CHITKARA

Mathematical Journal of Interdisciplinary Sciences

Journal homepage: https://mjis.chitkara.edu.in/

A Better Approach to Generating Random Numbers

Nachandiya Nathan[1*] **and Samaila Andrew Mamza**[2]

[1*]*Lecturer, Department of Computer Science, Adamawa State University, Mubi, Nigeria*
[2]*Student, Department of Computer Science, Adamawa State University, Mubi, Nigeria*

[*]Email: nachandiya@yahoo.com

ARTICLE INFORMATION

Received: April 10, 2019
Revised: June 12, 2019
Accepted: July 17, 2019
Published online: September 11, 2019

Keywords:
Random Number Generator, Pseudocode Random Number, True Random Number, Mid Square Method, Linear Congruential Method, Fibonacci Series, Seed value

DOI: https://doi.org/10.15415/mjis.2019.81005

ABSTRACT

The term random number has been used by many scholars to explain the behaviour of a stochastic system. Many of such scholars with statistical or mathematical background view it as an organized set of numbers produced by a function in a numerical way in which the next number to be produced is unknown or unpredictable. This paper produced software that generates a sequence of random number and also compared the algorithm with the commonly used method of random number generator. The three most common methods selected were the Mid Square method, Fibonacci method and Linear Congruential Generator Method (LCG). The result shows that the LCG provides a more acceptable result in terms of speed, long cycle, uniformity and independenceApplications of this random numbers can be seen in Monte Carlo simulations, simulation or modelling, password generation, cryptography, and online games.

1. Introduction

Random number is a set of numbers produced by a function in a numerical pattern in which the next number to be produced is unknown or unpredictable (Shakir, Mohd and Zuraifah 2016; Kale 2013). There are basically two types of random numbers: Pseudo Random Number (PRNG) and True Random Number (TRNG). The Pseudo Random Number uses seed value for its generation Maheshwari, Gupta, Sharma and Chauhan 2014).While the True Random Number doesn't use any seed value and the number generated has no defined pattern (Katyal, Mishra and Baluni 2013).This paper tries to explore Pseudo random numbers, the different methods of generating them (Mid-Square, Fibonacci Method and Linear Congruential Method). It is also actually looks at the advantages and disadvantages of each method in terms of speed, independence, reproducibility and uniformity.

The application of Random Number can be seen in many disciplines especially in Computer Science, Mathematics and Statistics (Li 2012; Rahman, Xiao, Forte, Zhang, Shi & Tehranipoor 2014; Tong, Liu, Zhang, Xu & Wang 2015). The impactis greatly felt in optimization, modeling, simulation, numerical analysis, selection and cryptography.

2. Related Literature

One cannot talk about Random Number without acknowledging the work of John Von Neumann who first introduced the Mid-Square method of generating random number in 1946. This method brought a lot of excitement to scholars interested in Random Number Generator in those days (Li 2012). In this approach, every middle value becomes a seed value in the next iteration.

The Table 1 below shows how this method works. The seed value 76 is selected arbitrary based on the discretion of the user.

Table 1: Illustrating Mid-Square Method

S/No	N (two digits)	N^2
1.	76	5776
2.	77	5929
3.	92	8464
4.	46	2116
5.	11	121

The generation continues until the value of N^2 become less than 4 digits or zero (0). This method has a lot of disadvantages e.g. it is statistically unsatisfactory, relatively

©Author(s) 2019. This article is published with open access at https://mjis.chitkara.edu.in.

ISSN No.: 2278-9561 (Print) ISSN No.: 2278-957X (Online) Registration No. : CHAENG/2013/49583 Periodicity: Bi-Annually

slow, produce short cycle and no correlation between the initial seed and the length of the sequence of random numbers (Li 2012).

