Lounge A of HRF

Representation Theorem of Random Number Generators on Computers

Naoya Nakazawa/ Hiroshi Nakazawa

(January 31, 2024)

1. Random Numbers on Computers

Consider a simulation problem with a random number sequence on computers. Nowadays solid state memories are not much expensive and we are not surprised by 1 teraBytes ($\simeq 2^{40}$ Bytes) of memories. Yet this amounts only to 2^{37} 8 Bytes random numbers.[1] At present and in near future we shall not be able to store and use excellent random numbers solely on memory.

Random numbers on computers are expected for use in large scale simulations. The simulation programs face severe problems in their corrections, *debugging*. Typically we need exactly identical random number sequences to be generated; we would like to see how the corrected programs will give changes to results. Therefore, random numbers on computers should have the *reproducibility*, that is, they should emit a completely identical sequence.[2] And, random number sequences should have the *transportability* that changes in computing language and / or computers themselves do not alter the otputs.

As the whole of necessary random numbers cannot be included in memory, we have to *produce* them by computing. Reproducibility and transportability demand the *computing* to be on integer arithmetic; the number of times for random number generation is huge, and any real arithmetic cannot prevent accumulations of truncation errors and others which depends on computers and computing languages. Integer

[1] This amount of random numbers will be used up on desktop computers in 5 hours.

[2] In this sense we shall feel hindrance in the use of *natural* random numbers such as the irreproducible outputs of radio-active counters.

arithmetic is the sole way to avoid difficulties without uncontrollable accumulation of errors. This need for integer arithmetic, however, causes new type of difficulties to which we have not accustomed.

Retrospectively speaking, the circumstance that force us to depart from real arithmetic is a bless. We shall find a structure valid generally for the representation of random number. As we see soon, generators are sternly and universally restricted to be *multiplicative congruential (MC) metnod* by the very nature of the *integer* arithmetic. An associated decisive merit is that MC generators allows for examinations on the goodness of emitted random numbers. Remember this exclusive merit of MC generators decisively.

A classical conceptual problem ever raised was whether a sequence of numbers computed by numeircal programs can be called *random* at all. We now think that the aim of random number generators is to emit *a sample sequence* that has the *disguise of random outlook*, not the whole of sample processes taking real values according to some probability measure structures. All matters of our concern are whether *a* sequence of numbers emitted from a generator may be regarded as random, for simulations or for computer games. We thus rephrase the problem as that of *assumptions and examinations*. This is answered by knowing properties of the emitted sequences. We go into discussions of this *properties* to be emitted on computers filling some tight requirements.

What we need to endevor after are following **(A)-(C)**.

(A) We should take a sufficiently large integer $z \gg 1$, and generate a sequence of integers $\{x_0, x_1, \cdots, x_{T-1}\}$ satisfying

$$\{x_0, x_1, \cdots, x_{T-1}|\ 0 \le x_k < z,\ \ k = 1, 2, \cdots, T\}$$

up to a necessarily large T.

(B) As random numbers we should produce rational numbers[3]

$$\{u_k := x_k/z|\ 0 \le u_k < 1, \quad k = 0, 1, \cdots, T-1\}.$$

(C) We should test to select excellent sequence $\{u_k|\ k = 0, 1, \cdots\}$ or its generator that negates in the weakest possible way the stochastic assumption that it is a sample process of uniform and independent random numbers.

[3] This is a fundamental departure from mathematical random numbers assumed to take values according to a probability measure on the real axis and to be set out to the proof of some ergodicity.

How can we find a generator giving sample processes satisfying these requirements? Well, the road ahead is very harsh, but we have a versatile approximation theorem arising from the structure of *rational number sequences*.[4] We herefter discuss this comprehension, which will also indicate ways to select excellent generators.

We leave considerations to generate random numbers with other distributions. This is justified by the existence of precision-guaranteed transformations from uniform random numbers, Please refer to the standard textbooks, typically D. E. Knuth.[5] Hereafter we concentrate on methods to generate uniform and independent random numbers.

2. To the Multiplicative Congruential (MC) Generators

Aside the classical method to throw dice, the oldest way to compute random numbers is the multiplicative congruential (MC) method.[6] We need the minimum of definitions and the knowledge on integers to proceed into unambiguous discussions on this way.

