3D Nuclear Reactor Simulator: An Application of Monte Carlo Simulation Techniques

Hussain Hyder Ali Khowaja, Mohd. Noman Saeed, Aleem Khalid Alvi
Computer Engineering Dept., Sir Syed University of Engineering & Technology, Karachi, Pakistan

***Abstract**. The experimentation of nuclear reactor is perhaps one of the most expensive and may produce hazardous effects if not controlled properly but with the advancements of 3D graphics and high speed processing; the computational simulation gives almost same result that could be performed on physical experimentation. Once the software is written, countless experiments to examine the factors affecting a reactors performance can be carried out at little cost and with no dangerous side effects. Our simulation software is developed using C# & CsGL (An Open GL wrapper for C#). It provides a 3D environment of a Nuclear Reactor controlled by input parameters interacted with output result simulations. Its purpose is to helps students & engineers in education for understanding energy generation process. The implementation of Monte Carlo method for this simulation involves following lives of a large group of neutrons in the reactor, with each neutron taking it's own random walk from release; either absorption, exiting reactor, or causing further fission. From this it is possible to work out how enrichment level and fuel:moderator ratio affect the reactors performance. It also allows to vary control rods, f:m ratios, pump speed, along with its inner & outer components views.*

***Keywords**: 3D Graphics, Monte Carlo techniques, Reactor, Simulation*

1. Introduction

The Chain reaction is initiated, when a slow neutron strikes a heavy nucleus as illustrated by the equation given below

${}_{92}U^{235} + {}_{0}n^{1} \rightarrow {}_{56}Ba^{144} + {}_{36}Kr^{90} + 3{}_{0}n^{1}$ +Energy (200Mev)

These three neutrons are proportionally absorbed by control rods positions for reaction level & the heat energy produces steam that passes through turbine to rotor converting mechanical energy into electrical energy. A user can provide its specifications & our software gives resultant information depending about various inputs entered. The calculated amount & positions of the rods are examined and then output results are generated.

1.1. Monte Carlo Techniques

The Monte Carlo are statistical simulation techniques are defined in general terms to be any method which utilizes sequences of random numbers to perform the simulation. It has been used for centuries, but only in the past several decades gained the status of a full-fledged numerical method capable of addressing the most complex applications. The name "Monte Carlo" was coined [1] by Metropolis (interest in poker card game on the value of hands) during the World War II because of the similarity of statistical simulation to games of chance. It is now used routinely in many diverse fields, such as the simulation of complex physical phenomena such as simulation of radiation transport in the earth's atmosphere, sub-nuclear processes in high energy physics experiments or even for the players in Bingo game. Monte Carlo simulation methods may be contrasted to conventional numerical discretization methods, which typically are applied to ordinary or partial differential equations described as underlying physical or mathematical.

1.2. Monte Carlo Applications

In the applications of Monte Carlo, the physical process is simulated directly, and there is no need to even write down the differential equations which describe the behavior of the system and the only requirement is that the physical (or mathematical) system be described by probability density functions (pdf's). In many practical applications, one can predict the statistical error (the "variance") in this average result, and hence an estimate of the number of Monte Carlo trials that are needed to achieve a given error. Assuming that the evolution of the physical system can be described it can be proceed by sampling from these pdf's, which necessitates a fast and effective way to generate random numbers uniformly distributed on the interval [0,1).

2. Primary Components for Monte Carlo Simulation Algorithm

The following are primary components of [1] Monte Carlo simulation algorithm

2.1. Probability distribution functions

The physical (or mathematical) system must be described by a set of pdf's (probability distribution functions).

2.2. Random number generator

A source of random numbers uniformly distributed on the unit interval must be available.

2.3. Sampling rule

A prescription for sampling from the specified pdf's, assuming the availability of random numbers on the unit interval, must be given.

2.4. Scoring (or tallying)

The outcomes must be accumulated into overall tallies or scores for the quantities of interest.

2.5. Error estimation

As estimate of the statistical error (variance) as a function of the number of trials and other quantities must be determined.

