TRIMON:
THEORY AND GUIDELINES

by M. R. Omar

PART I:
THEORETICAL BACKGROUND OF TRIMON

By M. R. Omar

Table of Contents

CHAPTER 1
AN OVERVIEW OF NUCLEAR MONTE CARLO

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*.

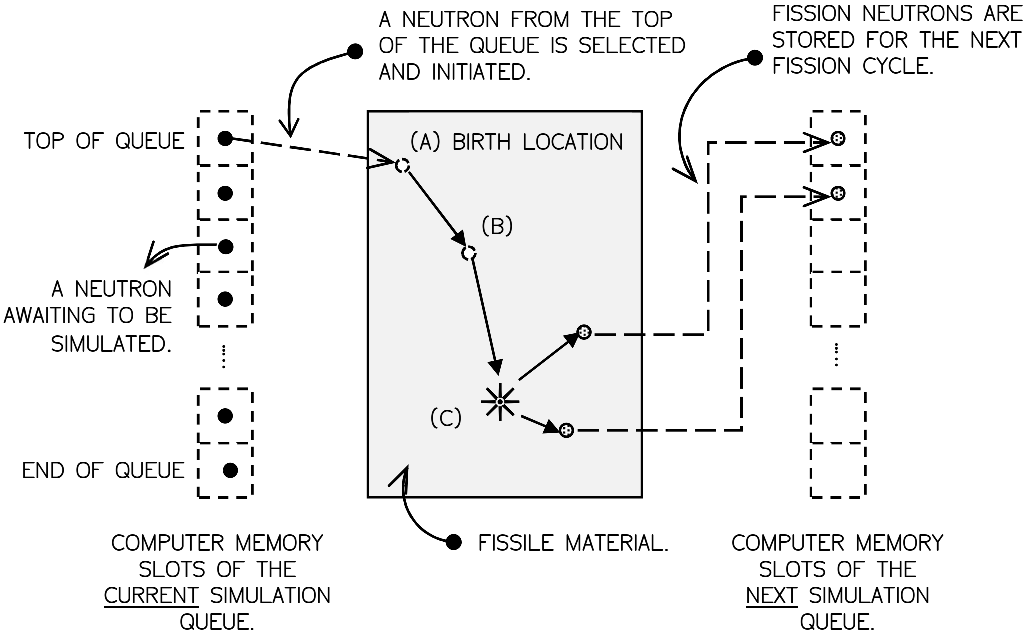

A NEUTRON FROM THE TOP OF THE QUEUE IS SELECTED AND INITIATED.
FISSION NEUTRONS ARE STORED FOR THE NEXT FISSION CYCLE.
TOP OF QUEUE
(A) BIRTH LOCATION
(B)
A NEUTRON AWAITING TO BE SIMULATED.
(C)
END OF QUEUE
COMPUTER MEMORY SLOTS OF THE CURRENT SIMULATION QUEUE.
FISSILE MATERIAL.
COMPUTER MEMORY SLOTS OF THE NEXT SIMULATION QUEUE.

Figure 1.1: Overview of Monte Carlo neutron transport simulation.

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).

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 Daiichi 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

CHAPTER 2
NEUTRON TRANSPORT THEORY

In a reactor core, neutrons move in complicated trajectories due to constant collisions with nuclei. Typically, these recurring collisions cause the neutron trajectories to appear to be zigzag. For instance, source neutrons were originated from their corresponding birth locations, $\mathbf{r}$, moving with particular energy, E, and direction $\widehat{\mathbf{\Omega}}$. Afterwards, they appear at other positions, $\mathbf{r}'$, at a later time. These neutrons could also change its energy and direction into E' and $\widehat{\mathbf{\Omega}}'$, respectively after a collision at $\mathbf{r}'$. In that sense, these neutrons are said to have been transported from the current state $(\mathbf{r}, E, \widehat{\mathbf{\Omega}})$ to the next subsequent state $(\mathbf{r}', E', \widehat{\mathbf{\Omega}}')$. Correspondingly, the study of such a process is coined as the neutron transport theory. In this chapter, an exact equation which describes the neutron transport phenomena will be introduced. Such an equation is recognized as *the neutron transport equation* and the key objective of this study is to solve the equation. The readers will also be introduced with the basic concepts of the neutron transport theory before jumping into the battle of solving the equation.