Definition 1. (Primes) Let x, y, z denote integers.
(A) If an integer $x > 0$ has no positive divisor other than 1 and x itself, then x is named a prime.
(B) If an integer x is divided by an integer y, then y is named a divisor of x. If an integer x is divided by a prime $p > 1$, then p is named *a prime facor* of x. **(End of Definition 1)**

Suppose primes are arranged as $1 < p_1 < p_2 < \cdots$ in the increasing order. Any integer $x > 0$ may be tried of divisions by them from below, and prime factors of x are determined in this way.[7] We define:

Theorem 2. (Least common divisor LCD) Define on integers a, b the set I of integers

$$I := \{Aa + Bb|\text{ integers } A \text{ and } B \text{ take all possible values}\}.$$

Let $m > 0$ be the smallest *positive* integer in I. Then m is the greatest

[4] This miraculous clue works only with rational outputs from computers, not with any *real* random numbers.

[5] D. E. Knuth, *The Art of Computer Programming*, Vol. 2, *Seminumerical Algorithms* (3rd edition) p.119.

[6] D. H. Lehmer: *Mathematical methods in large-scale computing units*, Annals Comp. Lab. Harvard **26** (1951) pp. 141-146.

[7] A small anxiety is: What will come about if we alter the order of divisions by primes? There is a proof of Gauss that such alterations make no problem. We shall later see a proof on this point.

common divisor (CCD) of a and b, to be denoted (a, b).
(Proof) The set I includes

$$a = 1 \cdot a + 0 \cdot b, \quad b = 0 \cdot a + 1 \cdot b.$$

Let the smallest *positive* integer m in I have the form $m = Xa+Yb$. And let an arbitrary integer n in I has the form $n = X'a + Y'b$. Divide n by m and denote q for the quotient and r for the remainder in $0 \leq r < m$, We show $r = 0$. For, the equation of division is

$$n := X'a + Y'b = qm + r = qXa + qYb + r.$$

This implies

$$r = (X' - qX)a + (Y' - qY)b, \quad 0 \leq r < m.$$

Thus, $r \in I$ holds true. Since m is the smallest positive integer in I, we have $r = 0$. In other words, the smallest posibive $m \in I$ divides any integer $n \in I$. In particular, m is the greatest common divisor LCD of a and b. ■

The final preparation is the arithmetic of integers with a modulus.

Lemma 3. (Equivalence modulo d) Take an integer $d > 0$. If integers a and a' satisfy

$a' = a + kd$ holds with an integer k, or

$a' - a$ is divided by d,

then a' and a are defined to be equivalent modulo d. This equivalence relation is denoted as $a' \equiv a \pmod d$. The following **(A)-(B)** hold true.
(A) If $a \equiv a' \pmod d$ holds, then $a' \equiv a \pmod d$.
(B) if $a \equiv a' \pmod d$ and $a' \equiv a'' \pmod d$ are true, then $a \equiv a'' \pmod d$ is also true.
(C) If integers $a, b, c, \cdots$ and $a', b', c', \cdots$ satisfy $a \equiv a' \pmod d$, $b \equiv b' \pmod d$, $c \equiv c' \pmod d$, $\cdots$, then any function $f(a, b, c, \cdots)$, formed by addition, subtraction and multiplication of variables with arbitrary integer coefficients, fulfill equivalence relations

$$f(a, b, c, \cdots) \equiv f(a', b, c, \cdots) \equiv f(a', b', c, \cdots)$$
$$\equiv \cdots \equiv f(a', b', c', \cdots) \pmod d.$$

Namely, any number of integer variables in the function $f(a, b, c, \cdots)$ may be replaced with their equivalents modulo d at any computing

stage retaining the equivalence relation unaltered.
(Proof) (A)-(B) The equivalence $a \equiv b \pmod d$ is the same that $a - b = kd$ or $a = b + kd$ holds with a positive or negative integer k. Conclusions **(A)** and **(B)** are obvious.
(C) Typically, suppose $a' = a + kd$ with an integer k, As

$$a'b = (a + kd)b = ab + kdb$$

holds with any integer kb, we have $a'b \equiv ab\,(\mathrm{mod} d)$. Any other equivalence relations are manifestly true. ■

Verbally. the above **(C)** may be rephrased that any function $f(a, b, c, \cdots)$ formed by addition, subtraction and multiplication with integral coefficients may be replaced any of its elements by d-equivalent integers without changing the original quantity f modulo d. In computing, we may exploit this to reduce the magnitude of integers at any stage. To us hyuman being, however, the replacement or the reduction makes the meaning of equations dim, so that we may well retain the original structure of f till the end of computing.