The attempts to mitigate the shortcoming or drawback of Mid-Square method give rise to the Fibonacci method. The Fibonacci method give rise to Fibonacci series. This is named after the founder of this method in 1950's, this renowned mathematician called Leonardo Pisano believes that for every integer n≥0, Fibonacci sequence $\{f_n\}$ is defined by the second order linear recurrence relation of $F_{n+2}= F_{n+1}$, F_n where $F_0 = 0$ and $F_1 = 1$. This method has been accepted in many scientific researches. The Fibonacci sequence is given by $F_n = 0,1,1,2,3,5,8,13$………n. Any change in the initial value or recurrence relation or both results to further generalized Fibonacci sequence $\{q_n\}$ (Kalman and Mena 2002; Gupta, Panwar and OSikhwal 2012; Gupta, Singh and Sikhwal 2014; M. Edson and O. Yayenie 2009).

$$q_n = a^{1-\xi(n)} b^{\xi(n) q_{n-1} + q_{n-2} (n \geq 2)}$$

with initial values $q_0 = 0$ and $q_1 = 1$, where a & b are positive real numbers and

$$\varepsilon(n) = \begin{cases} 0 \text{ if n is even} \\ 1 \text{ if n is odd} \end{cases}$$

Further research by scholar on this method and other Pseudo Random Numbers invalidate the claim that Fibonacci method is the best method of generating Random Number in which the following were considered to be the pitfalls of Fibonacci series (Rubinstein and Kroese 2011).

(a) The Initialization: selecting the initial value has been very challenging since the cycle directly or indirectly depends on the initial value or conditions. Results from the series totally depends on the initial value.

(b) Lack of theory to guides every procedure involved in the method. There has never been an established theory aside the statistical test which many researchers believe is insufficient for generalization.

(c) Despite the fact that the method can reproduce same results when tried severally with a same limited value, this method lack uniformity and is poorly distributed (Rubinstein and Kroese 2011).

To challenge the existing and trending Fibonacci method, a congruential method was established by Lehman in 1949. Even though it lacks popularity by that time, the good properties of this method, made it suitable for programmer interested in developing random number to be used for optimization, modeling and simulation experiments. In the early 1960's (Li 2012 and Tirdad 2010). This method was believed to be very easy to use and understand, its implementation is easy and fast at the same time. It integrates linear equation called the linear congruential generator (LCG). This algorithm proves a sequence of pseudo-Random numbers based on discontinues piecewise linear equation (Gurubill and Garg 2010).

The Generator is Defined by a Recurrence Relative
m, the modulus; $m > 0$.
a, the multiplier; $0 \leq a > m$.
c, the increment; $0 \leq c < m$.
X_0, the starting value; $0 \leq X_0 < m$.

Table 2: The table below shows how this method works by assuming $X_0 = 79$, $m = 100$, $a = 263$, and $c = 71$ Then.

X_1=79*263 + 71(mod 100) = 20848 (mod 100) = 48,
X_2=48*263 + 71(mod 100) = 12695 (mod 100) = 95,
X_3=95*263 + 71(mod 100) = 25056 (mod 100) = 56

The LCG has proven to be easy to implement and produced a PRNGs that have long cycle.

2.1 Benchmarking the Three PRNG Discussed Against Each Other

This section explores the advantage, disadvantage and properties of the three PRNG identified in the study.

2.1.1 Middle Square Method

Consider the example in Table 3 below, the middle square method is demonstrated using 8765as a seed value.

Table 3: Illustrating Sample of Random Number using Mid-Square Method

S/No	N (four digits)	N2
1.	8765	76825225
2.	8252	68095504
3.	0955	00912025
4.	9120	83174400
5.	1744	03041536
6.	0415	00172225
7.	1722	02965284
8.	9652	93161104
9.	1611	Etc.

The Mid Square method has some limitations, amongstare: it repeats itself, it tends to degenerate and is not uniformly distributed (Maheshwari, Gupta, Sharma and Chauhan 2014).

2.1.2 Linear Congruential Generator

Table 4 Shows how PRNGS are generated using the LCG method.
Example a = 2,175,143, seed = 3553, c = 293732, and m = 1,000,000:

Table 4: Illustrating Sample random numbers generated using Linear Congruential Method

(2,175,143 x3553 + 293732) mod 1,000,000 gives 576732				
293732	576732	859811 142890	425969	709048
992127	275206	558285 841364	124443	407522
690601	973680	275790 558869	841948	125027
408106	691185	691264 974343	257422	540501
823580	106695	389774 672853	955932	239011
522090	805169	088248 371327	654406	937485
220564	503643	786722 069801	379880	662408
945487	747832	030911 313990	597069	880148
163263	446342	729421 012500	295579	578658
861737	144816	427895 710974	994053	277132
560211	843290	126369 409448	692527	975606
208179	491258	774337 057416	340495	623574
906611	189690	176979 460058	743137	026216
309295	592374	875453 158532	158611	441690
694769	977848	260563 543735	826814	435008

