Pseudo Random Number Generation & Evaluation

Meifang Zheng

This project is including 3 parts: 1. Prologue; 2. Theories; 3. Evaluation.

1. Prologue—3 questions

What is a random number?

My understanding: random = equal possibility.

A number generated for or part of a set exhibiting statistical randomness.

What we mean in computer science is the numbers from a uniform distribution that lie within a specified range (typically 0 to 1).

Are the numbers generated from rand ( ) really random?

We can get random numbers easily from rand () in a C program. But think: the original code for rand () is mathematical algorithm, the number we get from it every time are actually dependent, that is the random number are predictable. Thus, numbers generated from rand () are called pseudo random number. Actually, the result of any random programming cycles, some sequence of numbers we get looks really random, because it is a segment of a long cycle.

So what we talk below are all pseudo random numbers.

How to generate pseudo-random numbers?

In Numerical Recipes in C, Chapter 7, the author provides the original code of how a computer generates a random number. However, nowadays the real world requires more algorithms rather than just one, here I will discuss 3 kinds of algorithms I learned from internet, books, and compare the different theories they are base on, and evaluate them by graphing. The basic idea of generating random numbers is use algorithms to make the last number and the next number we get more far-away, more irrelated, thus more "random".

2. Theories for 3 different kinds of random number generation

1) Generation 1—square iteration

The iterative formulas are as following:

$$Xn+1=(Xn^2/10^s)(\text{mod } 10^2s)$$

$$Rn+1=Xn+1/10^2s$$

Xn+1 is the iterator.

Rn+1 is the desired random number produced each time.

The first formula squares Xn, shifts it right by s position, then get the last 2s digits. The second formula is the result in which the final number (2s digits) is divided by pow(10,2s) which is apparently between 0 and 1.

2) Generation 2 –multiplicative iteration with modulus

The main idea of this generation is to use the below 2 iterations.

$$Xn+1=Lamda*Xn(\text{mod } M)$$

$$Rn+1=Xn/M$$

Xn+1 is the iterator.

Rn+1 is the desired random number produced each time.

Lamda and M are fixed parameters of these two formulas, and M must be a prime.

Old hands give us two good conditions so that the random number generated will not be short-time-cycle and large-relative.

1.) lamda=5^5,M=2^35-31

2.) lamda=7^5,M=2^31-1

3) Generation 3—mixed iteration with modulus

The iterations of the third generation are mixed of additive and multiplicative iterations:

$$X_{n+1}=(Lamda*X_n+Miu)\%M$$

$$R_{n+1}=X_n/M$$

It is showed that when M=2^Q, Lamda, miu and x0 as below, the generation will be long-period and with good statistical characteristics.

Lamda=2^c+1, c is the number near q/2.

Miu=(1/2+sqrt(3))/M

X0 can be any nonnegative integer.