Definition 4. (Multiplicative congruential (MC) generator) Take an integer $d > 2$ for the *modulus,* with an integer z coprime to d as the *multiplier.* We need also to choose an integer n named *seed,* which is coprime to d in the interval $0 < n < d$ and fix the start of the sequence of random numbers. An MC generator defined by the triple (d, z, n) emits a *multiplicative congruential sequence* $\{x_0, x_1, x_2, \cdots\}$ *of integers, to be called multiplicative congruential (MC) integer sequence,* recurrsively by the following modular arithmetic:

$$\begin{aligned}&x_0 \equiv n \pmod d, \quad 0 < x_0 < d,\\&x_k \equiv zx_{k-1} \equiv nz^{k-1} \pmod d, \quad 0 \le x_k < d, \quad k = 1, 2, \cdots,\end{aligned} \; .$$

Finally, the sequence of uniform random numbers

$$\{r_k := x_k/d |\, 0 \le r_k < 1, \quad k = 0, 1, 2, \cdots\}$$

is emitted from the MC generator (d, z, n). **(End of Definition 3)**

There seems to be nothing special in this definition of MC random numbers However, we shall find a new miraculous view on this MC random number generators.

3. The Approximation Theorem

We start a thought experiment to reach the approximation theorem in the title. Consider first on possible forms of *rational numbers* emitted from a generator of uniform random numbers. They should generally be fractions of integers,

$$\{x_1/z_1,\ \ x_2/z_2,\ \ x_3/z_3,\ \ \cdots,\ \ x_T/z_T\},$$

where $0 \le x_k < z_k$ and $z_k > 0$ should hold for all $1 \le k \le T$. A small but significant point is that we may deform these fractions so as to have one and the same denominator z. Namely, define

$$\zeta := \mathrm{GCD}(z_1, z_2, \cdots, z_T).$$

Introduce the least common multiple of denominators

$$z = \mathrm{LCM}(z_1, z_2, \cdots, z_T) := z_1 z_2 \cdots z_T/\zeta^{T-1},$$

as we do in additions of rational numbers. The finite length of T makes this opeation possible by the finite length of T. The magnitude of T is suggested by the classification of integers on computers:

single precision or 4-Bytes integer x has $|x| < 2^{32}$,
double precision or 8-Bytes integer x has $|x| < 2^{64}$,
quadruple precision or 16-Bytes integer x has $|x| < 2^{128}$.

At present all scientific simulations employ double precision real numbers, and we shall need random numbers prepared by double precision integers. The cirdumstance will be similar with computer games.[8] If $T \simeq \mathrm{O}(2^{64})$ holds, then our desktop computers will need a year to exhaust all these random numbers, and this length T will be the best for present random number generators. So, please be prepared that $T \approx 2^{64}$ may arise.

These devices justify us to consider only the sample of the output integer sequence of the form,

$$\{x_k |\, 1 \le k \le T\},\quad 0 \le x_k < z.\ \ T < 2^{64}\},$$

with random number outputs $\{x_k/z|\, 1 \le k \le T\}$.

We now notice that this length T sequence may be considered as the

[8] Experientially, single precision random numbers with $T \simeq \mathrm{O}(2^{32})$ will be used up in an hour at the best.

first period of the decimal, or better the z-mal, cyclic sequence of a rational number X,

$$\begin{aligned} X &= 0.x_1x_2x_3\cdots x_T\; x_1x_2x_3\cdots x_T\; x_1x_2x_3\cdots \\ &:= 0.\dot{x}_1x_2\cdots\dot{x}_T \\ &= \{x_1z^{-1}+x_2z^{-2}+\cdots+x_Tz^{-T}\}\sum_{j=0}^{\infty}z^{-jT} \\ &= \{x_1z^{-1}+x_2z^{-2}+\cdots+x_Tz^{-T}\}\cdot\frac{1}{1-z^{-T}} \\ &= \frac{x_1z^{T-1}+x_2z^{T-2}+\cdots+x_T}{z^T-1} =: \frac{n}{d}. \end{aligned}$$

Here the final fractional form n/d is taken to be *reduced* or, that the numerator n and the denominator d are coprime. And we assume $X > 0$ or, assume that integers in $\{x_k|\ 0 \leq x_k < z\}$ are not all 0.

Let us summarize. The numerator n and the denominator d of $0 < X < 1$ are coprime by definition. Also, d is a divisor of $z^T - 1$. Hence d has no common divisor with z, or d and z are coprime.