2.6. Parallelization and Vectorization

The algorithms to allow Monte Carlo methods to be implemented efficiently on advanced computer architectures.

2.7. Variance reduction techniques

These are methods for reducing the variance in the estimated solution to reduce the computational time for Monte Carlo simulation

3. Theory

The neutrons move within a medium, which may be gaseous, liquid or solid, they collide with the nuclei of the atoms in the medium. In a collision, a neutrons may be absorbed by the nucleus or it may be scattered, elastically or inelastically [2]. Absorption may result in loss of the neutron, e.g., by radiative capture or in an increase in the number of neutrons. In figure 1 shows the consequence of scattering of neutrons and during its propagation they may have change in the position, energy and direction of motion.

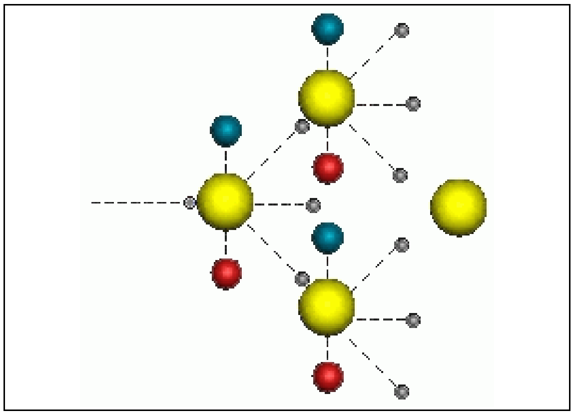


Figure 1: Scattering of Neutrons in Fission

After a neutron has been emitted, it is slowed down to thermal energies (~0.025 eV) by a series of elastic collisions with moderator atoms (ie carbon). As the reactor consists of various types of atoms, for each collision we consider we must make a decision [3] as to whether the neutron hits carbon, U^{235} or U^{238}. The process of making these decisions (scatters or absorbs) is based on branching probabilities, where a uniform deviate is used to decide which of all the possible outcomes occurs for this collision. If, for example, the decision to be made, concerns whether a hypothetical coin comes up heads or tails, then the branching probability situation is as described in figure 2. As the probability of each is 1/2, we simulate throwing a coin by creating a uniformly distributed random number and seeing which range it lies in. If the number is between 0 and 0.5, we count a head, and if the number lies between 0.5 and 1, we count a tail. For a many coins simulation, this will give the result that the probability of a head or a tail is 1/2. Although this is a pointlessly simple example, this method is very useful for making decisions in more complex systems, & all the decisions made for the reactor simulator work in this way.

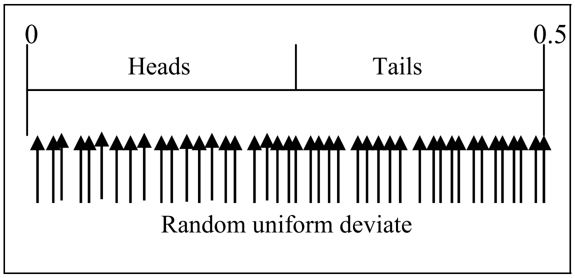


Figure 2: The technique of branching probabilities

The actual probabilities we need to decide between come from the 'cross-sections' of each possibility, where a cross-section represents the effective area that an atoms places in the path of a particle for the appropriate interaction to take place [3]. This usually has the symbol σ, for example; σ_a for absorption, σ_s for scattering and so on. To carry out the calculation, we require the probability per unit length Σ, given by:

$$\Sigma = N\sigma$$

where N is the number density of the particular type of atoms in question, which can be calculated from:

$$N = \frac{0.6025\times 10^{27} \times \text{Density}}{\text{Atomic weight}}$$

This allows us to calculate the mean free path for the system by finding the total probability of any kind of collision occurring, and simply inverting it (i:e dividing 1 by the result). For example, if the total probability of a collision is given by:

$$\Sigma_{total} = N_a\sigma_a + N_s\sigma_s$$

then the mean free path, usually denoted by an λ, is:

$$\frac{1}{\Sigma_{total}}$$

Therefore, the calculation can proceed under the assumption that the neutron travels an average distance λ between each collision, and that we only have to make any decisions about the neutrons random walk at these points. To do this we need to know the branching probabilities of scattering and absorption for a collision, and what the probabilities of the atom being carbon, U^{235} or U^{238} are. These are simple to calculate, for example the probability of scattering occurring is:

$$P(\text{scat}) = \frac{\Sigma_s}{\Sigma_{\text{total}}}$$

As mentioned before, the principle mechanism for neutron loss is through resonance absorption by U^{238}. This is slightly more complicated than before, because while most of the cross-sections involved are independent of reactor configuration, the absorption cross-section for U^{238} depends strongly upon it (more specifically, it depends on $\Sigma s / N_{238}$). This dependence is described by the table of (Σ_s / N_{238}) against σ_a in, and linear interpolation is used to approximate a value for σ_a from this data. From all of this we can calculate all the probabilities we need. The whole point of the slowing down phase is for the neutrons energy to be decreased through collisions, and to be able to do this we need to know how the energy is affected by a collision. The energy loss in an elastic collision is related to the angle of deflection θ, through the equation:

$$\frac{E_1}{E_0} = \frac{1 + 2A\cos\theta + A^2}{(1 + A)^2}$$

where E_0 and E_1 are the energies before and after collision, and A is the mass number of the scattering atom. It takes around 100 collisions with carbon (low mass number, high energy exchange) to slow a neutron down to thermal energies.

4. 3D Geometry and computational aspects:

Let a rectangular volume element dv with dimensions dx, dy, dz be located at a point where coordinates x, y and z as shown in figure 3.

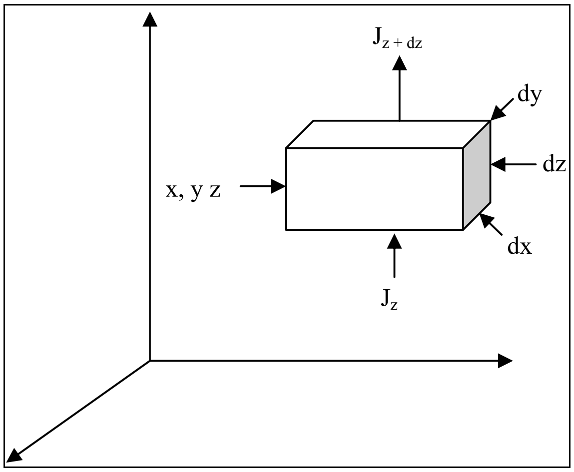


Figure 3: Calculation for neutrons movement

Consider the two faces of area dx dy which lie parallel to the x,y lane i.e. at right angles to the z direction. The number of neutrons entering the lower faces per second is then J_z dx dy, whereas the number leaving the upper face can be represented by J_z and J_{z+dz} are the neutron current density at the respective faces. The net rate of loss of neutrons out of the given volume element [2] through the faces parallel to the x,y plane is then

$$(J_{z+dz} - J_z)\, dx\, dy = \frac{\partial J_z}{Jz} (dx\, dy\, dz)$$

Upon dividing through by dx dy dz, the volume of the element, it follows that

$$\text{Rate of neutrons per unit volume in z direction} = \frac{\partial J_z}{\partial z}$$

Similar expressions are applicable to the x and y directions, so that

$$\text{Rate of neutrons per unit volume} = \frac{\partial J_x}{Jx} + \frac{\partial J_y}{Jy} + \frac{\partial J_y}{Jz}$$

where J is the neutron current density vector, i.e. the net number of neutrons flowing in a given direction in unit time through a unit area normal to the direction of flow; the symbol ∇ represents the divergence operators in vector notation.