2.1 Neutron Density and Flux

The central objective of this section is to familiarize the ways of counting the number of neutrons within a nuclear system. In this study, it is essential to understand the approach of characterizing neutrons within a medium. To begin with, we define the *neutron density*, $N(\mathbf{r},t)\,d^3r$, at a point $\mathbf{r} \in \mathbb{R}^3$ within a reactor core and at time t, as the expected number of neutrons in the unit volume d^3r. It is convenient to characterize neutrons according to their energy, E, and direction, $\widehat{\boldsymbol{\Omega}}$, such that,

$$\begin{aligned}\widehat{\boldsymbol{\Omega}} = \frac{\mathbf{v}}{|\mathbf{v}|} &= u\mathbf{e}_x + v\mathbf{e}_y + w\mathbf{e}_z \\ &= \underbrace{\sin\theta\cos\theta}_{u}\,\mathbf{e}_x + \underbrace{\sin\theta\sin\varphi}_{v}\,\mathbf{e}_y + \underbrace{\cos\theta}_{w}\,\mathbf{e}_z\end{aligned} \tag{2.1}$$

where $\mathbf{v}$ is the neutron velocity, $\{\mathbf{e}_x, \mathbf{e}_y, \mathbf{e}_z\}$ are the basis vectors of the Cartesian coordinate, $\theta \in [0,\pi)$ and $\varphi \in [0,2\pi)$. Correspondingly, the zenith angle, θ, and the azimuthal angle, φ, are indicated in Fig. 2.1. Also,

$$\sqrt{u^2 + v^2 + w^2} = 1 \tag{2.2}$$

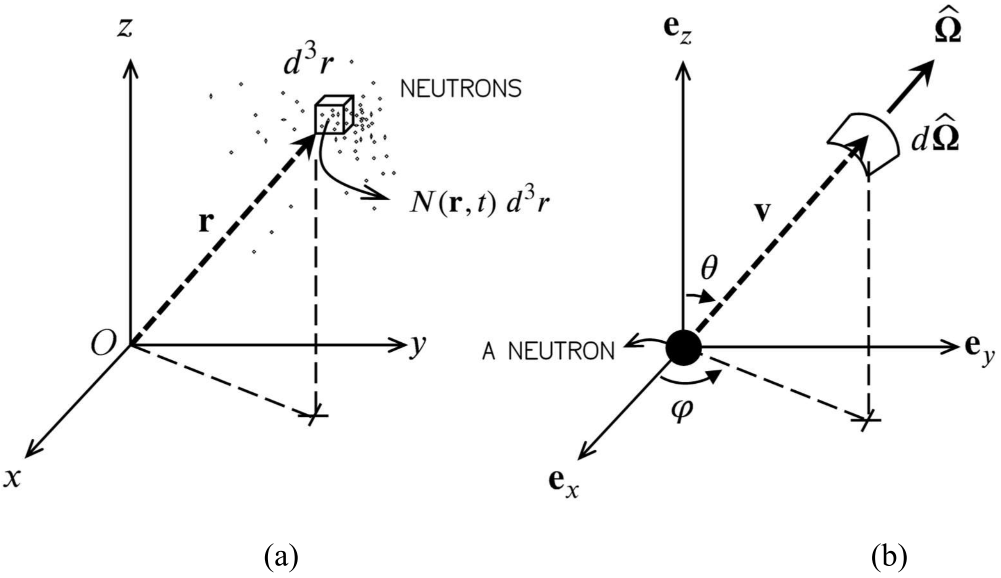


Figure 2.1: (a) The neutron density, $N(\mathbf{r},t)$; and (b) the direction variables characterizing a neutron.

