A Glimpse on Random Number Generation

Gursharn Kaur, Smruti Abhyankar, Shwetab Sinha, Sagnika Chakrabarty, and Kaustav Nandy

Indian Statistical Institute
M. Stat II Year

Abstract. In our project we mainly try to give some algorithms to generate pseudo random numbers using Von Neumann algorithm and K algorithm. Also we have discussed some basics of Linear Congruence Generator. Also the performance of the RNGs are studied by several types of graphs. Lastly a comparitive study is done between Von Neumann and K algorithm.

1 Introduction

Starting from a game of chance or gambling to survey sampling, from cryptography to Brownian Motion, from Statistics to Physics everywhere random numbers are very useful. But if one can generate a single uniform random number everything else follows. So the problem is to generate a single uniform random number. But how can one generate a uniform random number? If we generate a random number using some algorithm, will it be really random?

2 Random Number and Its Properties

2.1 What is a Random Number?

A random number is a number generated by a process, whose outcome is unpredictable and which can not be subsequently reproduced. Loosely speaking the word random means absence of apperent cause, planning or design. In other words random means inability to predict outcomes or to find any pattern of the outcomes.

2.2 What are the difficulties?

To generate true random numbers we need a non deterministic process. But in practice it is impossible to get. There does not exist any process which is totally unpredictable.

2.3 Seeking an alternative:

Since true random numbers are unachieveable, we seek an alternative in form of Psedo Random Number. Pseudo random numbers are outcomes of a deterministic causal process and hence deterministic.

3 Pseudo Random Number Generators

3.1 Properties of PNRGs

- PNRGs are efficient. They produce many numbers in a short time.
- They are deterministic in the sense that a given sequence of numbers can be reproduced at a later stage if the starting point in the sequence is known.
- PNRGs are periodic. That is eventually they are repeated.

3.2 Desirable properies of a good PNRG:

- The period of the sequence should be very high.
- There should not be any pattern among the digits.
- The probability of occurance of all the digits in the sequence should be the same.

4 Von Neumann Random Number:

John Von Neumann has invented his Middle square method in 1946. It is a computer based PNRG. Itarating Von Neumann's produces numbers generated by a deterministic process intended merely to imitate a random sequence

4.1 Description of the Method:

Von Neumann random Numbers are generated by the following algorithm:

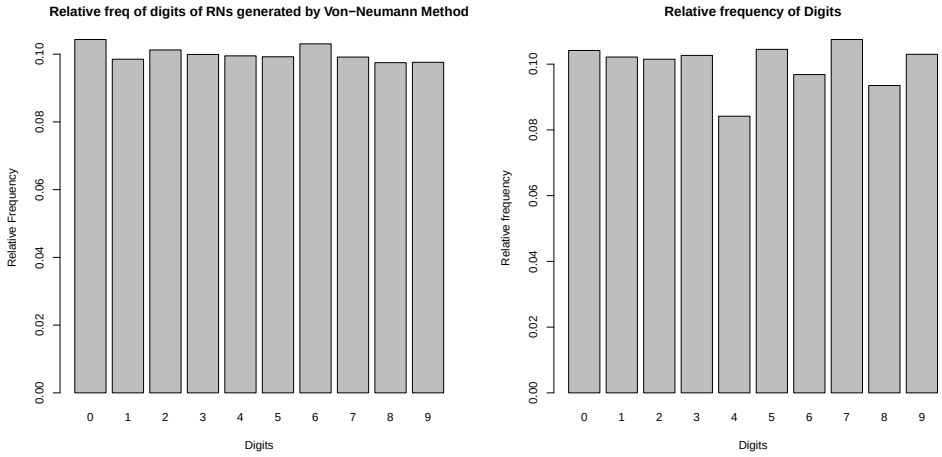
- Take a random number of n digits.
- Square it.
- Take the middle n digits.
- Use this number as the new seed and carry on the algorithm.