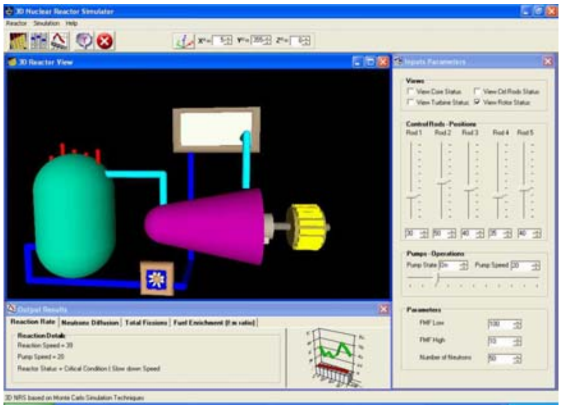
5. Experimental Setup:

In this computational experiment we consider a simple reactor model consisting of carbon atoms and two isotopes of uranium (U^{235} and U^{238}), all of which are intimately mixed (i:e the reactor can be considered homogenous).

The nuclear chain reaction propagates as follows:

- Splitting U^{235} atoms releases high speed neutrons.
- High speed neutrons are slowed down to thermal energies by collisions with carbon atoms.
- Atoms of U^{235} absorb thermal neutrons and fission occurs, creating more high speed neutrons.

The aim is to analyze how the proportions of constituent atoms in the reactor affected its performance and so how to balance these proportions in order to create an efficient system. This whole process can be done with help of an Interactive User-friendly Interface. User can change the location of camera to view reactor in x, y, z axes. It has also graphical orientation, 3D sound-effects and representation of parameters that are involved in the process. Reaction speed can be controlled as more neutrons will be produced to strike Uranium atoms with the user to move Control Rods position can be changed with sliding up-down as shown in figure 4.



```
Reaction Details
Reaction Speed = 30
Pump Speed = 20
Reactor Status = Critical Condition ! Slow down Speed
```

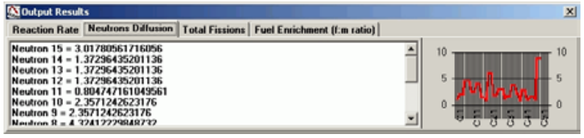
Figure 4: User Input Controls in 3D Nuclear Reactor Simulator

For a number of absorber materials and geometries, the computed control rod worths are compared with detailed Monte Carlo calculations and where possible, with experimental measurements. In general, control rod worths based on these diffusion-theory methods are found to be consistent with those from Monte Carlo calculations and with measured values [4]. Reactor control rods are composed of materials which strongly absorb thermal neutrons. In such materials the low-energy neutron flux varies rapidly as a function of position, which causes steep flux gradients near the absorber surface. It is assumed that a set of effective diffusion parameters can be found which depend on the nuclear cross sections of the absorber, its dimensions, and the mesh spacing used in diffusion-theory calculations to describe the control rod, but which are independent of the environment outside the absorber.

6. Conclusions:

The reactor performance is based on the balancing process between there being too little carbon to slow the neutrons down before absorption and there being too much carbon for the neutron to stand a chance of hitting a U^{235} atom and causing fission before being absorbed. These two effects are combined to produce a balance point.

For the controlled fission chain reaction, the neutrons diffusion as shown in figure 5 is dependent on appropriate position of control rods movement along with the factors of fuel moderator ratio.



```
Neutron 15 = 3.01780561716056
Neutron 14 = 1.37296435201136
Neutron 13 = 1.37296435201136
Neutron 12 = 1.37296435201136
Neutron 11 = 0.804747161049561
Neutron 10 = 2.3571242623176
Neutron 9 = 2.3571242623176
```

Figure 5: Output Results of Neutrons Diffusion

References

1. William R. Martin, “Introduction to Monte Carlo Methods”, DRAFT ORNL (1991).
2. Samuel Glasstone and Alexander Sesonske, “Nuclear Reactor Engineering”, US Department of Energy, CBS (1986).
3. Andrew N. Jackson, “The Nuclear Reactor simulation methods”, Journal of Nuclear Sciences, Vol 26 (1995).
4. M. M. Bretscher "Computing Control Rod worths In Thermal Research Reactors", ANL/RERTR/TM-29 (1997)
5. Roger D. Smith, “Simulation 101”, SISO Simulation Technology Magazine 2000