By referring to Fig. 2.1(a), consider a unit volume d^3r containing a 'mixture' of neutrons with assorted energies and directions. One could possibly select the neutrons with a specific energy E and direction $\widehat{\boldsymbol{\Omega}}$ from d^3r. Thus, the expected number of neutrons in $d^3\mathbf{r}$ at position $\mathbf{r}$, with energy E about dE, moving towards the direction $\hat{\mathbf{\Omega}}$ in solid angle $d\hat{\mathbf{\Omega}}$ at time t is given by,

$$n(\mathbf{r}, E, \hat{\mathbf{\Omega}}, t)\ d^3r\ dE\ d\hat{\mathbf{\Omega}} \tag{2.3}$$

Note that the *angular neutron density*, $n(\mathbf{r}, E, \hat{\mathbf{\Omega}}, t)$, is introduced in Eq. (2.3). Subsequently, $n(\mathbf{r}, E, \hat{\mathbf{\Omega}}, t)$ is defined similarly as $N(\mathbf{r}, t)$, however, the former considers the neutron energy and direction. In the neutron transport theory, it is convenient to express Eq. (2.3) in terms of the *angular neutron flux*, $\psi(\mathbf{r}, E, \hat{\mathbf{\Omega}}, t)$, where,

$$\psi(\mathbf{r}, E, \hat{\mathbf{\Omega}}, t) = v\ n(\mathbf{r}, E, \hat{\mathbf{\Omega}}, t) \tag{2.4}$$

where v is the neutron speed ($v = \sqrt{2E/m_{\mathrm{n}}}$, m_{n} is the neutron rest mass). The angular neutron flux has a unit of cm^{-2} s^{-1}. Subsequently, the *angular neutron current density* is defined as,

$$\mathbf{j}(\mathbf{r}, E, \hat{\mathbf{\Omega}}, t) = \hat{\mathbf{\Omega}}\ \psi(\mathbf{r}, E, \hat{\mathbf{\Omega}}, t) \tag{2.5}$$

Remark that $\mathbf{j}$ is a vector quantity such that,

$$\mathbf{j}(\mathbf{r}, E, \hat{\mathbf{\Omega}}, t) \cdot d\mathbf{A}\ dE\ d\hat{\mathbf{\Omega}} \tag{2.6}$$

is defined as the net rate at which neutrons with energy E about dE and direction $\hat{\mathbf{\Omega}}$ in $d\hat{\mathbf{\Omega}}$ crossing a unit area dA at time t. In certain circumstances, it is favourable to express Eqs. (2.4) and (2.5) without considering the neutron direction, thus,

$$\phi(\mathbf{r}, E, t) = \int_{4\pi} \psi(\mathbf{r}, E, \hat{\mathbf{\Omega}}, t)\ d\hat{\mathbf{\Omega}} \tag{2.7}$$

and,

$$\mathbf{J}(\mathbf{r}, E, t) = \int_{4\pi} \mathbf{j}(\mathbf{r}, E, \hat{\mathbf{\Omega}}, t)\ d\hat{\mathbf{\Omega}} \tag{2.8}$$

where the integration is taken over the entire solid angle, $d\hat{\mathbf{\Omega}}$, i.e. $[0, 4\pi)$. At this time, the quantity $\mathbf{J}$ is now defined as the *neutron current density*, where,

$$\mathbf{J}(\mathbf{r}, E, t) \cdot d\mathbf{A} \tag{2.9}$$

is the net rate at which neutrons with energy E pass through a surface area $d\mathbf{A}$. Note that the units of both $\mathbf{J}$ and ϕ are equivalent i.e. cm^{-2} sec^{-1}. However, $\mathbf{J}$ is a vector quantity characterizing the *net* rate at which neutron flow through a surface, $d\mathbf{A}$, oriented in a given direction, $d\mathbf{A}/|d\mathbf{A}|$. Whereas ϕ characterizes the total rate at which neutrons pass through an area, regardless of its orientation. Thus, $\mathbf{J}$ is a more convenient quantity to describe neutron leakage from a system such as a nuclear reactor core.

