MC$(d = p_1 p_2, z, n)$ 乱数を integer*8 と real*8 だけの算術で得る事ができる、そして乱数列そのものは法 p_1 と p_2 のいずれも integer*4 の 2 つの部分列の生成でよい、何も失われる事はない、と目覚しい理解に至ります。節を替えて議論を続けます。

3. #001 乱数の計算

次の数値的状況 **(1)-(2)** が明らかとなります。

(1) integer*4 の $i=1$ や 2 に対する部分乗数 z_i を掛けて法 p_i を取る部分 MC 乱数列

$$\{x^{(i)}{}_k \equiv n_1 {z_i}^k \ (\mathrm{mod} p_i)|\ k=0,1,2,\cdots,\quad i=1,2\}$$

を作る部分は、高々integer*4 を 2 つ掛け合わせて法 p_i で integer*4 に引き戻す作用だけですから、すべて integer*8 の変数で行なえば精度や周期の損失に関わる事は全くありません。

(2) 問題はある列番号 k でシミュレーションに用いる一様独立乱数を作る操作に生じます。実際に用いる MC 乱数列

$$\{nz^k \ (\mathrm{mod}\ d=p_1p_2)|\ k=0,1,2,\cdots,\quad i=1,2\}$$

の構成は上の通りですが、具体的に整数番号 $k=0,1,2,\cdots$ を止めて、対応する部分 MC 整数乱数に孫子の定理の係数を掛けて法 $d=p_1p_2$ で足し合わせなければなりません。これを具体的に計算プログラムに組み上げた中澤 直也の研究 (2021 年 12 月) を次に開示します。

最終目的は整数列 $\{x_k \equiv mz^k|\ k=0,1,2,\cdots\}$ から一様独立乱数の良い統計的外見を持つ実数列 $\{v_k=x_k/d|\ k=0,1,2,\cdots\}$ を得る事です。難しい本質は整数列にあり、それは前小節の**補題**から次の計算になります。

$$\begin{aligned} x_k &\equiv \mathrm{mod}(nz^k,d) \\ &\equiv \mathrm{mod}(p_2 * \mathrm{mod}({p_2}^{-1},p_1) * x^{(1)}{}_k + \\ &\qquad +p_1 * \mathrm{mod}({p_1}^{-1},p_2) * x^{(2)}{}_k, p_1p_2). \end{aligned}$$

下の**第 1 図**は MC#001 乱数を 1000 万個計算する fortran プログラムです。

第 1 図　#001 乱数 1000 万個の計算プログラム

```
main program
implicit integer*8(i-n),real*8(a-h,o-z)
common ip1,ip2,id,ad,iz1,iz2,mz1,mz2,ip2mp1,ip1mp2
ip1=134265023
ip2=134475827
```

```
id=ip1*ip2 ！id はおよそ 2^54 です
ad=id
iz1=19061252
iz2=77600525
n1=10 ！初期値 1 は 10
n2=13 ！初期値 2 は 13
ip2mp1=52577007
ip1mp2=81816271
mz1=mod(n1,ip1)
mz2=mod(n2,ip2)
do i=1,10000000
call random(rand)
end do
      ......
end

subroutine random(rand)
implicit integer*8(i-n),real*8(a-h,o-z)
common ip1,ip2,id,ad,iz1,iz2,mz1,mz2,ip2mp1,ip1mp2
mz1=mod(mz1*iz1,ip1)
mz2=mod(mz2*iz2,ip2)
mz1a=mod(mz1*ip2mp1,ip1)
mz2a=mod(mz2*ip1mp2,ip2)
az=mod(ip2*mz1a+ip1*mz2a,id)
rand=az/ad
return
end
```

(第 1 図終り)

このプログラムが要した CPU 時間は 0.281sec / 1000 万個です。

一方これとよい対照となるのが優れた素数-原始根の MC 生成機構#M001 です。これは素数 $p = 17179869989 \approx 2^{34.00}$、原始根乗数は $z = 7928410072 \approx 2^{32.86}$ です。2019 年に発見されました。出力乱数としては#001 と同様に高い統計精度を誇り、周期はおよそ $2^{33.00}$ です。しかしその倍精度をほんの少し超える大きさの出力のために途中 4 倍精度の実数による法 p での計算を必要とし、同じ 1000 万個の乱数出力にはほぼ 10 倍の CPU 時間を要し、その 2^{33} 程度の周期は#001 の周期の 2^{-20} 倍と小さく、実用にはならず、出力乱数は 2^{-32} 単精度に毛の生えた程度の大きさしかありません。

この#M001 より約 10 倍高速で、しかも十分な倍精度実数乱数を 2^{53} の長周期生成できる#001 の性能は比べるべくもありません。生成には倍精度整数、倍精度実数の演算だけでよい、これは孫子の定理の与える奇跡です。算出された倍精度乱数 1000 万個末尾からのさらに 100 個の出力を第 2 図として示します。

large **第 2 図: プログラム第 1 図末尾からの 100 個の計算結果**

0.653816355434 0.162395903492 0.666319058508 0.192823573723
0.489788498203 0.327381692216 0.006207372410 0.817190447249
0.639382522876 0.999243182851 0.717807517328 0.582888069563
0.751959446280 0.456610909409 0.201265518413 0.197352588136
0.185468833692 0.026325012527 0.798951425190 0.980168183205
0.728774197785 0.895636674003 0.746279846438 0.334966215203
0.163132201425 0.161807776678 0.478463418819 0.402555313497
0.412925462471 0.228549325709 0.116935094385 0.887686052660
0.748053624507 0.372387517800 0.401887611920 0.513438563398
0.218008135464 0.479107340785 0.371799991246 0.610473874869
0.495998588197 0.704020149239 0.125946074052 0.689497113384
0.296979898817 0.664141855353 0.967378082658 0.861373665256
0.146986132091 0.320681156594 0.293103638455 0.410906693576
0.830103955630 0.320270139018 0.924042828676 0.087350373543
0.943936287549 0.994736329597 0.408045545191 0.472523592537
0.765124568761 0.630798202897 0.873021775270 0.612753906188