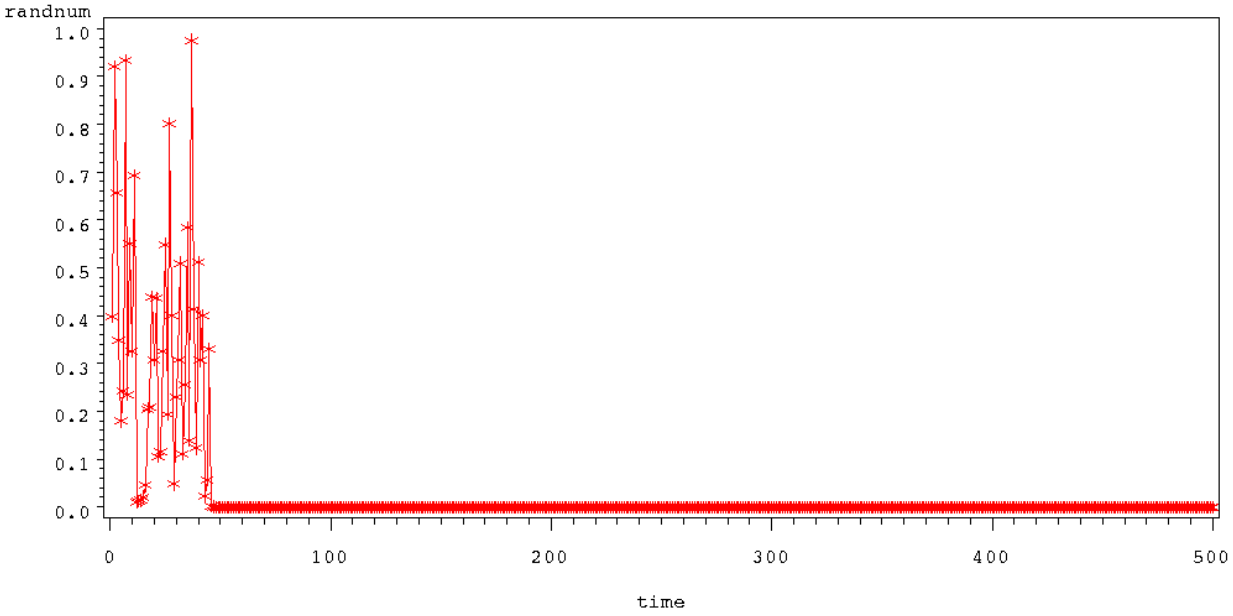
Those formulas make data far away from each other, given by people with profound mathematical knowledge for computer science, we just borrow and use, standing on the shoulders of giants.

3. Evaluation

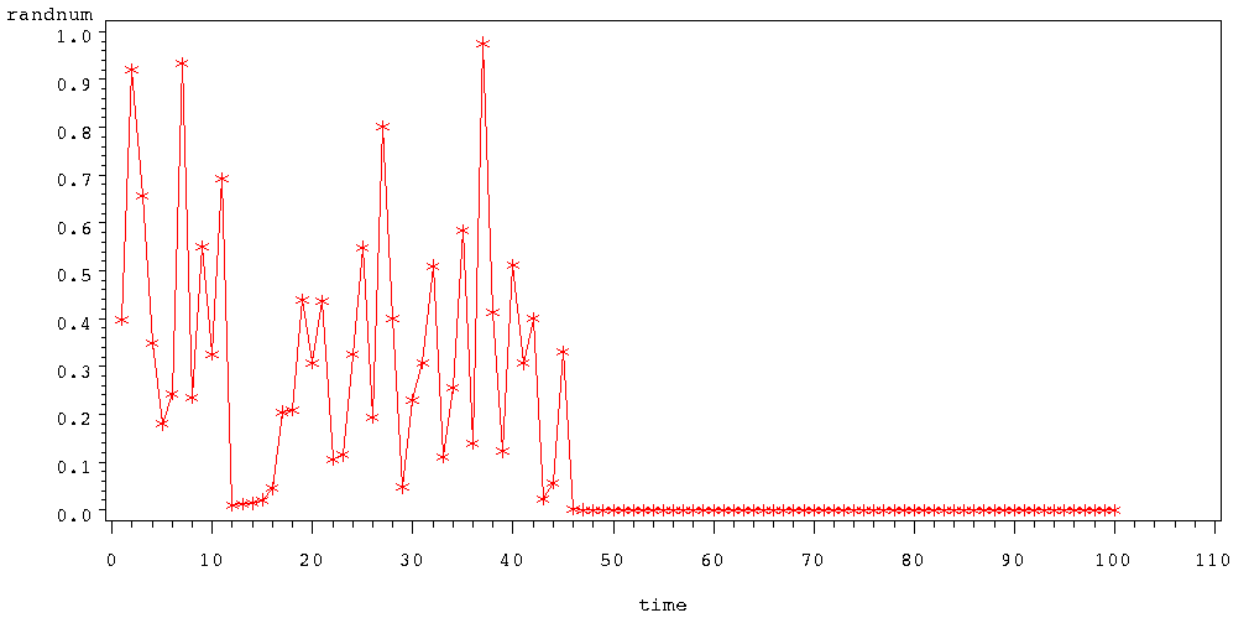
The nice random numbers we are looking for should be non-degenerate, long-time-cycle and small relative (small-relative). According to these 3 evaluation standards, I generate 500 (100 the second time for clear graphs) data from each program and use these 3 set of data to draw sequential graphs in SAS.