2.2 Neutron Cross Sections

In the neutron transport theory, the concept of neutron cross section is one of the central aspects that determine neutron behaviour within a system. It conveys the likelihood of an interaction between an incident neutron and a target nucleus to occur (Lamarsh & Baratta, 1955). Intentionally, consider a stream of incident neutrons travels through the material within a nuclear reactor. Intuitively, there is a probability that a fraction of these neutrons interacts with the nuclei of the material. The physical quantity that expresses the likelihood of a neutron-nuclear interaction is known as the *microscopic neutron cross section*, $\sigma(\mathbf{r}, E)$. Here, $\sigma(\mathbf{r}, E)$ is a function of the neutron position, $\mathbf{r}$, and the incident neutron energy, E. The former indicates that the σ is dependent on the material properties. Respectively, its value varies across the distinct parts of the reactor core region. Whereas, the latter indicates that the value of σ also varies according to the incident neutron energy. Moreover, the larger the value of σ, the greater the possibility of a neutron-nucleus interaction characterized by σ to occur. Also, σ has a unit of cm^{-2} or "barn" ($\times 10^{-24}$ cm^{-2}).

There is also another form of neutron cross section that considers the likelihood of interaction between an incident neutron and a target nucleus in a chunk of material instead of an individual atom. It is known as the *macroscopic neutron cross section*, $\Sigma(\mathbf{r}, E)$. Formally, $\Sigma(\mathbf{r}, E)$ is defined as the probability of a neutron interaction to happen per unit path length travelled by the neutron. Also, $\Sigma(\mathbf{r}, E)$ has a unit of cm^{-1} and the relationship between $\Sigma(\mathbf{r}, E)$ and $\sigma(\mathbf{r}, E)$ is given by,

$$\Sigma(\mathbf{r}, E) = N_D \sigma(\mathbf{r}, E) \tag{2.10}$$

where N_D is the number of atoms of the material per unit volume or simply called as *the number density*. Equally important, there are various types of neutron-nuclear interactions. These interactions include neutron capture, scattering, fission, and etc. It is now convenient to introduce the microscopic total neutron cross section, $\sigma_\mathrm{t}(\mathbf{r}, E)$,

$$\sigma_\mathrm{t}(\mathbf{r}, E) = \sum_{j=1}^{m} \sigma_j(\mathbf{r}, E) \tag{2.11}$$

and similarly, the macroscopic total neutron cross section, $\Sigma_{\mathrm{t}}(\mathbf{r}, E)$,

$$\begin{aligned}\Sigma_{\mathrm{t}}(\mathbf{r}, E) &= \sum_{j=1}^{m} \Sigma_j(\mathbf{r}, E) \\ &= N_D \sum_{j=1}^{m} \sigma_j(\mathbf{r}, E)\end{aligned} \tag{2.12}$$

where m is the total number of different types of neutron-nucleus interaction with the summation index, j, representing the different types of interaction.

2.3 Double-differential Scattering Cross Sections

Essentially, $\sigma(\mathbf{r}, E)$ does not conveys the probability of a neutron to possess the outgoing energy, E', and the outgoing direction, $\widehat{\mathbf{\Omega}}'$, after an interaction event. For instance, scattering reaction is a type of interaction that changes the incident neutron energy and direction, i.e. $(E, \widehat{\mathbf{\Omega}})$, into a new set of energy and direction, i.e. $(E', \widehat{\mathbf{\Omega}}')$. Thus, the likelihood of a scattering reaction that causes the change in neutron energy and direction $(E, \widehat{\mathbf{\Omega}})$ into $(E', \widehat{\mathbf{\Omega}}')$ is expressed in term of $\sigma_{\mathrm{s}}(\mathbf{r}, E \rightarrow E', \widehat{\mathbf{\Omega}} \rightarrow \widehat{\mathbf{\Omega}}')$. At this point, $\sigma_{\mathrm{s}}(\mathbf{r}, E \rightarrow E', \widehat{\mathbf{\Omega}} \rightarrow \widehat{\mathbf{\Omega}}')$ is known as the *microscopic double-differential scattering cross section*. Now, the dependency of σ_{s} on the incident neutron direction, $\widehat{\mathbf{\Omega}}$, is usually neglected because the nuclei in any material are usually randomly oriented. Consequently, when all possible nuclear orientations are considered, the dependency of σ_{s} on $\widehat{\mathbf{\Omega}}$ averages out.