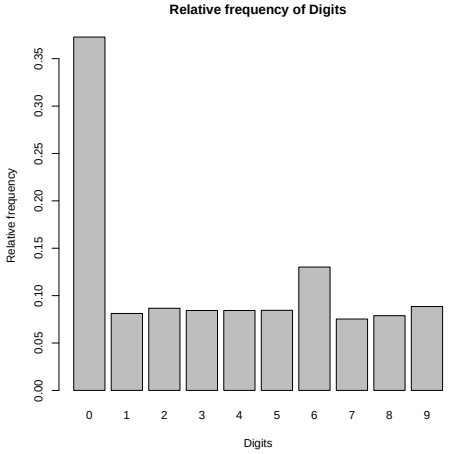
4.2 Results:

In this section we will study some properties of the generated random numbers generated, after implementig the algorithm in computer, mainly using graphs. It is implemented in both **R** and **C**. We use two methods, viz, direct method and bitwise shift operator method. A comparative study is also done here.

Frequency Density Plot:

It is obvious that in a long series of random digits all the digits should occur with equal probabilty. i.e. in long run the probability of occurance of each one of them should be almost same. Let us consider a few set of random digits and study the relative frequency plot.





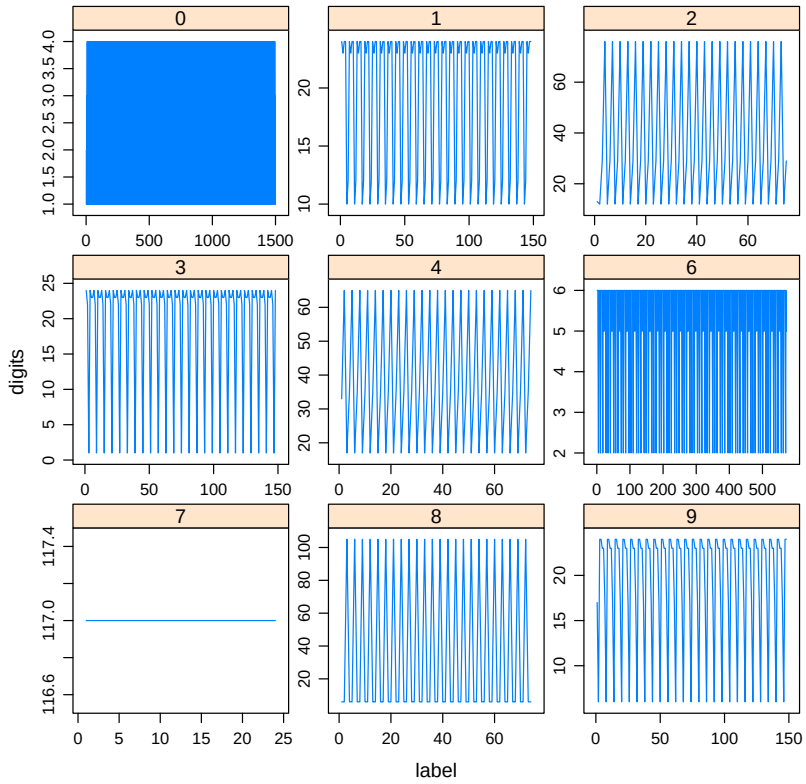
The first two graphs show that the digits in the generated random numbers occur with almost same probability. In the third graph the probability of occurance of the digit 0 is much high as compared to other digits. The choice of seed value determines the nature of these graphs.

Plot of Durations:

The following graphs are plotted in the following way:

- First seperate out the digits from the numbers generated.
- Calculate the time points at which a particular digit appears.
- Calculate the waiting time.
- Plot this.

Notice that if the plot of waiting time appears to follow any particular pattern, then we can easily predict what is going to happen. So any pattern in the waiting time plot is not at all desirable.



In this grpah we have taken our seed value as 100010. From the graph we conclude that:

- The digit 7 appears at a constant duration.
- The duration of the digits $1, 2, 3, 4, 6, 8, 9$ follows a particular pattern. But the digit 6 appears more frequently.
- The apperance of digit zero is very frequent!

