

Examining Random Number Generators used in Stochastic Iteration Algorithms

László Szirmay-Kalos, Gábor Liktó, Anton Penzov, Ivan Dimov, Karolj Skala, and László Szécsi

Abstract— This paper investigates random number generators in stochastic iteration algorithms that require infinite uniform sequences. We take a simple model of the general transport equation and solve it with the application of a linear congruential generator, the Mersenne twister, the mother-of-all generators, and a true random number generator based on quantum effects. With this simple model we show that for reasonably contractive operators the theoretically not infinite-uniform sequences perform also well. Finally, we demonstrate the power of stochastic iteration for the solution of the light transport problem.

Index Terms—Monte Carlo methods, random number generators, stochastic iteration, transport problems.

I. INTRODUCTION

Transport problems, such as global illumination, neutron transport, etc. often lead to the solution of integral equations of the following form

$$L(x) = L^e(x) + \mathfrak{I}L(x),$$

which expresses the intensity (i.e. radiance of the light) $L(x)$ of point x as a sum of the emission $L^e(x)$ and the reflection of all point intensities that are visible from here. The reflection is expressed by an integral operator, called *transport operator*

$$\mathfrak{I}L(x) = \int K(x, y)L(y)dy,$$

where $K(x, y)$ is the kernel that expresses the coupling between points x and y . We are interested in contractive transport operators, where there exists $a < 1$ and function norm $\|L\|$ such that

$$\|\mathfrak{I}L\| < a\|L\|.$$

The contractive property is usually a consequence of energy conservation.

Solution algorithms can be classified as random walk and iteration techniques [1].

A. Random walk

Random walk algorithms are based on the Neumann series expansion of the transport equation

$$L(x) = \sum_{i=0}^{\infty} \mathfrak{I}^i L^e.$$

The terms of this series are ever increasing high-dimensional integrals that are estimated by Monte-Carlo

quadrature in order to avoid the exponential core of classical quadrature rules. A sample for the n th term is a *path* of random points x_1, x_2, \dots, x_n , which is also called a random walk. The convergence of Monte-Carlo quadrature is in the order of $O(m^{-0.5})$, where m is the number of paths.

Random walks are generated independently, thus this approach is free from error accumulation and can be easily ported to parallel machines. The obtained result will be the asymptotically correct solution of the original problem. Unfortunately, we have to pay a high price for this asymptotically correct solution. Since the paths are generated independently, the earlier results cannot be efficiently stored and reused in the computations.

A. Iteration

Iteration techniques are based on the fact that the solution of the rendering equation is the fixed point of the following iteration scheme:

$$L_m(x) = L^e(x) + \mathfrak{I}L_{m-1}(x).$$

Iteration converges with the speed of a geometric series, i.e. the error from the limiting value is in the order of $O(a^m)$ where a is the *contraction* of the transport operator. Note that iteration uses the estimate of the complete radiance function, thus it can potentially exploit coherence and reuse previous information. Since the complete function is inserted into the iteration formula, parallelization is not as trivial as for random walks, and the error introduced in each step may accumulate to a large value.

To store the radiance estimates, finite-element approaches should be used which represent the radiance function in a finite function series form:

$$L(x) = \sum_j L_j b_j(x),$$

where functions $b_j(x)$ are pre-defined basis functions and parameters L_j are scalars.

Comparing random walk and iteration we can conclude that random walk requires just one path to be stored while iteration needs very many variables, but random walk uses practically no coherence information while iteration can strongly exploit it. Iteration is slow due to the handling of the very many finite elements, while random walks are slow due to the lack of the utilization of the coherence.

Although a single iteration step requires much more computation than a single random path, the $O(a^m)$ convergence of iteration still seems to be far superior to the $O(m^{-0.5})$ convergence of random walks. However, random walk converges to the real solution while iteration to the solution of the finite-element approximation of the original problem.

Furthermore, if the transport operator is not exactly

This work was supported in part by the Bulgarian Academy of Sciences, the Hungarian Academy of Sciences, by the Croatian-Hungarian Action Fund, and by OTKA ref. No.: T042735.

evaluated, the limiting value is also distorted by the cumulative error. Thus only the initial behavior of iteration overcomes random walk.