Crucially, the relationship between $\sigma_{\mathrm{s}}(\mathbf{r}, E)$ and $\sigma_{\mathrm{s}}(\mathbf{r}, E \rightarrow E', \widehat{\mathbf{\Omega}} \rightarrow \widehat{\mathbf{\Omega}}')$ is given by,

$$\sigma_{\mathrm{s}}(\mathbf{r}, E) = \int_{4\pi} \int_{0}^{\infty} \sigma_{\mathrm{s}}(\mathbf{r}, E \rightarrow E', \widehat{\mathbf{\Omega}} \rightarrow \widehat{\mathbf{\Omega}}')\, dE'\, d\widehat{\mathbf{\Omega}}' \tag{2.13}$$

Likewise, the property in Eq. (2.10) can also be applied to obtain the macroscopic double-differential scattering cross section,

$$\Sigma_{\mathrm{s}}(\mathbf{r}, E \rightarrow E', \widehat{\mathbf{\Omega}} \rightarrow \widehat{\mathbf{\Omega}}') = N_D \sigma_{\mathrm{s}}(\mathbf{r}, E \rightarrow E', \widehat{\mathbf{\Omega}} \rightarrow \widehat{\mathbf{\Omega}}'). \tag{2.14}$$

In addition, both microscopic and macroscopic double-differential cross sections do not depend on the incident neutron direction. However, they depend on the scattering angle, α, which is the angle between $\widehat{\mathbf{\Omega}}$ and $\widehat{\mathbf{\Omega}}'$. It is convenient to express the change in the neutron direction in terms of the scattering cosine, μ_0,

$$\mu_0 = \widehat{\mathbf{\Omega}} \cdot \widehat{\mathbf{\Omega}}' = \cos\alpha \tag{2.15}$$

Finally, one may also express the macroscopic double-differential scattering cross section in terms of μ_0,

$$\begin{aligned}\Sigma_s(\mathbf{r}, E \to E', \widehat{\mathbf{\Omega}} \to \widehat{\mathbf{\Omega}}') &\equiv \Sigma_s(\mathbf{r}, E \to E', \widehat{\mathbf{\Omega}} \cdot \widehat{\mathbf{\Omega}}') \\ &\equiv \Sigma_s(\mathbf{r}, E \to E', \mu_0)\end{aligned} \tag{2.16}$$

2.4 Neutron Interaction Rate

Ideally, we begin introducing the reaction rate density, R, which is defined as the expected number of neutron-nucleus interactions that occur per unit volume and per unit time. Subsequently, the expected number of interactions per second, f, experienced by a neutron moving with an average speed of v within the material is given by,

$$f = v\Sigma(\mathbf{r}, E) \tag{2.17}$$

Thus, the total reaction rate density of a type of interaction in a volume d^3r caused by an incident neutron with energy and direction $(E, \widehat{\mathbf{\Omega}})$ at position $\mathbf{r}$ and time t is given by,

$$\begin{aligned}R(\mathbf{r}, E, \widehat{\mathbf{\Omega}}, t)\, d^3r &= f \times\ n(\mathbf{r}, E, \widehat{\mathbf{\Omega}}, t)\, d^3r \\ &= v\ \Sigma(\mathbf{r}, E)\ n(\mathbf{r}, E, \widehat{\mathbf{\Omega}}, t)\, d^3r \\ &= \Sigma(\mathbf{r}, E)\ \psi(\mathbf{r}, E, \widehat{\mathbf{\Omega}}, t)\, d^3r\end{aligned} \tag{2.18}$$

