INFLUENCE OF RANDOM NUMBER GENERATORS IN AIR POLLUTION MODEL SAMPLING

MSc. Eng. Vrglevski G.[1], Prof. D-r. Eng. Mitreski K.[1], Ass. Prof. D-r. Eng. Naumoski A.[1], Prof. D-r. Eng. Zajkova S.G.[2]
Faculty of Computer Science and Engineering, University Ss. Cyril and Methodius in Skopje[1], Laboratory of Eco-informatics Republic of Macedonia
Faculty of Electrical Engineering and Information Technologies, University Ss. Cyril and Methodius in Skopje[2], Republic of Macedonia
E-mail: {[1]goce.vrglevski,kosta.mitreski, andreja.naumoski}@finki.ukim.mk,
[2]szajkova@feit.ukim.edu.mk

***Abstract*:** The research covers the usage of the random numbers in Monte Carlo simulations for air pollution models. Two new random number generators are developed; their strengths are compared with the existing random number generators. The results in this paper showed that the two newly developed random number generators achieved better results on a basis of failed test, however it extended the time for generating random numbers. In future we plan to use the newly developed random generators for filling the missing values in the measurements.

Keywords: *Monte Carlo, Random Numbers, Ecology, Modeling, Random Generators*

1. INTRODUCTION

The random numbers are widely used with the Monte Carlo simulation methods. In this research we will check the impact of the choice of the random number generator to the results of the air pollution simulations using the Monte Carlo methods.

The Monte Carlo methods can be described as a class of computational algorithms based on repeated random sampling to obtain numerical results. Their essential idea is using randomness to solve any problem having a probabilistic interpretation. They are used in the air pollution field to obtain numerical values about pollutions or in the sampling process where input data is generated for the simulations.

The research paper is organized as follows: in Section 1 we give an introduction to the problematic in the research, while in section 2 we present some examples for Monte Carlo simulations used in the air pollution. In Section 3 of the paper, we present the development of two algorithms for generating random numbers and compare them with the most popular random number generators. Section 4, presents the evaluation results from the computed air pollution error due to choosing different algorithm for the sampling process, while section 5 concludes the paper and outlines out future research directions.

2. AIR POLITION RESEARCHES THAT USE MONTE CARLO METHODS FOR AIR POLUTION

Monte Carlo methods are widely used in the air pollutions models. They are used for sampling the data that is used in the air pollution models, for Monte Carlo air pollution simulations or other uses.

2.1 Taking samples with Monte carlo method

Data is collected for the air pollution model. The countries or the areas are huge so an efficient way for collecting data is needed. Monte Carlo method provides a way for taking samples in random period of time in order to collect enough data for the model. This is used to make scale down version of a whole system and make some conclusions about that.

The sampled data can be used to compute various scenarios like the air pollution in the countries in case they produce all of their products that they consume. This is especially important for big producers and consumers to determine stake holders of the air pollution. Like the impact of the Chinese export to the to the air pollution in USA, shown on Fig. 1 [1].

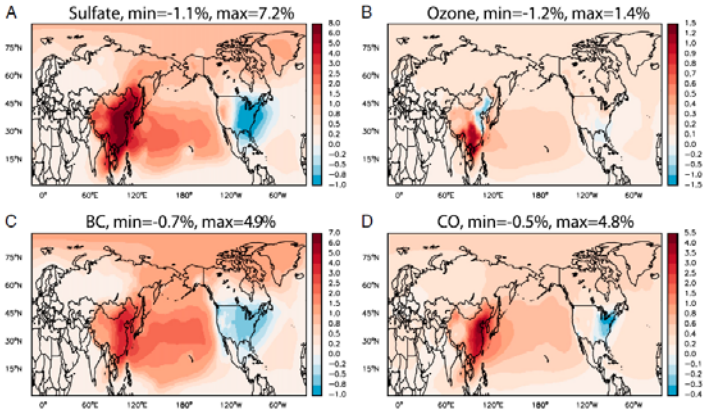

Fig. 1. Chinese export of products VS home production of products air pollution in USA [1].

2.2 Monte Carlo method for simulation of air polution

The Monte Carlo method can be used to make air pollution simulations making probability models based on small amount of data that is available for the air pollution in the region.

