CHAPTER 1
AN OVERVIEW OF NUCLEAR MONTE CARLO

by M. R. Omar

When talking about nuclear reactor analysis with Monte Carlo method, physicists often fathomed that the simulated fission neutron random movements are tracked from the moment it is created during a fission event until its death. Such a direct simulation is done repeatedly for a large number of trials using a powerful computer. As a result, the statistical assemble of the simulated neutronic behaviour within a nuclear reactor can be observed. Particularly, the simulated neutronic behaviour is analysed by counting the number of neutrons occupying various regions within a nuclear reactor core. This information will finally form a spatial distribution of neutrons count over the entire reactor core. Traditionally, the spatial distribution of neutrons count is calculated by solving a specialised partial differential equation rather than simulating a large number of actual neutron movements. Here, the former technique is formally known as the deterministic method and the latter is known as the Monte Carlo method. Each of these techniques entails several pros and cons in terms of problem-solving capability. And of course, the Monte Carlo method is a formidable tool in nuclear reactor analysis due to its ability to simulate neutron movements in various complicated reactor core geometries.

1.1 The Monte Carlo Neutron Transport Method

At this level, it is convenient to briefly illustrate a simple Monte Carlo simulation of neutron movements within a typical system, say, a slab of fissile material. Among nuclear Monte Carlo physicists, such movements are recognized as neutron transport phenomenon. When a neutron travels from point $A(x_A, y_A, z_A)$ to point $B(x_B, y_B, z_B)$, one can alternatively say that the neutron is being *transported* from point A to point B. Initially, a Monte Carlo physicist will routinely provide an initial guess of fission source locations. Each of these locations will be the starting point (or sometimes termed as the birth location) of a simulated fission neutron created in the computer memory. Afterwards, a fission neutron batch size, say, $M=10^6$ is assigned and kept constant throughout the entire simulation. Subsequently, a queue which holds $M=10^6$ fission neutrons awaiting to be simulated is created in the computer memory. Their birth locations are randomly picked from the initial guess of fission source locations

initialized beforehand. A fission neutron is chosen from the queue and simulated. Conventionally, the starting locations of these fission neutrons form a spatial distribution which is known as the *fission source distribution*.

Fig. 1.1 depicts the random series of collisions of a fission neutron selected from the top of a queue. The neutron is programmatically ejected from its birth location at *A* and randomly transported within a slab of fissile material. Next, numbers between zero and unity are randomly generated by the computer. Later, these numbers are used to decide where the neutron collision takes place and what type of neutron-nucleus interaction occur at the collision location. Whether the neutron undergoes scattering reaction, or fission, or being captured by the nucleus at the collision site – it all depends on the fate dictated by the generated random numbers. And of course, these random choices are based on the rules of physics and probabilities represented by a quantity known as the *neutron cross section*. The value of the neutron cross section for various types of materials and reactions are gathered and stored in a formatted data file. Such a data file is commonly identified as a nuclear data file. A nuclear data file is provided by various organizations. For example, ENDF (Evaluated Nuclear Data File) is provided by Los Alamos National Laboratory (LANL) (Chadwick et al., 2011) and JENDL (Japanese Evaluated Nuclear Data Library) is provided by Japanese Atomic Energy Agency (Igarasi, Nakagawa, Kikuchi, Asami, & Narita, 1979).

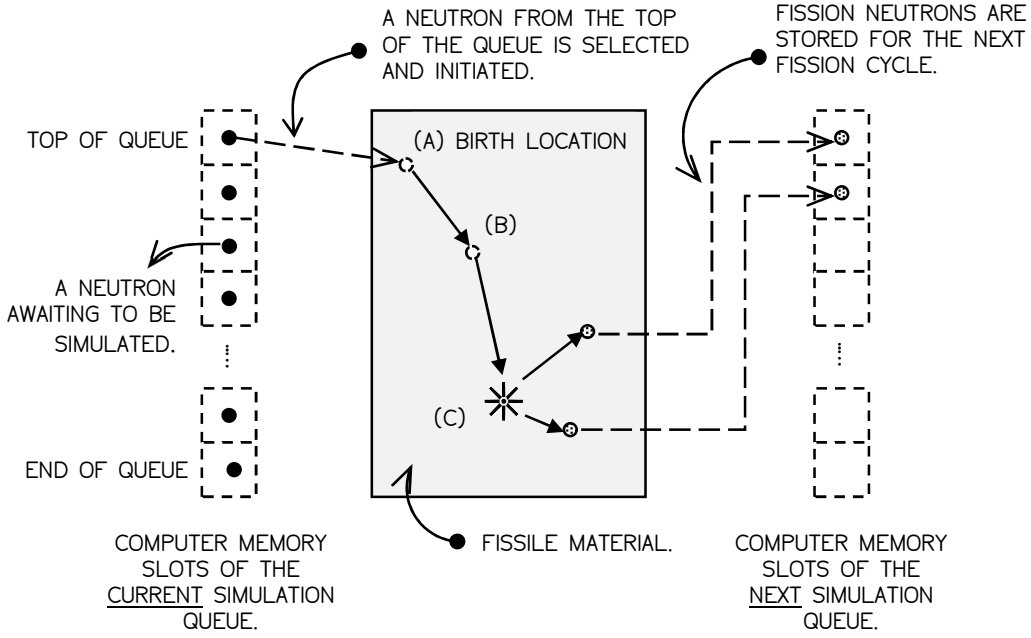

Figure 1.1: Overview of Monte Carlo neutron transport simulation.