Results:

1) Generation 1



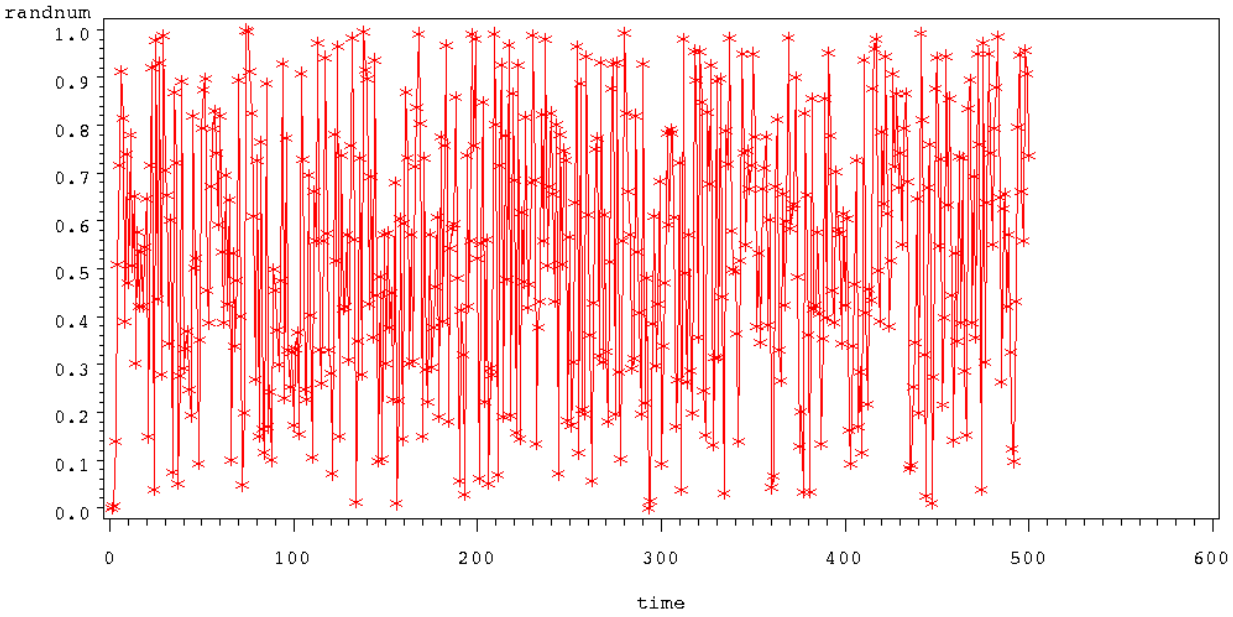
the sequential graph of 500 random numbers from generation1