The method can be used to compute the air pollution in the kitchen due to the use of wooden stove for cooking. The simulation takes very small amount of data like the thermal efficiency of the most used stoves and the distribution for the kitchen sizes. The simulation explores randomly a lot of possible placements of the stove in the kitchen, the size of the kitchen, the air currents. One such research is performed for India, where it is found that huge percentage of the people are exposed to a lot of air pollutions from the stoves that they use for cooking presented in Fig. 2 [2].

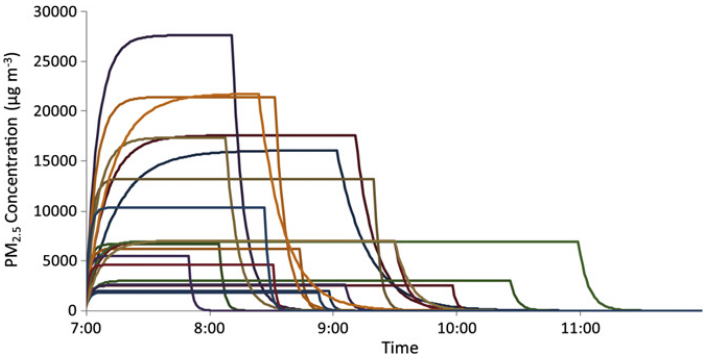

Fig. 2. Air polution due to using the G3300 stove for cooking [2].

3. DEVELOPMENT AND TESTING OF RANDOM NUMBER GENERATORS

3.1 Development of white noice random number generator

The algorithm is based on physics concepts that are widely known. The choices of concepts are those that have randomness. The following physics concepts are used in the algorithm:

- Entropy is the disorder in the system that is according to the second law of thermodynamic. Bigger the entropy means that the system is in more random state. [3].
- The waves in the quant mechanics are probabilistic distribution of the position of the particle location. A particle can be found in multiple positions according to the explanations for the double slit experiment, which describe the interference pattern [4].

The source of randomness that is used in this algorithm is the memory addresses that are provided by the operating system to the running program. When new variable is created then memory address is allocated by the operating system. The memory address alone is not enough to create something random, so swarm of addresses are used for that purpose in order to increase the entropy in the system. The acquired memory addresses form the operating system along side with the memory content in their location is feed into wave simulation. The wave function is generated using simple hash function. The salt is used as different wave function type. The amplitude can be defined as module of the wave function with the swarm functions number.

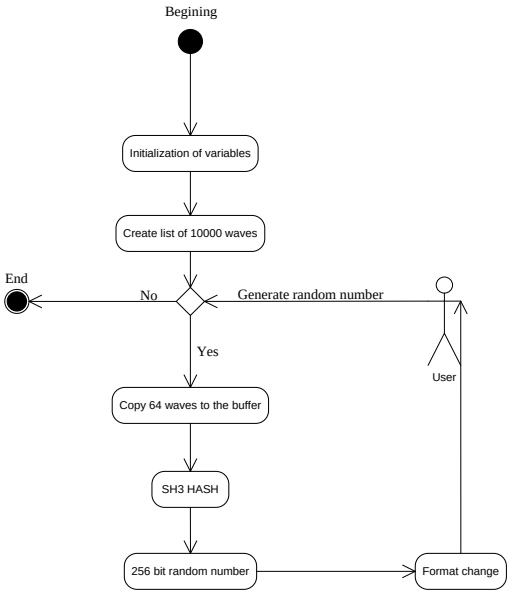

Fig. 3. Activity diagram for the white noice simulation random number generator

64 functions are copied to a buffer; the buffer is hashed using SHA3 algorithm and produced 256-bit random number. The most important part is the way wave functions are chosen to be copied. The first function that is copied is the 17th function from the swarm, the amplitude of the 17th function is determined (the number of functions in the swarm is 10000, the amplitude is number between 0 and 9999) and used to find the next wave function that will be copied into the buffer. After one wave function is copied into the buffer then it is replaced by new wave function using new memory location. The process continues until 64 functions are copied into the buffer. The process starts all over when new 256-bit random number is required, with one change that the start location is not the 17th wave function but the one determined by the amplitude of the last wave function in the previous cycle.

This random number generator showed on Fig. 3 can be considered as hardware random number generator because the random numbers cannot be precompiled due to the use of memory addresses that are not known at the beginning of the program execution. The difference with the other hardware random generators is that the algorithm has strong software influence when generating the random numbers.