II. STOCHASTIC ITERATION

In order to combine the advantages of iteration and random walks, the iteration scheme is randomized, which leads to the definition of *stochastic iteration* [2].

Suppose that we have a random linear operator \mathfrak{T}^* so that it gives back the effect of the original transport operator in the expected case:

$$E[\mathfrak{T}^*L] = \mathfrak{L}L.$$

During stochastic iteration a random sequence of operators $\mathfrak{T}_1^*, \mathfrak{T}_2^*, \dots, \mathfrak{T}_i^*$ is generated, which are instantiations of \mathfrak{T}^* and this sequence is used in the iteration:

$$L_m = L^e + \mathfrak{T}_m^*L_{m-1}.$$

Since in computer implementations the calculation of a random operator may invoke finite number of random number generator calls, we are particularly interested in those random operators which have the following construction scheme:

1. Random value r_m is found from a uniform distribution in a unit cube.
2. The uniformly distributed value is transformed to p_m to mimic a probability density. This density may or may not depend on function L . Making the density depend on current estimate L allows adaptive Monte Carlo approaches.
3. Using p_m a deterministic operator $\mathfrak{T}^*(p_m)$ is applied to current function L . Note that the randomness of the transport operator stems from p_m . That is why we call it the *randomization point* of the operator.

Using a sequence of random transport operators, the value $L(x)$ will also be a random variable, which does not converge but fluctuates around the real solution. However, the solution can be found by averaging the estimates of the subsequent iteration steps:

$$L(x) = \lim \frac{1}{n} \sum_{m=1}^n L_m(x).$$

This limiting value gives the solution if the following conditions hold:

- Iterated values \mathfrak{T}_m^*L are not strongly correlated.
- The variance of $\mathfrak{T}_{i+k}^* \mathfrak{T}_{i+k-1}^* \dots \mathfrak{T}_i^* L^e$ goes to zero as k increases with at least the speed of a geometric series.

Stochastic iteration can also be viewed as a single walk which uses a single sequence, and the $\mathfrak{T}_{i+k}^* \mathfrak{T}_{i+k-1}^* \dots \mathfrak{T}_i^* L^e$ terms are included in integral quadratures simultaneously for all k . It means that the randomization points should support not only single integration, but using subsequent pairs also double integration, using the subsequent triplets triple integration, etc. Sequences that support k -dimensional integrals when subsequent k -tuples are selected are called k -uniform sequences [3]. The widely used Halton or

Hammersley sequences are only 1-uniform, thus theoretically they should provide false results. This is obvious for the Hammersley sequence, in which the first coordinate is increasing. It is less obvious, but is also true for the Halton sequence. Due to its construction using radical inversion, the subsequent points in the sequence are rather far, thus the subsequent pairs will not cover region close to the diagonal of the square.

A truly random sequence is infinite-uniform, but pseudo-random sequences are usually not, despite to the fact that infinite-uniform sequences can be generated by deterministic algorithms. According to the *Franklin's theorem* [3], for irrational number $\xi > 1$, the sequence $\{\xi^n\}$, i.e. the fractional part of its n th power, is also infinite-uniform with probability one. However, as the set of such numbers is infinite, if we take a particular irrational number (e.g. π), we cannot be sure that its induced sequence will be uniform. Another problem is that irrational numbers cannot be represented in computers, and we always substitute them with finite bits, i.e. with rational numbers in base 2. The length of the approximation imposes limits on the length of the stochastic iteration.

In this paper we investigate random number generators considering their application in stochastic iteration algorithms. In fact, we test how close they are to the expectation of infinite-uniform distribution.