2.1.3 Fibonacci Generator

Consider the example in Table 3 below, the Fibonacci method which begin with either 0 or 1

Table 5: Illustrating Sample of Random Number generated using the Fibonacci methods

First Term = 0
Second term = 1
Third Term = First+ Second = 0+1 =1
Fourth term = Second + Third =1+1 =2
Fifth Term = Third + Fourth = 2+1 =3
Sixth Term = Fifth + Sixth = 3+5 =8
Eighth Term = Sixth + Seventh = 5+8 =13 ...etc

F_0	1
F_1	1
F_2	2
F_3	3
F_4	5
F_5	8
F_6	13
F_7	21
F_8	34
F_9	55
F_{10}	89
F_{11}	144
F_{12}	233
F_{13}	377

More of the numbers can be generated using different initial values. The Figure 1 below shows how Pascal Triangles can be used to generates Fibonacci Random numbers. (Gary 2012).

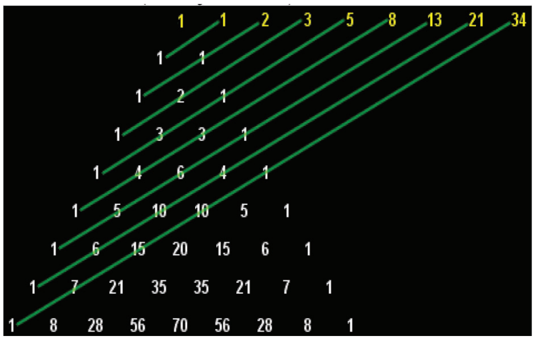


Figure 1: Pascal's Triangle

2.2 Summary of Findings

Table 6: Comparing the three PRNG discussed against each other

Mid-Square Method	**Fibonacci Method**	**Linear Congruential Method**
Mid-Square Method produce a short cycle that make PRNGs to degenerate and quite slow than Linear Congruential Method	Fibonacci Method provide a PRNGs that are not uniformly distributed	Linear Congruential Method provide a PRNGs that are uniformly distributed, independent and provide long cycle

Following the above tables, the following comparison were made and we decided that LCG provides an enhanced result in comparison to other PRNGs.

3.1 System Analysis

3.1.1 Feasibility Study

According to findings, there are random number generators available on the internet and there is also an increasing demand for efficient and reliable random number generators that can generate a sequence of random numbers for statistical and simulation purposes respectively. The fact finding techniques used in order to study the existing system includes: observation, interview and research.

Observation: This technique was used to obtain the functionalities of the existing system. This takes into consideration the use of some random number generators and how they present their outputs, coupled with the number of digits they generate as a single random number.

Interview: Using this method, university students from some of the departments in ADSU were interviewed. Precisely students from Biological Sciences Department, Department of Accounting, Mathematics, Computer Science Departments and Economics. They welcomed the idea of having an automatic random number generator for experiment purposes. Their response was captured in a questionnaire.

Secondary Sources: Various academic journals were reviewed by the researcher the outcome revealed the need for an efficient and reliable random number generator which could be used for general purposes. This however, prompted the researcher to fill the discovered gap.

Problems of the existing methods for generating random numbers.

i The existing system is non reproducibility: this is a situation where the system cannot give the same results even when all the initial conditions are met.

ii Further, the existing system tend to de-generate: this means the system tends to give a static or zero result after certain time or circle of generating the numbers.

iii The existing system lack speed; very slow in generating the numbers.example the manual method of generating Radom numbers of tossing a dice, shuffling of cards, etc. the existing method is slow more especially when a large numbers of the random numbers are needed.

3.2 Proposed system

Random number is a set of numbers produced by a function in a numerical pattern in which the next number to be produced is unknown or unpredictablewhich can be used precisely for simulation and mathematical purposes (Shakir, Mohd and Zuraifah 2016; Kale 2013). This system is a web-based application which runs on a browser to produce a random number. The generated random numbers however can be printed,

3.3 System Design and Development

3.3.1 System flowchart

The Figure 2 below is a flow chart illustrating how the new system is to be defined. The system captures values as inputs, generate numbers and give results.