3.2 Development of genetic random nuber generator

The algorithm is based on biology concepts that are widely known. The choices of concepts are those that have randomness like the genetic. The following biology concepts are used in the algorithm:

- Telomeres are segments of DNA attached to the end of the chromosomes. The role in the cell life it to prevent uncontrolled number of copies, the telomeres are shortened every time the cell is copied. When the telomeres are too short part of the chromosome will not be copied, so the result cell will not be functional [5].
- Meiosis represents generation of reproductive cells, the DNA from the mother and father are recombined in order to produce new unique organism [6].
- Mutation is cell information error, which can happen when the genetic code is changed due to radiation, chemicals or spontaneous [7].

The meiosis random number generator algorithm utilizes all those concepts. The seed of the algorithm are 256 bits numbers for the mother and father; the seed need to be random and generated with hardware random number generators. The seed is the genetic material for the mother and the father. The genetic material is recombined, which mean parts of the mother and father material are exchanged. The new combination is hashed with SHA3 algorithm and 256-bit random number is created. The telomeres are added to the buffer, hashed and used to mutate the genetic material (randomly flip bits in the genetic material). The process is repeated for every new random number that needs to be generated. More details about the algorithm in the activity diagram in Fig. 4. The meiosis random number generator is pseudo random number generator because knowing the seed and the algorithm can be determined the random numbers.

3.3 Comparing the strength of the random number generators

The library *TestU01* contains a lot of tests that can be used to determine the validity of the random number generators. There are a lot of groups of tests that can be used. The test group called *Crash* contains 144 tests for the random number generators, it takes a lot of time to execute due to huge number of tests that need to be performed to the random number generator [8].

The *gsl* library is scientific library that contain a lot of popular implementation of random number generators [9].

List of nine random number generators is created including the two newly created and tested using the *Crash* test from the *TestU01*, the popular random number generators are provided by *gsl*, the two new random number generators are called *noise* and *meiosis*.

The result of the random number generator tests is that the first group of random number generators failed less than 5% of the tests (*ranlxd2*, *noise*, *meiosis*, *gfsr4*, *mt19937* and *taus2*), the second group of random number generators failed more than 50% of the tests (*mrg*, *cmrg* and *rand*).

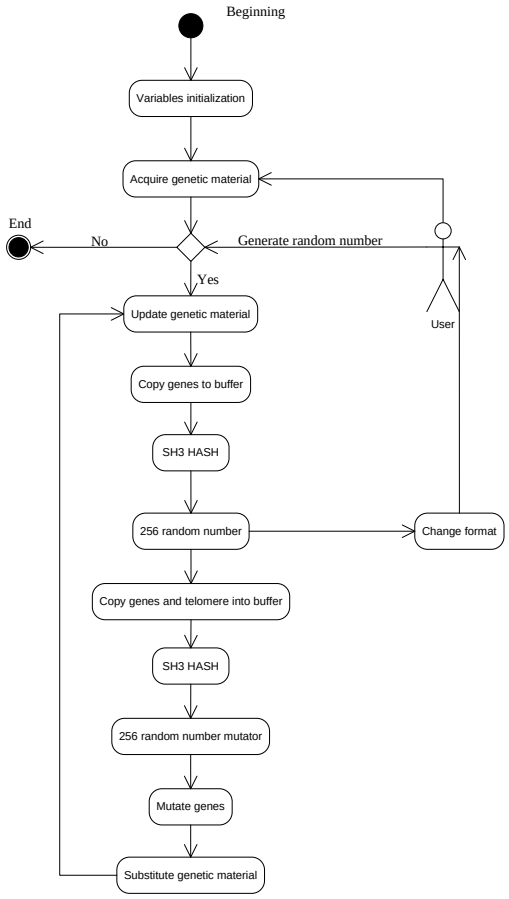

Fig. 4. Activity diagram for the meosis random number generator

TABLE 1. RANDOM NUMBER GENERATORS COMPARISON

RNG name	Failed tests	Failed tests percentage	Used time
ranlxd2	0	0.00%	03:16:36
noise	1	0,69%	29:45:02
meiosis	1	0,69%	06:03:32
gfsr4	2	1,38%	00:50:35
mt19937	3	2,08%	00:53:14
taus2	7	4,86%	00:48:08
mrg	76	52,77%	01:01:30
cmrg	76	52,77%	01:12:35
rand	85	59,02%	00:57:44

According to the test the two new created random number generators scored good with just one test failed and are part of the first group. The second group is clearly the one that has worst score as random number generators then the first group. So, the most obvious conclusion from those tests is that using the first group of random number generators has more advantage then using the second group of random number generators due to more randomness in the first group.