III. TESTING SCENARIO

In order to test random number generators, we consider a very simple scalar “transport equation”

$$L = L^e + aL$$

where the convergence speed can be controlled by value a . Similarly to integral equations, we suppose that there is no division and we can just approximately multiply by value a (in an integral equation, the multiplication by a would correspond to an integral, which cannot be inverted and can usually be only numerically evaluated, introducing some approximation error). In order to guarantee that the transport operator is a contraction, we require value a to be less than 1. We solve this equation by stochastic iteration replacing the multiplication with a by multiplying with a_m^* that can be either zero or one, with probabilities $1-a$ and a , respectively. Taking a random value r_m uniformly distributed in $[0,1]$, we set $a_m^*=1$ if $r_m < a$, and zero otherwise. The stochastic iteration scheme

$$L_m = L^e + a_m^*L_{m-1}$$

either adds the previous guess or restarts the iteration from L^e . The random numbers are generated with one of the following methods:

- `rand()` function, which is a standard linear congruential generator [4] (*rand()* in Figures 1-3) implementing the following formula:
$$r_{m+1} = (214013 \cdot r_n + 2531011) \bmod 2^{32},$$
- a true random number generator, which is based on quantum effects [5] (*true random*),
- The Mersenne twister generator [6] (Mersenne),
- The generator called the “mother-of-all-generators” [7] (*mother*).

The error curves are shown by Figures 1-3, where value a was set to 0.9, 0.99, and 0.99969, respectively. Value 0.99969 is equal to $1-10/\text{RND_MAX}$, giving the chance to the `rand()` function to generate random values with value 0.

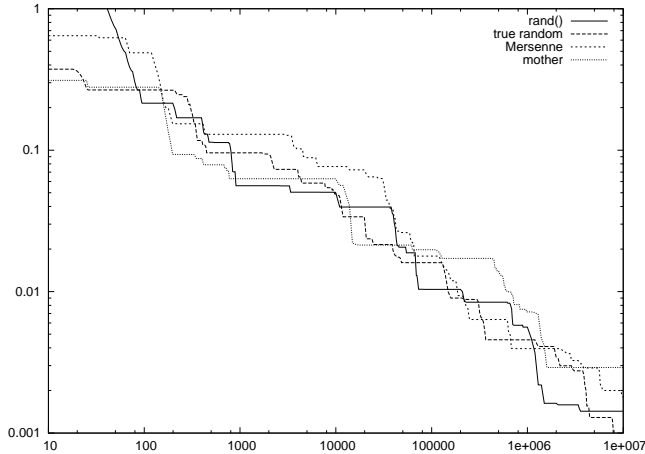


Fig. 1. Error curves of stochastic iteration when $a=0.9$. The generators behave similarly.

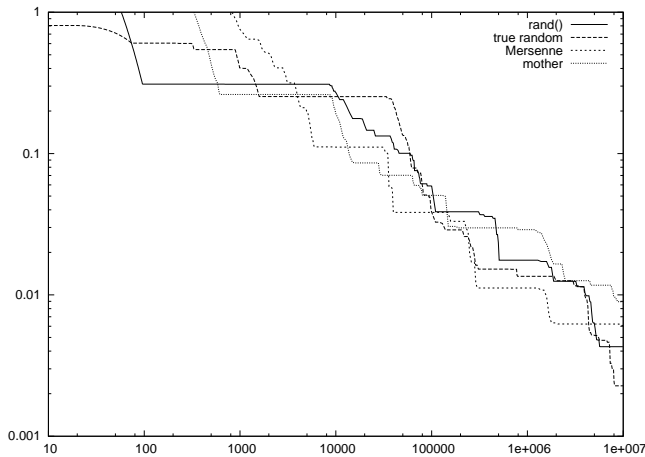


Fig. 2. Error curves of stochastic iteration when $a=0.99$. The random number generators still behave similarly.

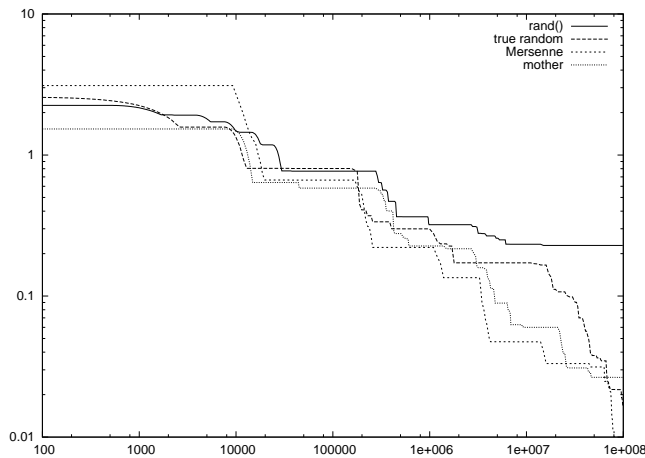


Fig. 3. Error curves of stochastic iteration when $a=0.99969$. The linear congruential generator deteriorates, but others still perform well.