0.124335863299 0.816173368849 0.470611672694 0.880574814564
0.668048788679 0.380757635873 0.439593072176 0.532108985606
0.235689661806 0.023857729188 0.606829757993 0.762549693572
0.627268755976 0.592982603016 0.803692768860 0.555192595035
0.568754093418 0.482014270564 0.450138517310 0.960974827043
0.500236510179 0.285096197971 0.920893782638 0.842851188064
0.083495098650 0.555523403307 0.712500499476 0.525529497885
0.921528283135 0.667901000917 0.272780491599 0.962922804725
0.924506422608 0.495041819614 0.783468131560 0.851983710989

(第 2 図終り)

繰り返しになりますが、孫子の定理が与える計算の速さ、計算に必要なのは integer*8 と real*8 だけである便宜、は見事です。ここ示した、統計が同様に優れている MC 乱数#M001 を#001 を見比べて、乱数生成機構として多くの示唆を得る事ができます。現在までの所、単独の倍精度 integer*8 以内の範囲素数 p を法とする生成機構ではまだ優れた正単体基準合格の統計を持つ『素数-原始根』の MC 乱数生成機構が見つかっていません。すべてを調べたのではないので『ない』とは断言できませんが、あったとしても#M001 の例が示すように実用的ではありません。この事は $p \approx 2^{50}$ 程度までの integer*8 の倍精度の単独素数の法であっても、途中で 4 倍精度 real*16 の計算を要して、しかも周期は短く、得られる乱数の実数精度も現在のシミュレーションでは普通である倍精度 real*8 に足りません。実際には 2 つの integer*4 の異なる奇素数を組み合わせた法の MC 生成機構がが 10 倍も高速に算出でき、real*8 として十分な精度の乱数を与える事ができるのです。孫子の定理がこの様な場面で発揮する魔術的な力は、#001 や次の#003 の発見で漸く私達は気付きました。数の神様の深遠な御配慮、御恵みと言うべきでしょうか。勿論孫子の定理を使う事が正単体基準で初めて可能になった事は我々発明者の誇りであり、強調してもよいでしょう。

3. 大規模で良好な統計精度の MC 乱数#003

優れた MC 乱数生成機構の探索は年単位の時間を要する苦しい作業です。生成機構#003 これは漸く 2020 年 5 月に中澤直也によって発見されました。我々は #001 と比較して#003 とどちらを用いたいか、という自問に直面しましたが、正直

な所予算をシミュレーションにつぎ込んで研究を行うとすると、矢張り最良の素材が望まれ、#001 が優位と考えます。しかし#003 も優れた結果で、劣るとは考えられません。異なる乱数でも同様な結果が得られる、とシミュレーションを確かめる事は重要ですから、こちらも詳しく述べます。

(構成) 奇素数の積の法 d を持つ MC 生成機構 (d, z) です。

$d = p_1 p_2 = 18015370515269401 \approx 2^{54.000}$

NN 素数の部分法 $p_1 = 134224829 \approx 2^{27.000}$

NN 素数の部分法 $p_2 = 134217869 \approx 2^{27.000}$

p_1 の原始根乗数 $95967890 \approx 2^{26.516}$

p_2 の原始根乗数 $4256141 \approx 2^{22.021}$

孫子の定理による法 d での乗数 $z = 16048994718289548 \approx 2^{53.833}$

使用可能最小公倍数周期 $T = 2251921280853338 \approx 2^{51.000}$

同じ法で『使用可能周期と検定結果も共有する乗数』は:

$$\begin{aligned} z^{-1} &\equiv 10990185200333827 \approx 2^{53.29}\, vspace2mm \\ -z &\equiv 1966375796979853 \approx 2^{50.80}, \\ -z^{-1} &\equiv 7025185314935574 \approx 2^{52.64}, \end{aligned}$$

となります。 **(検定結果 1)** (p, z^i) の $1 \leq i \leq 11$ に対する一般化 2 次検定値

1.12378644 1.22759925 1.15381455 1.07582363 1.12113014
1.90830600 2.56595210 1.64729694 1.10578807 1.10728840
2.12669792

(検定結果 2) 3-6 次正単体基準スペクトル検定評価値

1.14537815 1.06716995 1.13487872 1.21563615

(検定結果 3) 3-6 次正単体基準稜検定評価値

(3 次最大最小稜検定値) 0.77772641 1.16750024
(4 次最大最小稜検定値) 0.74018574 1.20907497
(5 次最大最小稜検定値) 0.68729723 1.23300972
(6 次最大最小稜検定値) 0.69782364 1.23425488

検定評価値は優れていますが、一般化 2 次検定は z^5 までしか合格しません。こ

れは『格子上に分布させる場合』にはすこし弱点になりますが、ここでは触れません。1 つの格子点に 3 個の相続く MC 乱数』までが最大数です。#001 では 4 個まで原理的には可能ですから、その点では少し見劣りがしますが、他の点では #001 と変わりません。

この#003 での MC 乱数生成は、孫子の定理を利用して#001 と全く同様に行なう事がでます。詳細は**第 1 図**から直ちに得られますから再記はしません。