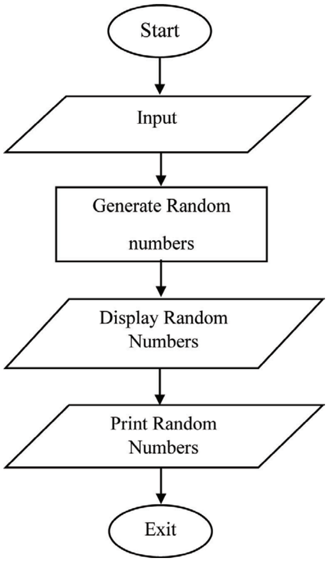


Figure 2: System flowchart

3.3.2 Technologies for Development

Learning Management System (LMS) is an online web based platform for managing learning. Therefore, the following technologies are being used for the development of the system the technologies used in developing this system includes:

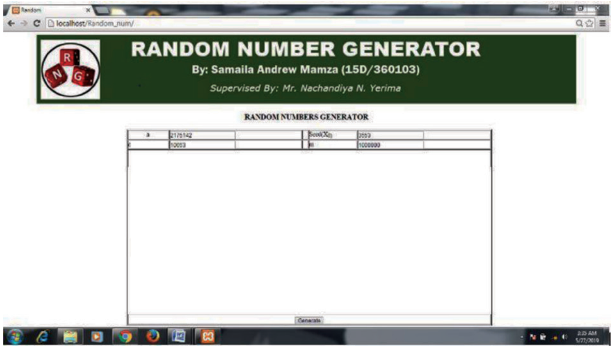
i PHP stands for Php: Hypertext Preprocessor which is a scripting language for server side backend development.

ii HTML, CSS, Ajax, and JavaScript for frontend

iii Bootstrap: is a framework used for responsive frontend web design.

4. Results

The Figure 3 below shows the system Homepage. Which is the first page after launching the system



RANDOM NUMBER GENERATOR

By: Samaila Andrew Mamza (15D/360103)

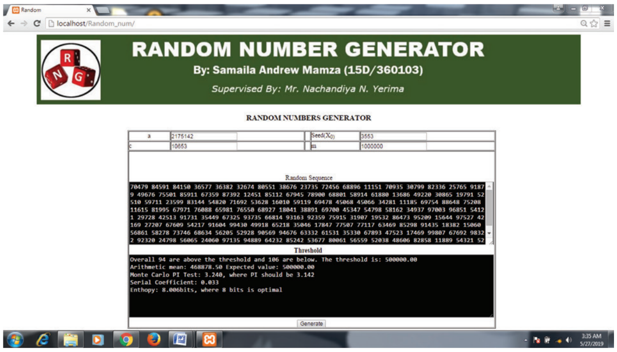
Supervised By: Mr. Nachandiya N. Yerima

RANDOM NUMBERS GENERATOR

a	2175142	Seed(X_0)	3553
c	10653	m	1000000

Figure 3: Showing the System Home page

The Figure 4 below shows a sequence of generated random numbers when the button "generate" was clicked for the first time.



RANDOM NUMBER GENERATOR

By: Samaila Andrew Mamza (15D/360103)