The similar concept also relevant to the scattering reaction, where the expected number of scattering reactions that changes the energy and direction of a neutron from $(E, \widehat{\mathbf{\Omega}})$ into $(E', \widehat{\mathbf{\Omega}}')$ is given by,

$$R(\mathbf{r}, E, \widehat{\mathbf{\Omega}}, t)\, d^3r = \Sigma_s(\mathbf{r}, E \to E', \widehat{\mathbf{\Omega}} \to \widehat{\mathbf{\Omega}}')\psi(\mathbf{r}, E, \widehat{\mathbf{\Omega}}, t)\, d^3r \tag{2.19}$$

2.5 Neutron Transport Equation

Principally, the neutron transport theory uncovers the distribution of neutrons in a system. The theory considers the movement of neutrons and the way they interact with the materials contained in the system. The distribution of neutrons in a system, typically in a reactor core, can be obtained by solving the neutron transport equation. One can derive the neutron transport equation by balancing various mechanisms that cause gain or loss of neutrons within a system.

At this instance, it is appropriate to begin deriving the neutron transport equation by considering the rate of change of the neutron density, $n(\mathbf{r}, E, \hat{\mathbf{\Omega}}, t)$ within an infinitesimal volume, d^3r,

$$\begin{aligned} \int \frac{\partial}{\partial t} n(\mathbf{r}, E, \hat{\mathbf{\Omega}}, t)\, d^3r &= \int \frac{1}{\upsilon} \frac{\partial}{\partial t} \psi(\mathbf{r}, E, \hat{\mathbf{\Omega}}, t)\, d^3r \\ &= R_+ - R_- \end{aligned} \tag{2.20}$$

where R_+ is the total rate of interactions that cause the gain of neutrons in d^3r, and R_- is the total rate of interactions that cause the loss of neutrons in d^3r.

2.5.1 Neutron Loss via Net Leakage

At first, consider a few neutron currents, $\hat{\mathbf{\Omega}}\psi$, entering and leaving an infinitesimal volume d^3r of a material through the surface S which defines the boundary of d^3r. In essence, the difference between the rate of neutrons entering and leaving d^3r through S is equal to the resulting neutron leakage rate,

$$\begin{aligned} \begin{Bmatrix} \text{Net} \\ \text{Leakage} \end{Bmatrix} &= \oint_S \mathbf{j}(\mathbf{r}, E, \hat{\mathbf{\Omega}}, t) \cdot d\mathbf{S} \\ &= \oint_S \hat{\mathbf{\Omega}}\ \psi(\mathbf{r}, E, \hat{\mathbf{\Omega}}, t) \cdot d\mathbf{S} \end{aligned} \tag{2.21}$$

Here, Gauss' theorem of vector calculus can be applied, and Eq. (2.21) will reduce into,

$$\begin{aligned} \begin{Bmatrix} \text{Net} \\ \text{Leakage} \end{Bmatrix} &= \int \mathbf{\nabla} \cdot \left(\hat{\mathbf{\Omega}}\psi(\mathbf{r}, E, \hat{\mathbf{\Omega}}, t) \right) d^3r \\ &= \int \hat{\mathbf{\Omega}} \cdot \mathbf{\nabla}\psi(\mathbf{r}, E, \hat{\mathbf{\Omega}}, t)\, d^3r \end{aligned} \tag{2.22}$$

2.5.2 Neutron Loss via Disappearance Interactions

Suppose an incident neutron collides with a nucleus in an infinitesimal volume d^3r. Naturally, there is a possibility that an interaction that causes the disappearance of the neutron to occur. If such a disappearance interaction is possible, the neutron is considered loss from d^3r. For example, during a neutron capture interaction e.g. (n, γ) reaction and (n, α) reaction, an incident neutron is absorbed by the nucleus. Consequently, a secondary particle, e.g. γ-ray or α-particle, is released as the product of the reaction. Relevantly, scattering reaction is equally considered as a disappearance interaction. In scattering reaction, an incident neutron with the energy and direction $(E, \hat{\mathbf{\Omega}})$ is considered lost whilst a new secondary neutron with the energy and direction $(E', \hat{\mathbf{\Omega}}')$ is ejected from the nucleus. When a stream of neutrons with energy E and direction $\hat{\mathbf{\Omega}}$ travel through d^3r, the rate of neutron loss in d^3r due to disappearance reactions is given by,

$$\begin{Bmatrix} \text{Total} \\ \text{loss rate} \end{Bmatrix} = \int \Sigma_{\mathrm{t}}(\mathbf{r}, E)\psi(\mathbf{r}, E, \hat{\mathbf{\Omega}}, t)\, d^3r\,. \tag{2.23}$$

