Pseudo Random Number Generation Algorithms with Fibonacci Sequence

Ryuya Ohta*, Anna Kuwana, Shogo Katayama, Haruo Kobayashi *(Gunma University)*
* *T221D017@gunma-u.ac.jp*

This paper describes pseudo random number generation algorithms using Fibonacci sequence [1] for Monte Carlo simulation. Our proposed algorithm to get pseudo-random signal R_i with N-digit fractional number between 0.0 and 1.0 takes the following steps:

(1) Generate Fibonacci sequence ($F_0 = 0,\ F_1 = 1,\ F_{i+2} = F_{i+1} + F_i$).
(2) Decide an integer value N.
(3) Define R_i as the fractional part of $F_i/10^N$.

For example, for $F_i = 2584$ and $N = 2$ then its corresponding R_i is 0.84.

Notice that we can calculate simply using the following modulo calculus property:

$$\text{modulo}_{10^N}(F_{i+2}) = \text{modulo}_{10^N}(F_{i+1}) + \text{modulo}_{10^N}(F_i)$$

For digital circuit implementation, we use 2^M instead of 10^N for binary implementation suitability.

We have evaluated the proposed algorithm with 2-dimentional Monte Carlo simulation to obtain the one-fourth area of the unit circle or $\pi/4$. Fig. 1 shows the error (absolute value) between $\pi/4$ and the value obtained by Monte Carlo simulation using our proposed Fibonacci sequence pseudo-random signal. The x-coordinates of the points are generated by a proposal 6place ($N = 6$) signal, and we set the number of the fractional digits from 2 (indicated as "2place", $N = 2$) to 9 (indicated as "9place", $N = 9$) as the y-coordinates of the points. We see that as the difference of fractional digits between the x-coordinate and y-coordinate is large, and as the number of generated points increases, the error becomes smaller.

We have also compared the proposed algorithm with the linear congruential generators in [2] (indicated as "rand" in Fig. 2) and the Mersenne Twister in [3] (indicated as "MT"). We see in Fig. 2 that some proposed algorithm performances are comparable with them (for example, using 6 fractional digits for x-coordinate and 9 fractional digits for y-coordinate indicated as "X:6place Y:9place"). The proposed algorithm can be implemented with small logic circuitry and memory due to the recursion formula usage, and its random signal can be generated fast thanks to the simplicity. This would be the advantage over the other advanced pseudo-random signal generation algorithms in the viewpoint of the dedicated hardware implementation.

We will consider to apply the proposed algorithm to others than Monte Carlo simulation, and also investigate pseudo-random signal generation algorithms using Tribonacci and Tetranacci sequences. Generally, randomness, unpredictability and irreproducibility are required for random number. Our proposal cannot be used for encryption or other purposes, since N is easily predictable from some number sequences if the algorithm is known. It is a random number to be used for simulation with advanced randomness.

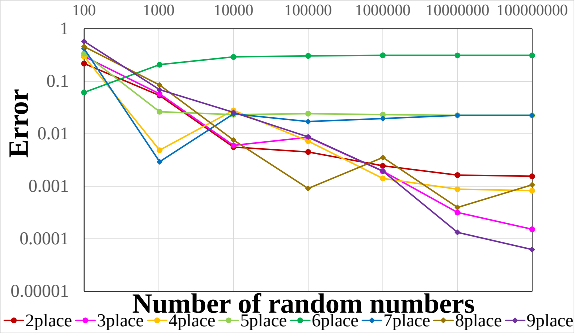

Fig. 1 Proposed algorithm performance evaluation.

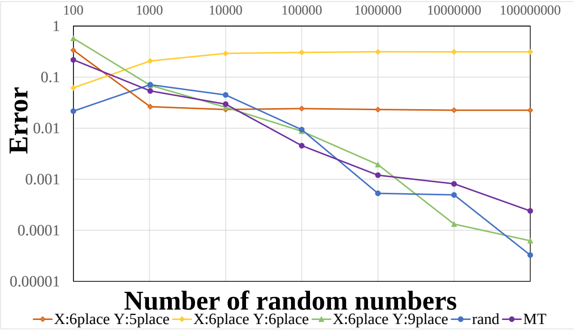

Fig. 2 Comparison with conventional algorithms.

References

[1] S. Yamamoto, et. al., "Metallic Ratio Equivalent-Time Sampling and Application to TDC Linearity Calibration", IEEE Trans. Device and Materials Reliability, Early Access (March 2022)

[2] B. Schneir, *Applied Cryptography - Protocols, Algorithms, and Source Code in C,* Wiley (2017).

[3] M. Matsumoto, et al., "Mersenne Twister: A 623-Dimensionally Equi-distributed Uniform Pseudo Random Number Generator", ACM Trans. on Modeling and Computer Simulation, Vol.8, No.1, pp.3-30, (1998).