Supervised By: Mr. Nachandiya N. Yerima

RANDOM NUMBERS GENERATOR

a	2175142	Seed(X_0)	3553
c	10653	m	1000000

Random Sequence

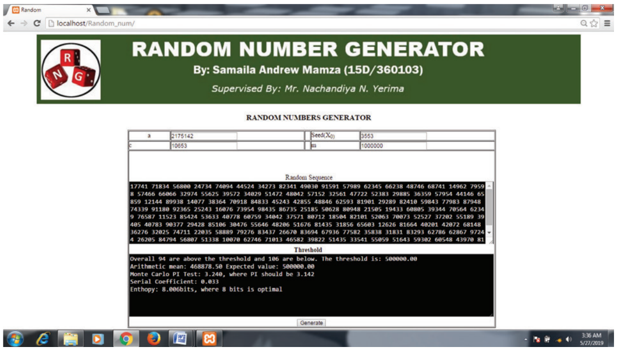
```
70479 84591 84150 36577 36382 32674 80551 38676 23735 72456 68896 11151 70935 30799 82336 25765 9187
9 49676 75501 85911 67359 87392 12451 85112 67945 78900 68801 58914 61880 13686 49220 30865 19791 52
510 59711 23599 83144 54820 71692 53628 16010 59119 69478 45068 45066 34281 11185 69754 88648 75208
11615 81995 67971 76088 65981 76550 68927 18041 38891 69700 45347 54798 58162 34937 97003 96851 5412
1 29728 42513 91731 35449 67325 93735 66814 93163 92359 75915 31907 19532 86473 95209 15644 97527 42
169 27207 67609 54217 91604 99430 49918 65218 35046 17847 77507 77117 63469 85298 91435 18382 15060
56861 58278 73746 68634 56205 52928 90569 94676 63332 61531 35330 67893 47523 17469 99807 67692 9832
2 92320 24798 56065 24060 97135 94889 64232 85242 53677 80061 56559 52038 48606 82858 11889 54321 52
```

Threshold

```
Overall 94 are above the threshold and 106 are below. The threshold is: 500000.00
Arithmetic mean: 468878.50 Expected value: 500000.00
Monte Carlo PI Test: 3.240, where PI should be 3.142
Serial Coefficient: 0.033
Enthopy: 8.006bits, where 8 bits is optimal
```

Figure 4: Showing Generated Random Numbers 1

The Figure 5 below shows a sequence of generated random numbers when the button "generate" was clicked for the second time.



RANDOM NUMBER GENERATOR

By: Samaila Andrew Mamza (15D/360103)

Supervised By: Mr. Nachandiya N. Yerima

RANDOM NUMBERS GENERATOR

a	2175142	Seed(X_0)	3553
c	10653	m	1000000

Random Sequence

```
17741 71834 56800 24734 74094 44524 34273 82341 49030 91591 57989 62345 66238 48746 68741 14962 7959
8 57466 66066 32974 55625 39572 34029 51472 48042 57152 32561 47722 52383 29885 36359 57954 44146 65
859 12144 89938 14077 38364 70918 84833 45243 42855 48846 62593 81901 29289 82410 59843 77983 87948
74339 91180 92365 25243 16076 73954 98435 86735 25185 50628 80948 21505 19433 60805 39344 70564 6234
9 76587 11523 85424 53633 40778 60759 34042 37571 80712 18504 82101 52063 70073 52527 37202 55189 39
405 40783 90377 29428 85106 30476 55646 48206 51676 81435 31856 65603 12626 81664 40201 42072 68148
36276 32025 74711 22035 58889 79276 83437 26670 83694 67936 77582 35838 31831 83293 62786 62867 9724
4 26205 84794 56807 51338 10070 62746 71013 46582 39822 51435 33541 55059 51643 59302 60548 43970 81
```

Threshold

```
Overall 94 are above the threshold and 106 are below. The threshold is: 500000.00
Arithmetic mean: 468878.50 Expected value: 500000.00
Monte Carlo PI Test: 3.240, where PI should be 3.142
Serial Coefficient: 0.033
Enthopy: 8.006bits, where 8 bits is optimal
```

Figure 5: Showing Generated Random Numbers 2

4.1 Discussions of Results of Implementation

This system as illustrated above, shows the different implementation interfaces or results of the system. The first figure Figure 3 shows the system home page where the button to be clicked when one wants to generate random numbers appears along with the date the random numbers were generated. The second figure Figure 4 and the third figure Figure 5 shows the random numbers generated as a result of clicking on the button. The random numbers can therefore be printed as hard copy for simulation or statistical experiments respectively.

Whenever one needs a different sequence of random numbers, one needs to click on the button "generate" and the random numbers will be generated.

5. Summary

Random number is a set of numbers produced by a function in a numerical pattern in which the next number to be produced is unknown or unpredictable.

The main aim of this paper is to develop a software that can generate a sequence of random numberwhich can be used precisely for simulation, cryptography, games and mathematical purposes. php and Mysql are used in designing the software. The system was tested and found to be more efficient and effective in generating random numbers.