Conveniently, all reactions at the collision site are considered as disappearance reactions, thus the total macroscopic neutron cross section is incorporated in Eq. (2.23).

2.5.3 Neutron Gain via In-Scattering

Recall that in the neutron transport theory, neutrons are characterised according to their energy and direction, i.e. $(E, \hat{\mathbf{\Omega}})$. In this section, our objective is to analyse the expected number of neutrons with energy and direction $(E, \hat{\mathbf{\Omega}})$ that appear in an infinitesimal volume, d^3r, due to in-scattering. The term "in-scattering" is coined to indicate the interaction where an incident neutron with any energy and direction, $(E', \hat{\mathbf{\Omega}}')$, are scattered *into* the energy and direction of interest, $(E, \hat{\mathbf{\Omega}})$. For generalization purpose, the neutron in-scattering accounts all types of scattering interaction, e.g. elastic scattering, inelastic scattering, potential scattering, etc.

Most importantly, the gain rate of neutrons with energy and direction $(E, \hat{\mathbf{\Omega}})$ in d^3r due to in-scattering of an incident neutron with energy and direction $(E', \hat{\mathbf{\Omega}}')$ is given by,

$$\begin{Bmatrix} \text{Total in-} \\ \text{scattering} \\ \text{rate} \end{Bmatrix} = \int \Sigma_{\mathrm{s}}(\mathbf{r}, E' \to E, \hat{\mathbf{\Omega}}' \to \hat{\mathbf{\Omega}})\, \psi(\mathbf{r}, E', \hat{\mathbf{\Omega}}', t)\, d^3r \tag{2.24}$$

Equally important, to obtain the total neutron gain rate via in-scattering, the sum of the contributions of all incident neutron energies, E', and directions, $\hat{\mathbf{\Omega}}'$, are considered,

$$\left\{\begin{matrix}\text{Total in-}\\ \text{scattering}\\ \text{rate}\end{matrix}\right\} = \int \int_{4\pi} \int_0^{\infty} \Sigma_s(\mathbf{r}, E' \rightarrow E, \hat{\mathbf{\Omega}}' \rightarrow \hat{\mathbf{\Omega}})\, \psi(\mathbf{r}, E', \hat{\mathbf{\Omega}}', t)\, dE'\, d\hat{\mathbf{\Omega}}'\, d^3r\,. \tag{2.25}$$

2.5.4 Neutron Gain via Fission

Undeniably, fission reaction is the most important neutron-nucleus interaction that drives the power generation in a nuclear reactor core. A typical nuclear fission reaction such as

$$\mathrm{n} + {}^{235}_{92}U \rightarrow {}^{236}_{92}U^{*} \rightarrow \text{fission products}$$

ejects out a mixture of reaction products, including the daughter nuclei and several fission neutrons plus numerous gammas, betas, and neutrinos. The fission fragment nuclei generated by the fission reaction are both highly charged and remarkably high in energy. They slow down through collisions with neighbouring nuclei and dissipating energy during the process. This is, in reality, the primary mechanism by which the fission energy finally appears as heat formed in the fuel material.