Now a magical stage of random number sequence reconstructions from the triple (d, z, n). As X is taken to be an infinite sequence repeating the period $0.x_1x_2\cdots x_T$, there holds $0 < X \leq 1$; remember the usual decimal number $0.\dot{9} = 1$. We may proceed with the division $X = n/d$ stepwilse. In the first z-mal division of n by d, we devide nz by d. The equation at the first z^{-1} place is

$$nz = q_1d + r_1 = x_1d + r_1, \quad 0 < r_1 < d.$$

where the first quotient q_1 should be x_1, and the first remainder is r_1. In the place of z^{-2} we have

$$zr_1 = x_2d + r_2.$$

Operations continue to the k-th place with

$$zr_{k-1} = x_kd + r_k, \quad k = 2, 3, \cdots, T$$

with the $(k-1)$-th remainder r_{k-1}, the k-th quotient $q_k = x_k$ and the k-th remainder r_k.

A miracle arises here with a very simple trick: A small remainder is followed by a small next quotient. Define $r_0 := n$. Divide above equations with dz. We have then

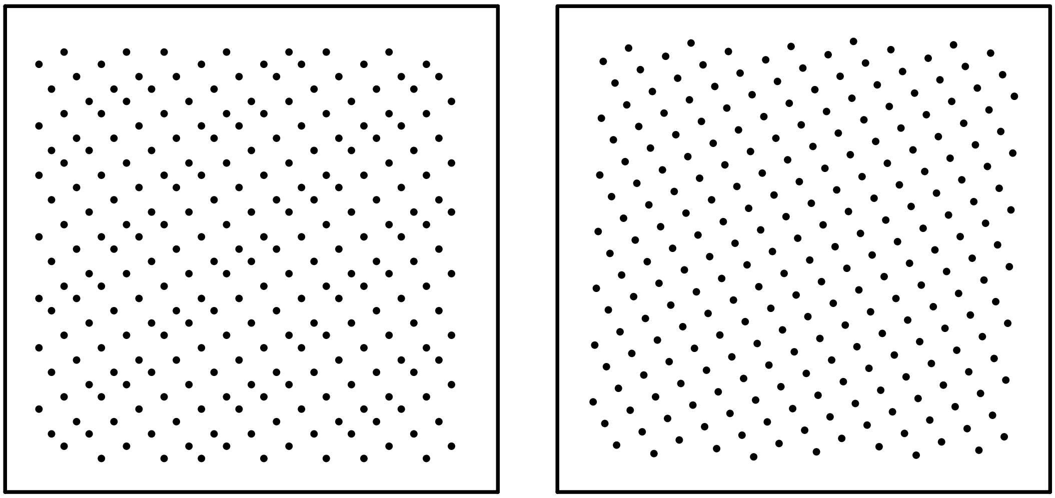
$$\frac{r_{k-1}}{d} = \frac{x_k}{z} + \frac{r_k}{dz}, \quad k = 1, 2, 3, \cdots.$$

In other words, the original sequence $\{u_k := x_k/z |\, k = 1, 2, \cdots\}$ is given the uniform estimates as follows by the introduction of a new notation $r_0 := n$ and the new sequence $\{v_k := r_k/d |\, k = 0, 1, 2, \cdots\}$,

$$0 < v_{k-1} = v_k/z + u_k, \quad k = 1, 2, \cdots.$$

This is the mechanism prepared by numbers ever since the big bang of this univers, and fishes, octopuses, dinasours and us mammals have been seeing or feeling. One notable point is that the approximation found obtains power only when the multiplier z is large and close to the modulus d. This requirement is all natural. If MC generator (d, z, n) gives good uniform and independent random numbers at all, then small z should not arise, since a small z will too frequently generate asending sequence $r_k < r_{k+1} < r_{k+2} < \cdots$. Thus $z < d^{1/2}$ will not be desirable. The uniform error $r_k/dz = v_k/z$ should be small with *good* MC generators.

Figure 1.



(End of Figure 1)

The above **Figure 1** on the right shows plots of consecutive 2-tuples of MC $(d = 251, z = 34)$ random numbers, viz.

$$\{(r_k/d, r_{k+1}/d|\ k = 0, 1, 2, \cdots, T - 1\}.$$

The right plots consecutive pairs of the original random numbers

$$\{(x_k/z, x_{k+1}/z|\ k = 1, 1, 2, \cdots, T\}.$$

We may reconstruct the original $\{u_k = x_k/d = v_{k-1} + v_k/z$ random number sequence for $k = 1, 2, \cdots$ from the MC sequence as

$$u_k = v_{k-1} + v_k/z, \quad k = 1, 2, \cdots.$$

Although we have some favor for the non-regular distributions of points in the left plot, the realistic problems prepare $z \approx 2^{52}$. The interest on the left plot might well be limited in view of the precision $1/z \approx 2^{-52}$.