Back to the neutron simulation, suppose the neutron collides at location *B*. After that, the neutron is scattered in the direction portrayed in Fig. 1.1. Plus, the scattering direction is randomly generated based on the physical scattering angle distribution. At collision point *C*, fission happens, ending in the death of the incoming neutron due to absorption and the birth of two outgoing fission neutrons. At this point, the simulation

is said to complete the first fission cycle of a single fission neutron. These two fission neutrons are not further tracked but they are saved in a new neutron queue for later tracking during the next fission cycle. This neutron history is now complete. The next neutron from the queue is ejected from its corresponding birth location and further tracked. This process continues until the neutron source queue is exhausted. As more histories are followed, the neutron distributions become favourably known. The quantities of interest, e.g. the neutron flux, track length or whatever the nuclear Monte Carlo physicist requests are tallied, simultaneously with the estimates of the statistical uncertainty of the tallies.

The Monte Carlo method (Metropolis & Ulam, 1949) is employed to reproduce a theoretically statistical phenomenon such as the way neutrons interact with materials. It is effective for simulating complex problems that cannot be modelled by computational codes that implement the standard numerical method. In the Monte Carlo method, the discrete probabilistic events that consist of realistic processes are simulated sequentially. Traditionally, neutrons behaviour is predicted by solving an integrodifferential equation identified as the neutron transport equation (Lamarsh & Baratta, 1955). The solution of the transport equation is the neutron flux distribution, which is a function of position, energy and time. The flux distribution is a useful piece of information in nuclear reactor analysis that enables nuclear engineers to design a practical and secure nuclear system. In the deterministic method, the neutron transport problem is solved using unique mathematical techniques, such as the Greens' function method (Öztürk, Anli, & Güngör, 2006) and the collision probability method (Raghav, 1977) (Lefvert, 1979). In contrast, the Monte Carlo method inherently 'solves' the neutron transport equation via the actual simulation of neutron random walk movements.

1.2 Monte Carlo Codes for Nuclear Reactor Analysis

A nuclear Monte Carlo code is a computer code that simulates nuclear processes, typically the way neutrons behave and move inside a nuclear reactor (Duderstadt & Hamilton, 1976). Recall that the example of such a simulation has been briefly described in the previous section. The execution of a reactor code produces information such as the stability of a nuclear reactor. It helps nuclear engineers to properly design and control the nuclear system. Most importantly, simulating the way neutrons behave in a reactor allows a nuclear Monte Carlo physicist to estimate the number of neutrons within the reactor. The number of neutrons in a particular reactor is proportional to the amount of fission power produced by the reactor. Essentially, the stability of a nuclear system is quantified using the multiplication factor, k. It assesses the rate of growth or decay of the total number of neutrons within the system. Briefly, the multiplication factor is defined as (Carter & Cashwell, 1975),

$$k = \frac{\begin{array}{c}\text{the rate of neutron production via} \\ \text{fission within the system}\end{array}}{\text{the rate of neutron loss within the system}} \tag{1.1}$$

To illustrate, an unstable nuclear system will have a typical value of $k > 1$, which can inadvertently lead to an exponential and uncontrollable growth of neutron population within the system. This will then increase the possibility of inducing an

uncontrollable amount of fission reactions. Consequently, a tremendous amount of fission energy is tapped and thus causing a nuclear disaster such as Fukushima Dai-ichi incident (Hirose, 2012) and Chernobyl incident (Beresford et al., 2016).

In nuclear reactor analysis, there are two distinct types of nuclear code and they are the nuclear design code and the reactor core management code. The solving technique implemented in both codes can be categorized as the deterministic method or the Monte Carlo method. The purpose of a nuclear design code is to enable nuclear engineers to first design nuclear devices or even a nuclear reactor. Typically, a design code allows the user to define arbitrary geometries that compose of various different types of materials. For example, a design code can be used to design a radiation shielding wall (Cho et al., 2004) or even to study the production of weapons-grade plutonium (Glaser & Ramana, 2007) for military purpose. In contrast, a core management code is a code specifically designed for a particular nuclear reactor core. It allows nuclear engineers to make decisions on managing reactor core compositions such as fuel reshuffling, nuclear fuel replacements and allocation of various irradiation facilities inside the reactor core.

Several examples of the deterministic code are TRIGLAV (Peršič et al., 2017) for generic TRIGA reactors, and APOLLO (Mathonniere & Stankovski, 1992) used by EDF Energy and Areva. Whereas some examples of the Monte Carlo nuclear system design code are MCNP by Los Alamos National Laboratory (LANL) (X-5 Monte Carlo Team, 2005) and OpenMC by Massachusetts Institute of Technology (MIT) (Romano & Forget, 2013).