Inspecting the error curves, we can note that despite the fact that pseudo-random number generators cannot produce infinite-uniform sequences, they perform fairly well for

reasonable contraction ratios. Even the linear congruential generator is quite good when contraction a is less than 0.99 and it fails only when the contraction is really close to 1. Interestingly, the Mersenne twister and the mother of all generators are good even at pathological cases. The explanation is that the contraction ratio also expresses the importance of higher order uniformity. The error of the estimation using k -uniform samples is weighted by a^k , which quickly ignores the error of the higher order terms when a is reasonably less than one.

IV. LIGHT TRANSPORT DEMONSTRATION

In order to demonstrate stochastic iteration in a real transport problem, we applied it to find the global illumination solution of the Cornell box (Figure 4). The light transport operator has been randomized in different ways.

The first algorithm selects a uniform random direction in each iteration step taking two random values, and transfers the radiance of all surface elements along this random direction. In this case, the randomization point is two-dimensional, which means that two subsequent pseudo-random numbers are needed in each iteration step. The randomization point (i.e. the direction) is independent of the current radiance function, thus the random transport operators are statistically independent. We also tried two-dimensional 1-uniform low-discrepancy series, such as the Halton (base 2 in one dimension and base 3 in the other dimension) or Hammersley (regular steps in one dimension and base 2 in the other dimension) sequences.

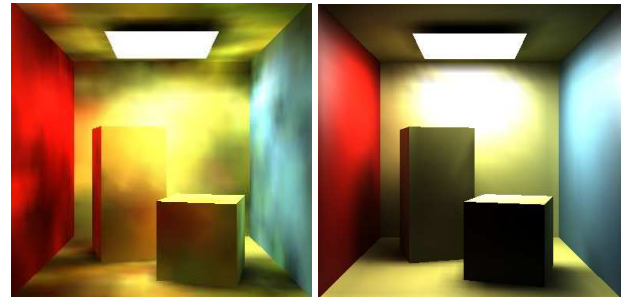


Fig. 4. The Cornell box after 100 stochastic iteration with random directions generated with the `rand()` function (left) and the converged image (right).

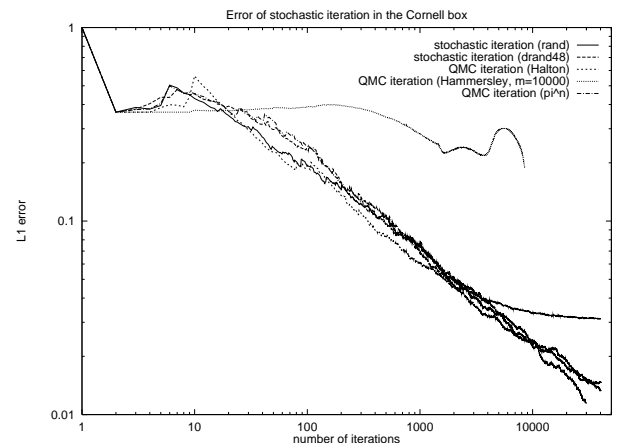


Fig. 5. Error curves of stochastic iteration working with random directions generated from random and quasi-random numbers.

Figure 5 shows the error curves with respect to different uniform sequences. We can see that the Hammersley sequence gives completely wrong result and the Halton sequence also deteriorates from the real solution. The two random generators (rand and drand48), however, performed well. The figure also included quasi-Monte Carlo sequence $\{\pi^n\}$. This is believed to be (but has not been proven to be) infinite-uniform [4].