So from the graph it is clear that the numbers are appearing at non random pattern. Not only the probability of occurance of the digits are different, but the time points at which they occur is totally deteministic. This is obviously not desirable.

So now we try to study for what types of seed values this happens. Also we will try to find out some 'good' seed values.

4.3 Some Observations:

Let us concentrate on the six digit seeds.

- It is a trivial observation that in Von Neumann algorithm, if at any stage we get the number 0...0, then the sequence is not going to give any number apart from 0. So we should not consider any number which has too much zeros as a seed.
- Also numbers like "$\alpha_1\alpha_2 5000$", where $\alpha_1 = 1(1)9, \alpha_2 = 0(1)9$, should not be chosen as these numbers get degenerated to a sequence of 625000 in 1 step.

– Infact we observe that any number which has three 0s at the last falls in a loop very quickly. Moreover the waiting time of their occurance is atmost 20.
– Also the numbers like $352811, 619962, 913745, 946372$ etc falls in a loop very quickly.
– Again the numbers like $711839, 962991$ do not fall in a loop quickly and the periods are also very high. We should take such numbers as seed.

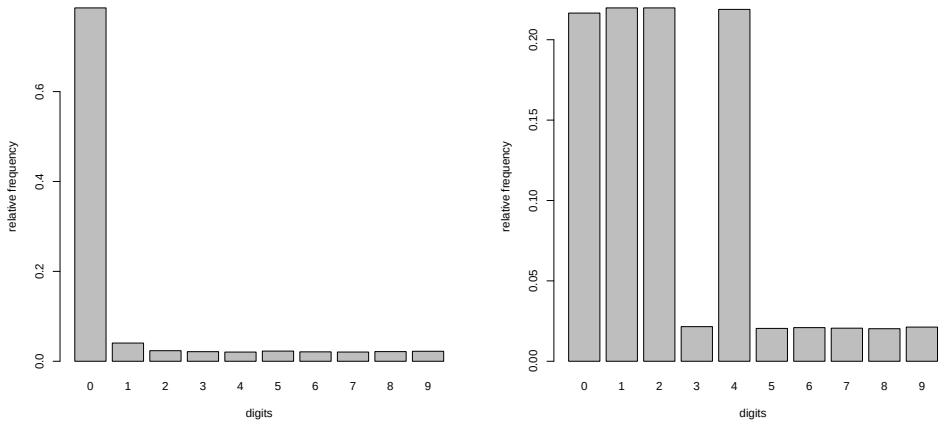
Bitwise Programming

Along with the direct method, we implement the Von Neumann method using Bitwise Shift Operator in **C**. For that our seed value will be a binary number and we wlll apply the Von Neumann Algorithm to it. The output will be given in form of decimal number as the binary numbers are always hard to interpret and can not be used readily.

Observatons:

Frequency Density Plot:

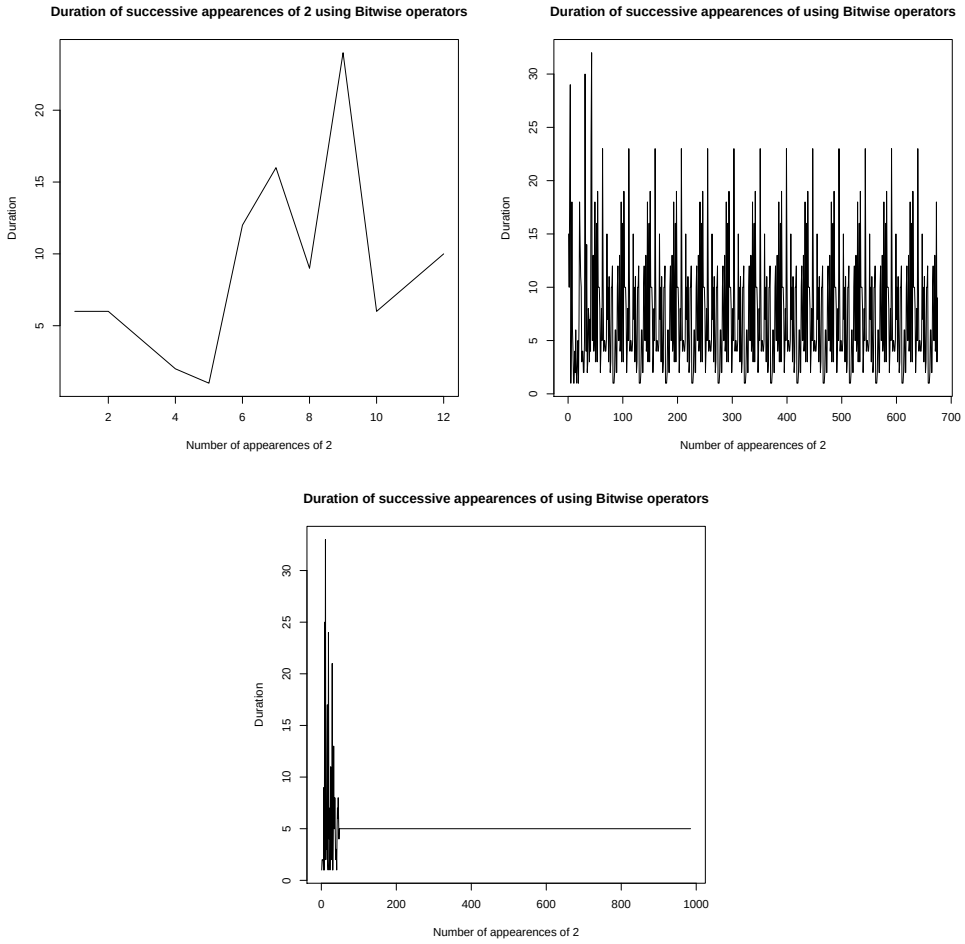
First we see the frequency density plots of the digits:



In both the plots we see that the digits are appearing with unequal probability. This is happenning in most of the cases.

Intuitive Reason: Whenever we are dealing with binary numbers there are too many 0s. We know once the sequence of random numbers falls in a loop of 0s it can not come out of it. So in Bitwise programming, almost in every case the sequence of random numbers will degenerate to 0s.

Plot of Wating time:



The above graphs show the waiting time of apperance of the digit 2 in case of three different seeds.

- First graph shows that the digit 2 is appearing only a few times.
- In second graph, we see that the waiting time of apperance of 2 falls in a loop.
- Same is for the third graph. Here eventually 2 appears in every $5th$ time.

So it is easy to see that there is a clear non-random pattern in the digits generated. Hence we can not consider the *Von Neumann-Bitwise shift operator* as a very good method for generating random numbers.

5 K-Algorithm

Given a 10-digit decimal number X, this algorithm may be used to change X to the number that should come next in a supposedly random sequence. The algorithm is as follows:

K1 [Choose Number of Iterations] Set $Y \leftarrow \lfloor X/10^9 \rfloor$, the most significant digit of X (We will execute steps K2 through K13 exactly $Y + 1$ times)

K2 [Choose Random Step] Set $Z \leftarrow \lfloor X/10^8 \rfloor \mathbf{mod} 10$, the second most significant digit of X. Go to step $K(3 + Z)$

K3 [Ensure $\geq 5 \times 10^9$] If $X < 5000000000$, set $X \leftarrow X + 5000000000$

K4 [Middle Square] Replace X by $\lfloor X^2/10^5 \rfloor \mathbf{mod} 10^{10}$

K5 [Multiply] Replace X by $(1001001001X) \mathbf{mod} 10^{10}$

K6 [Pseudo-complement] If $X < 100000000$, then set $X \leftarrow X + 9814055677$; otherwise set $X \leftarrow 10^{10} - X$

K7 [Interchange halves] Interchange the higher 5 digits of X with the higher-order five digits; that is, set $X \leftarrow 10^5(X \mathbf{mod} 10^5) + \lfloor X/10^5 \rfloor$