Checking whether or not the first group of random number generators have more advantage then the second group of random number generators is a big task due to a lot of checks needed for every different context that the random number generators are used. Some clues can be received if we check it in specific context like the sampling algorithms in the air pollution simulations. The initial hypothesis is that the better random number generator will produce less error when used with sampled data with Monte Carlo method in air pollution simulation. We will see whether or not this is true in the next chapter.

4. RANDOM NUMBER GENERATORS INPACTS IN THE SAMPLING PROCESS FOR AIR POLLUTION

4.1 Sampling data for air pollution simulation

The sampling mean creating subset of data that can provide information for the whole set of data. The whole set of data cannot be acquired for a lot of processes especially for the air pollution. The air pollution is changing second by second at every spot in the ground, so there is no device that can measure it accurate for every spot and time. The sampling for the simulations can be produced using Monte Carlo method, which is measurement of the pollution in random time. Sufficiently height number of measurements will reduce the error.

The sampling process will require gate in order to determine whether or not to make measurement and the specific moment. The simplest gate is to use random number to determine it. For example: random numbers between 0 and 10 are generated, if the number is bugger then 5 make measurement in that time. This is the simplest version of gate.

More complicated gates are wall and membrane.

- Wall represent gate that will let certain percentage of measurements to pass only if the random numbers are sufficiently high. For example: if the wall maximum size is 100 and the wall size is 90, then only the random numbers bigger than 90 will skip the wall and produce measurements. In this case the expected percentage of successful leap of the wall is 10% of the time.
- Membrane represent gate that contains opening that lets some measurement pass through it. For example: if the membrane has 10 opened holes from 100 possible (random numbers can be assigned to the wholes like 33, 22, 78, 45, 82, 63, 49, 87, 37 and 19), then only the numbers generated that are contained in the list will pass the membrane and produce measurements. In this case the expected percentage of successful pass through the membrane is 10% if the time.

4.2 Sampling error using air pollution data for Skopje

The air pollution data that is used is for Skopje from the period of 2007 to 2013. It contains 500000 records for pollutants PM_{10}, O_3, CO, NO_2, PM_{25} and SO_2. The data can be used to compute statistics like the average value of the pollutants in Skopje.

The Monte Carlo simulation will sample part of the data and compute the difference between the real values and the computed values of the average pollutant. Two gates are used to determine if the sampled data will be used for the computations (wall and membrane) or not, there are six pollutants, there are nine random number algorithms, the simulations will be performed five times per use case, and so the total number of simulations that will be computed is 6*2*9*5=540. The simulations had executed for several hours.

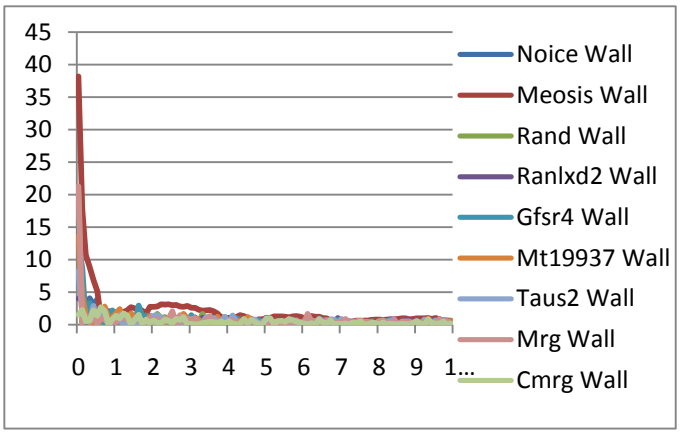

Fig. 5. Uncertainty when computing average value of PM_{10} when using wall.

The computation of the uncertainty for the values of the other pollutant follow the same pattern so only the values for PM10 are displayed at the graph. The X axis represents the percentage of data that is used for the computation of the average value for the pollutant. The Y axis is the absolute values in percentage of the difference between the computed values and the simulated values with the Monte Carlo method. If the percentage of data used for computation is very small, then the uncertainty is big. For example: 50 random measurements from the set of 500000 is 0.0001% of the data, the uncertainty for computation of the average values for the pollutant is about 40%, that is expected because 50 measurements are very few.