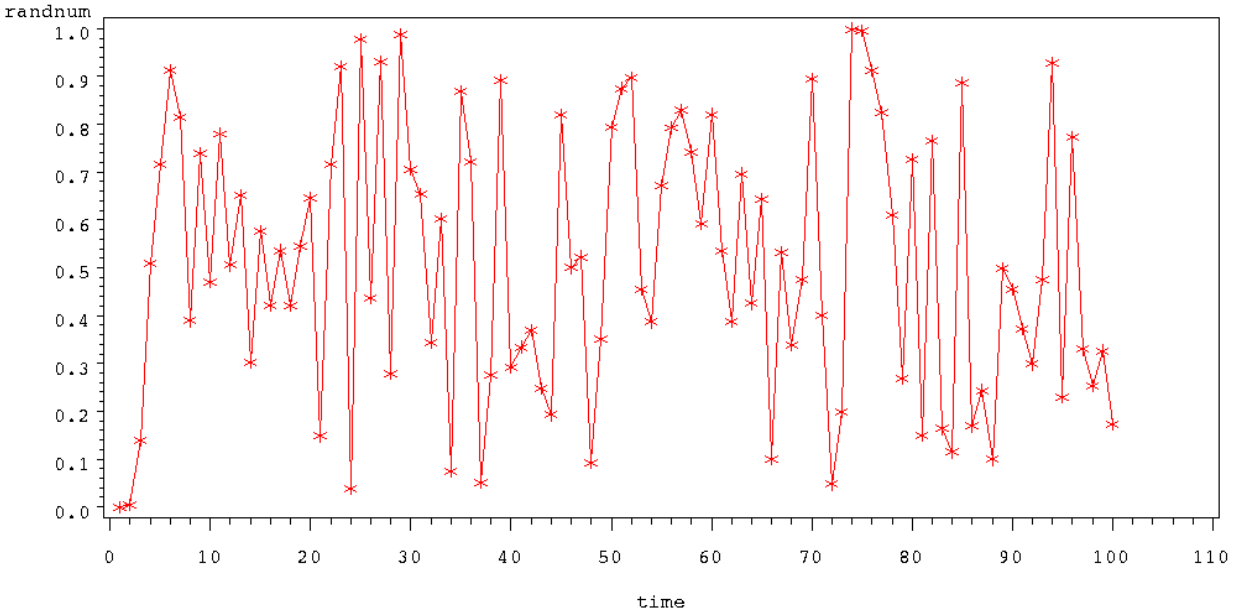
the sequential graph of 100 random numbers from generation1

We can see very clearly, after few generations the numbers go to 0, therefore, this kind of generation is degenerated.

2) Generation 2



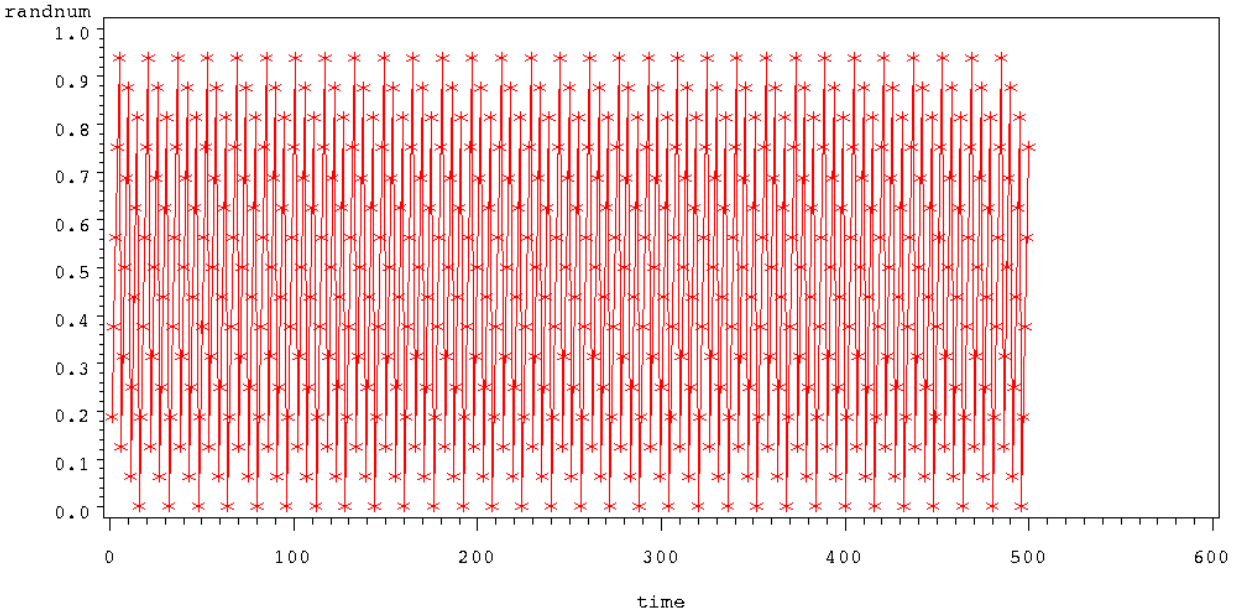
the sequential graph of 500 random numbers from generation2