Conclusion

Before looking at this research work, the improvement, efficiency and quick service that can be derived from the system has over weighted the manual method of generating random numbers such as shuffling of cards and rolling a dies. The program is user friendly and can be used by any authorized person. With the evolution of computer technology, its importance can be seen in every aspect of human endeavors.

Recommendations

1. University students especially those in Department of computer science, should adopt this software that can generate a random number for the process of getting the result especially during experiments involving the use of random numbers.
2. Programmers should adopt this software and generate random number to be used for cryptography, optimization, modelling and simulation.
3. Casino and gambling Gaming Industry should adopt this softwareto produce an output of their games

Further Studies

More researches be done on other methods of generating random numbers. Any results obtained should be benchmarked against this application to measure reproducibility, uniformity and independence. Also, criteria for selecting seed value should be developed so that it will not be at the discretion of the user.

References

Bonde, V. and Kale, D.: Design and Implementation of a Random Number Generator on FPGA, International Journal of Science and Research (IJSR) 4(5), 203-208 (2015).

Edson, M. and Yayenie, O.: A new generalization of Fibonacci Sequence and Extended Bine t's formula, Integer **9,** 639–654 (2009). doi.org/10.1515/INTEG.2009.051.2009.

Gary, M., Pascal's triangles, (2012)

Gupta, Y. K., Panwar, Y. K. and Sikhwal, O.: Generalized Fibonacci Sequences, Theoretical Mathematics and Applications 2(2), 115-124 (2012).

Gupta, Y. K., Singh, M. and Sikhwal, O.: Generalized Fibonacci - Like Sequence Associated with Fibonacci and Lucas Sequences, Turkish Journal of Analysis and Number Theory 2(6), 233-238 (2014). doi.org/10.12691/tjant-2-6-9.2014.

Gurubill, P. R., Garg, D.: Better technique of random number generation, National Conference on Emerging Trends in Engineering and Sciences (NCETES- 2010)

Kalman, D. and Mena, R.: The Fibonacci numbers - Exposed, The Mathematical Magazine **2,** (2002).

Katyal, Mishra and Baluni: True Random Number Generator using Fish Tank, Image International Journal of Computer Applications 78(16), 38–40 (2013). doi.org/10.5120/13609-1419.

Li, K: Performance analysis and evaluation of random walk algorithms on wireless networks. International Journal of Foundations of Computer Science 23(04), 779–802 (2012). doi.org/10.1142/S0129054112400369.

Maheshwari, R., Gupta, S., Sharma, V. and Chauhan, V.: Pseudo Random Number Generator, (2014).

Rahman, M. T., Xiao, K., Forte, D., Zhang, X., Shi, J., & Tehranipoor, M. TI-TRNG: Technology Independent True Random Number Generator. Paperpresented at the Proceedings of the The 51st Annual Design Automation Conference on Design Automation Conference, (2014). doi.org/10.1145/2593069.2593236.2014.

Rubinstein, R. Y. and Kroese, D. P.: Simulation and the Monte Carlo method, 707, John Wiley & Sons, 2011.

Shakir, A., Mohd, M. and Zuraifah, N.: Implementation of the Binary Random Number Generator Using the Knight Tour Problem, (2016). doi.org/10.5539/mas.v10n4p35.

Tong, X., Liu, Y., Zhang, M., Xu, H., & Wang, Z.: An Image Encryption Scheme Based on Hyperchaotic Rabinovich and Exponential Chaos Maps. Entropy, 17(1), 181–196 (2015). doi.org/10.3390/e17010181.

Tirdad, K. Ryerson University "Developing pseudo random number generator based on neural networks and neuro fuzzy systems" (2010).

Mathematical Journal of Interdisciplinary Sciences

Chitkara University, Saraswati Kendra, SCO 160-161,
Sector 9-C, Chandigarh, 160009, India

Volume 8, Issue 1 **September 2019** **ISSN 2278-9561**

Copyright: [© 2019 Nachandiya Nathan et al.] This is an Open Access article published in Mathematical Journal of Interdisciplinary Sciences by Chitkara University Publications. It is published with a Creative Commons Attribution- CC-BY 4.0 International License. This license permits unrestricted use, distribution, and reproduction in any medium, provided the original author and source are credited.