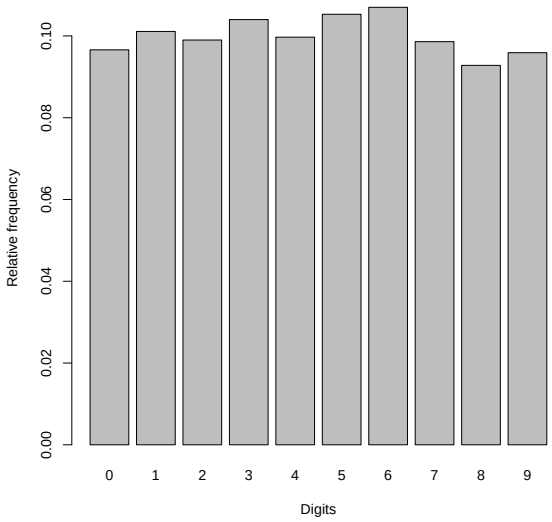
K8 [Multiply]Same as K5

K9 [Decrease Digits]Decrease each nonzero digits of the decimal representation of X by one.

5.1 Observations From Graphs:

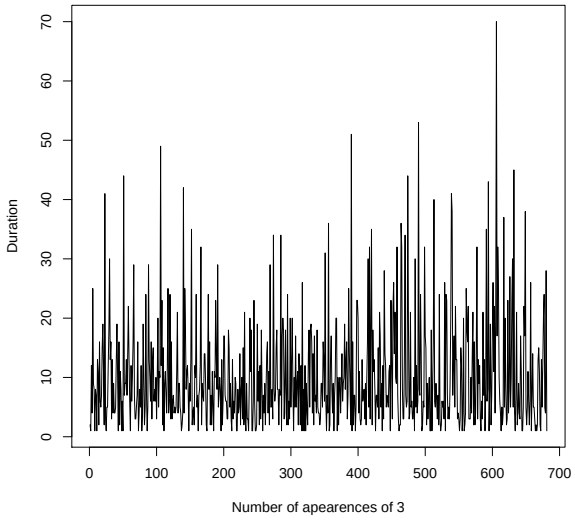
Here we have considered the same type of graphs and check the randomness of the digits.

Frequency Density Plot:



Here we see that in most of the cases the digits are appering almost with equal probability.

Waiting Time Plot:



From the graph it can be concluded that there is no particular pattern in the apperance of the digit. It is appearing almost at a random order.

So overall we can say that K-algorithm performs quite well as compared to Von Neumann algorithm.

5.2 Special Observation:

Note that the if we take the seed value as 6065038420 it returns the same number.

6 Methods:

In our project we have mainly used C as a programming tool and R to make graphs and plots. The Von Neumann (direct) method is implemented both in R and C. The Bitwise shift method is naturally implemented in C only. The K-algorithm is implemented in C. Since from C it is difficult to make plots and do further analysis, we export the output of C to R in two approaches, viz:

- By saving the output in a file and access the file from R.
- Making a shared object and using the interfacing between C and R.

6.1 Difficulties Faced:

While implementing both the algorithms, we are to save huge numbers in computer often exceeding the machine precision. In R the problem is more serious. Because of that often we have to stick to smaller seed values which is not at all a good thing to do. Still in C we are able to deal with numbers upto 20 decimal digits which is extremely essential for implementing K-algorithm.

6.2 Program Codes:

All the codes that we have used in implementing these two algorithms are given in the HTML. Please check the website:
`http://www.isid.ac.in/~deepayan/SC2010/project-submissions.html`

7 Acknowledgement:

- Knuth, The Art of Computer Programming,
- Dr. Deepayan Sarkar, ISI Delhi,
- For this project we have used:
  - GNU/Linux as the operating system
  - R and C as programming languages
  - LaTeXfor preparinging presentation and project report.

NOTE: `For any suggesstion, please mail to:` kaustav.ndyy@gmail.com