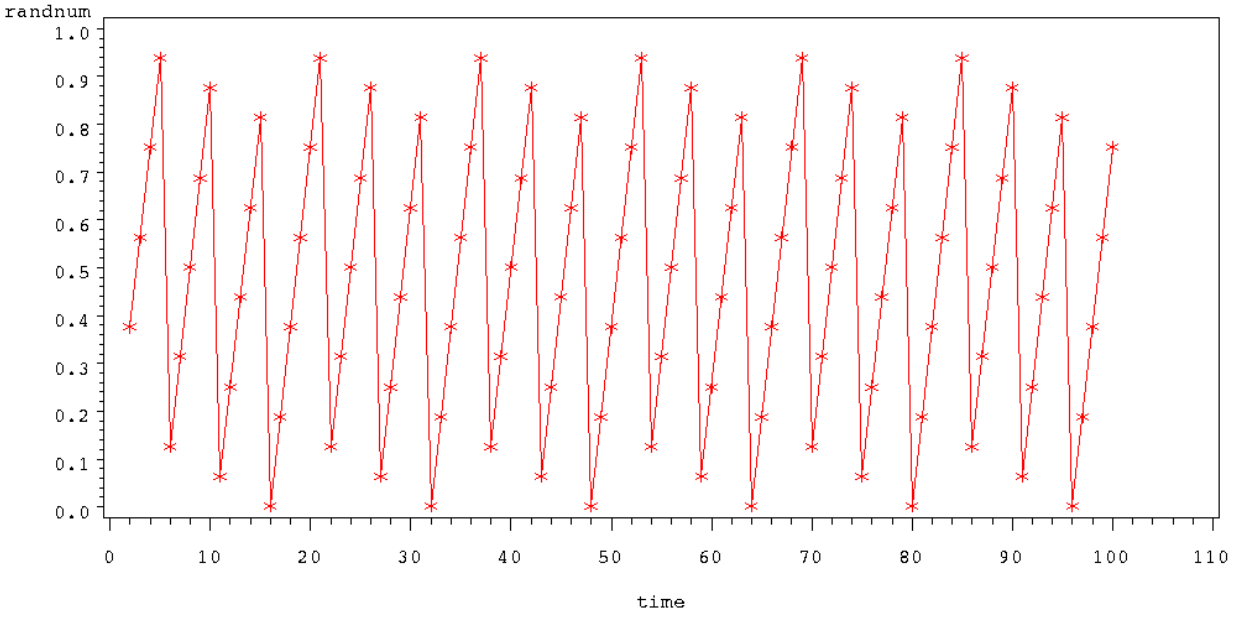
the sequential graph of 100 random numbers from generation2

Can see that the numbers here are more true of a uniform distribution between 0 and 1. However, in some area the data fluctuate largely.

3) Generation 3



the sequential graph of 500 random numbers from generation3



the sequential graph of 100 random numbers from generation3

These numbers are more like uniform distribution between 0 and 1, but we can see from the graph, the random numbers increase in each certain cycle, which is more like regulative than random.

Conclusion:

Thus, the generation 2 is more like to be a uniform random numbers between0 and 1 within these 3 models.