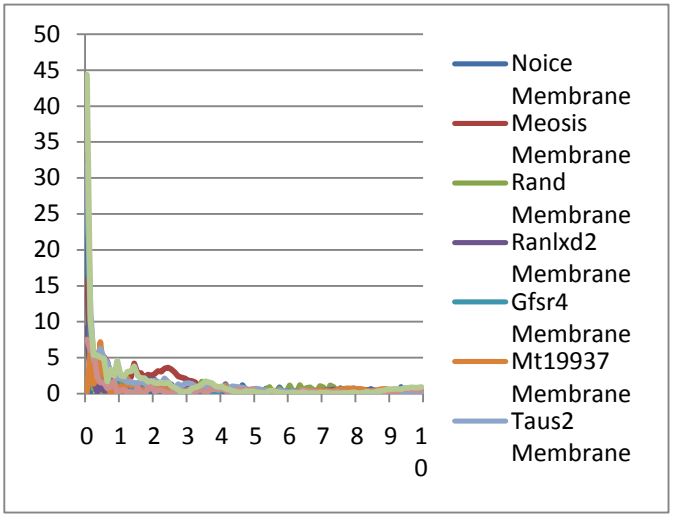

Fig. 6. Uncertainty when computing average value of PM_{10} when using membrane.

Increasing the number of measurements that are included into the computation to 4% result in uncertainty of just 2%, which depending on the application might be good precision. When the number of measurements that are included with the sampling are increased to higher percentage, the values of the uncertainty is reduced even more. The uncertainty of the computation is exponentially reduced depending on the percentage of data included into the simulation.

Fig. 5 displays the results sampled using the wall algorithm for sampling, Fig. 6 displays the results sampled using the membrane algorithm for sampling. When comparing them there can be concluded that the sampling doesn't depend on the chosen algorithm for creating the gate.

The sampling is performed using nine random number generator algorithms, the strength of the algorithms was different according to the testing of the algorithms, but the result of the Monte Carlo simulation for every one of them is very similar, so we can conclude that the choice of random number generator didn't impact the results of the computation in the simulation. This is expected from the point of few of sampling, since the measurements are slow change variables. There were two groups, the first group fails less than 5% of the tests, and the second group fails more than 50% of the tests for randomness.

5. CONCLUSION

The algorithm choice for random numbers in Monte Carlo simulations for air pollution doesn't impact the computed results as long as have decent performances. This is opposite from what was expected according to the algorithms tests for random number generators. This is due to the fact that the pollution parameters measurement doesn't change too much quickly over short period of time. The results from our research in future will focus on implementing random number generators in filling the gaps of missing values of the measurements as well as improving the Monte Carlo method with faster random generators.

6. LITERATURE

[1] Lin, J., Pan, D., Davis, S.J., Zhang, Q., He, K., Wang, C., Streets, D.G., Wuebbles, D.J. and Guan, D., 2014. China's international trade and air pollution in the United States. Proceedings of the National Academy of Sciences, 111(5), pp.1736-1741.

[2] Johnson, M., Lam, N., Brant, S., Gray, C. and Pennise, D., 2011. Modeling indoor air pollution from cookstove emissions in developing countries using a Monte Carlo single-box model. Atmospheric Environment, 45(19), pp.3237-3243.

[3] Planck, M., 2013. Treatise on thermodynamics. Courier Corporation

[4] Sakurai, J.J. and Commins, E.D., 1995. Modern quantum mechanics, revised edition.

[5] American Federation for Aging Research, Telomeres and telomerase, https://www.afar.org/docs/migrated/111121_INFOAGING_GUIDE_TELOMERESFR.pdf, 2011

[6] Campbell Biology, Cell division: mitosis and meiosis, https://www.tcd.ie/Biology_Teaching_Centre/assets/pdf/by1101/jfby1101/jfby1101-lecture3-2013-bw.pdf, 2013

[7] Presciuttini S., "Induced Mutations", http://statgen.dps.unipi.it/courses_file/genetica/28-Induced%20mutations.pdf, 2009

[8] Canada Research Chair in Stochastic Simulation and Optimization, TestU01, http://simul.iro.umontreal.ca/testu01/tu01.html, 2009

[9] Free Software Foundation, "Random number generator algorithms," https://www.gnu.org/software/gsl/manual/html_node/Random-number-generator-algorithms.html#Random-number-generator-algorithms, 2016