Criticality calculation, or sometimes called the eigenvalue calculation, is a well-known neutron transport simulation technique to determine the multiplication factor of a certain nuclear system (Duderstadt & Hamilton, 1976). Here, neutron productions via fission reaction are included in the simulation. Most nuclear design codes and core management codes have the capability of running criticality calculations. In a deterministic criticality code, a modified neutron transport equation called the k-eigenvalue equation is solved and computed by various mathematical methods available (Duderstadt & Hamilton, 1976). Also, a deterministic code is computationally less expensive since there are no random processes involved.

References

Beresford, N. A., Fesenko, S., Konoplev, A., Skuterud, L., Smith, J. T., & Voigt, G. (2016). Thirty years after the Chernobyl accident: What lessons have we learnt? *Journal of Environmental Radioactivity*, *157*, 77–89. https://doi.org/10.1016/J.JENVRAD.2016.02.003

Carter, L. L., & Cashwell, E. D. (1975). *Particle-transport simulation with the Monte Carlo method*. *Other Information: Orig. Receipt Date: 30-JUN-76*. https://doi.org/10.2172/4167844

Chadwick, M. B., Herman, M., Obložinský, P., Dunn, M. E., Danon, Y., Kahler, A. C., … Young, P. G. (2011). ENDF/B-VII.1 Nuclear Data for Science and Technology: Cross Sections, Covariances, Fission Product Yields and Decay

Data. *Nuclear Data Sheets*, *112*(12), 2887–2996. https://doi.org/10.1016/J.NDS.2011.11.002

Cho, il J., You, G. S., Choung, W. M., Ku, J. H., Kook, D. H., Hwang, Y. H., … Park, S. W. (2004). Evaluation on the Radiation Shielding for the Design of a Hot Cell Facility. *Journal of Nuclear Science and Technology*, *41*(sup4), 97–100. https://doi.org/10.1080/00223131.2004.10875654

Duderstadt, J. J., & Hamilton, L. J. (1976). Nuclear Reactor Analysis. *Mechanical Engineering*. https://doi.org/10.1109/TNS.1977.4329257

Glaser, A., & Ramana, M. V. (2007). Weapon-Grade Plutonium Production Potential in the Indian Prototype Fast Breeder Reactor. *Science & Global Security*, *15*(2), 85–105. https://doi.org/10.1080/08929880701609154

Hirose, K. (2012). 2011 Fukushima Dai-ichi nuclear power plant accident: summary of regional radioactive deposition monitoring results. *Journal of Environmental Radioactivity*, *111*, 13–17. https://doi.org/10.1016/J.JENVRAD.2011.09.003

Igarasi, S., Nakagawa, T., Kikuchi, Y., Asami, T., & Narita, T. (1979). Japanese Evaluated Nuclear Data Library, version-1. Retrieved from https://inis.iaea.org/search/search.aspx?orig_q=RN:11512725

Lamarsh, J. R., & Baratta, A. J. (1955). *Introduction to Nuclear Engineering*. *American Journal of Physics* (Vol. 23). https://doi.org/10.1119/1.1933896

Lefvert, T. (1979). New applications of the collision probability method in neutron transport theory. *Progress in Nuclear Energy*, *4*(2), 97–118. https://doi.org/10.1016/0149-1970(79)90014-3

Mathonniere, G., & Stankovski, Z. (1992). *The APOLLO-I assembly spectrum code*. International Atomic Energy Agency (IAEA). Retrieved from http://inis.iaea.org/search/search.aspx?orig_q=RN:24033459

Öztürk, H., Anli, F., & Güngör, S. (2006). Spectral Green's Function Method for Neutron Transport with Backward and Forward Scattering in 1-D Slab and Spherical Geometry. *Transport Theory and Statistical Physics*, *35*(5–7), 229–250. https://doi.org/10.1080/00411450600901771

Peršič, A., Žagar, T., Ravnik, M., Slavič, S., Žefran, B., Ćalić, D., … Snoj, L. (2017). TRIGLAV: A program package for TRIGA reactor calculations. *Nuclear Engineering and Design*, *318*, 24–34. https://doi.org/10.1016/J.NUCENGDES.2017.04.010

Raghav, H. P. (1977). Collision probability method and its applications. Retrieved from https://inis.iaea.org/search/search.aspx?orig_q=RN:8347582

Romano, P. K., & Forget, B. (2013). The OpenMC Monte Carlo particle transport code. *Annals of Nuclear Energy*, *51*, 274–281. https://doi.org/10.1016/j.anucene.2012.06.040

X-5 Monte Carlo Team. (2005). MCNP - A General Monte Carlo N-Particle Transport Code, Version 5. *Los Alamos Nuclear Laboratory*, *I*, 2-71-2–80. https://doi.org/LA-UR-03-1987