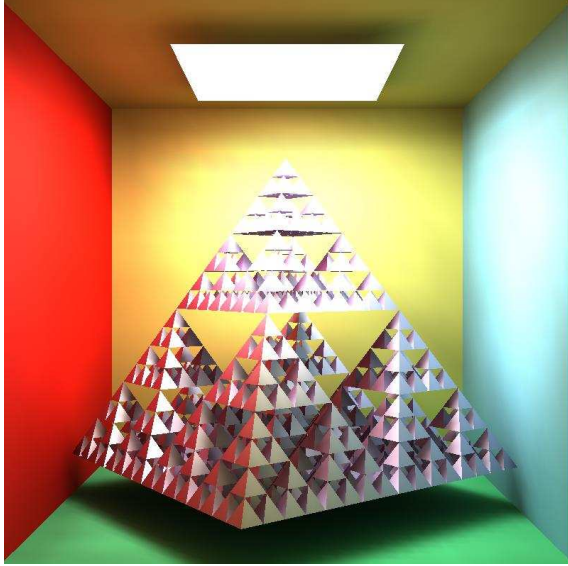


Fig. 6. The Sierpiński-set scene rendered with stochastic iteration using random directions.

Figure 6 shows another scene rendered with the same method using the rand() function.



Fig. 7. A cloud model after 10 (top), 50 (middle), and 90 (bottom) stochastic iterations.

Figure 7 depicts the convergence of the similar stochastic iteration method solving the light transport problem in participating media (a cloud), after 10, 50, and 90 iteration steps, respectively. Here, in a single iteration step, the light is transferred between neighboring particles in 128 quasi-randomly selected directions.

A. Non-independent randomization and combined strategies

In order to show that stochastic iteration works even when the randomization points are not statistically independent of each other, we also implemented another randomization strategy. In a single iteration step a single point is selected and the power of the scene is shot from this point toward all other points that are visible from here. According to the concepts of importance sampling, the shooter point is selected proportionally to its radiance, which makes its selection dependent of previous choices.

The result after 100 iterations and the converged image are shown in Figure 8.

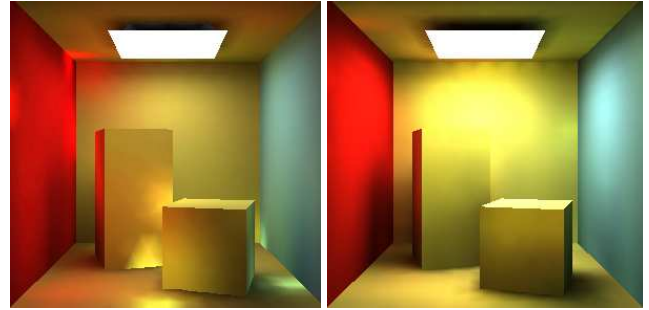


Fig. 8. The Cornell box after 100 stochastic iteration with random shooter points obtained with the rand() function (left) and the converged image (right).

V. CONCLUSIONS

This paper investigated random number generators in stochastic iteration algorithms. Theoretically, we need infinite-uniform sequences, but the simple test model and the light transport simulation demonstrated that stochastic iteration is robust enough and theoretically not infinite-uniform sequences can also be safely used. Even the simple linear congruential generators can be used if the contraction of the transport operator is reasonably less than one.

REFERENCES

- [1] Szirmay-Kalos, L.: Monte Carlo Methods in Global Illumination - Photo-realistic Rendering with Randomization, VDM Verlag Dr. Mueller e.K. 2008.
- [2] Szirmay-Kalos, L.: Stochastic Iteration for non-Diffuse Global Illumination, Computer Graphics Forum, Vol. 18, No. 3, pp. 233-244, 1999.
- [3] Knuth, D.E.: The art of computer programming. Volume 2 (Seminumerical algorithms), Addison-Wesley, Reading, Mass. USA, 1981.
- [4] http://en.wikipedia.org/wiki/Linear_congruential_generator
- [5] Radomir Stevanovic, Karolj Skala, Aneta Karaivanova, Emanouil Atanassov and Todor Gurov: A True Random Numbers Service and its Applications, MIPRO 2008, Grid and Visualization Systems, pp. 243-248, 2008.
- [6] Matsumoto, M. and T. Nishimura: Mersenne Twister: A 623-dimensionally equidistributed uniform pseudorandom generator. ACM Transactions on Modeling and Computer Simulation, 1988.
- [7] Marsaglia, G.: The Diehard Battery of Tests of Randomness. <http://www.stat.fst.edu/pub/diehard>