Equally important, several neutrons are also produced during the fission reaction. These neutrons can be utilised to breed a fission chain reaction. Majority of these fission neutrons are produced promptly (within 10^{-14} sec) of the fission event and these neutrons are attributed to as prompt (Duderstadt & Hamilton, 1976). Nevertheless, less than 1% of the neutrons produced appear with an apparent time delay from the subsequent decay of radioactive fission products. These delayed neutrons are essential for the practical control of the fission chain reaction. Essentially, the total number of neutrons released in a fission reaction will fluctuate from one reaction to another. However, the average number of neutrons released per fission, ν, is of greater concern. This quantity depends on both the nuclear isotope involved and the incident neutron energy. In general, ν tends to increase with increasing incident neutron energy (Duderstadt & Hamilton, 1976; Lamarsh & Baratta, 1955).

In particular, fission neutrons are ejected with a distribution of energies, with the average energy being about 2 MeV. Such a distribution will primarily depend on the fissionable isotope involved. The energy distribution may also depend on the incident neutron energy and will vary for prompt and delayed neutrons. It is convenient to introduce the *fission spectrum*, $\chi(E)$, which is defined as the probability of having a fission neutron ejected with energy E as a result of a fission reaction.

Presume that $\nu(E')$ is the average number of neutrons produced per fission induced by an incident neutron with energy E'. Then, the total fission rate at which fission neutrons are generated in an infinitesimal volume d^3r is given by,

$$\int \int_{4\pi} \int_0^\infty \nu(E')\Sigma_f(\mathbf{r}, E')\psi(\mathbf{r}, E', \hat{\mathbf{\Omega}}', t)\, dE'\, d\hat{\mathbf{\Omega}}'\, d^3r. \tag{2.26}$$

Since we are only interested in knowing rate of fission reaction causing the birth of the fission neutrons with energy E and direction $\hat{\mathbf{\Omega}}$, thus, Eq. (2.26) can be modified into,

$$\chi(E)P(\hat{\mathbf{\Omega}}) \int \int_{4\pi} \int_0^\infty \nu(E')\Sigma_f(\mathbf{r}, E')\psi(\mathbf{r}, E', \hat{\mathbf{\Omega}}', t)\, dE'\, d\hat{\mathbf{\Omega}}'\, d^3r \tag{2.27}$$

where $P(\hat{\mathbf{\Omega}})$ is the probability of having a fission neutron ejected towards the direction $\hat{\mathbf{\Omega}}$. If the fission neutrons are anticipated to get emitted isotropically, then $P(\hat{\mathbf{\Omega}})$ is simply the inverse of all possible solid angles subtended by a fission neutron, i.e. 4π. Finally, the *fission* term of the transport equation is defined as the rate of fission neutron appearing in $(E, \hat{\mathbf{\Omega}})$:

$$\begin{aligned} &\left\{\begin{matrix}\text{Fission}\\ \text{rate}\end{matrix}\right\} \\ &= \frac{\chi(E)}{4\pi} \int \int_{4\pi} \int_0^\infty \nu(E')\Sigma_f(\mathbf{r}, E')\psi(\mathbf{r}, E', \hat{\mathbf{\Omega}}', t)\, dE'\, d\hat{\mathbf{\Omega}}'\; d^3r. \end{aligned} \tag{2.28}$$

2.5.5 The Differential Form of Neutron Transport Equation

At this point, all of the general interaction rate equations that describe the neutron gain and loss mechanisms in d^3r have been expressed. The neutron transport equation can be derived by rewriting Eq. (2.20) in terms of the net rate of neutrons appearing in $(E, \hat{\mathbf{\Omega}})$ and net rate of neutrons loss from $(E, \hat{\mathbf{\Omega}})$,

$$\begin{aligned} &\int \frac{1}{v}\frac{\partial}{\partial t}\psi(\mathbf{r}, E, \hat{\mathbf{\Omega}}, t)\, d^3r \\ &\qquad = -\left\{\begin{matrix}\text{Net}\\ \text{Leakage}\\ \text{rate}\end{matrix}\right\} - \left\{\begin{matrix}\text{Total}\\ \text{loss}\\ \text{rate}\end{matrix}\right\} + \left\{\begin{matrix}\text{Total in-}\\ \text{scattering}\\ \text{rate}\end{matrix}\right\} \\ &\qquad + \left\{\begin{matrix}\text{Fission}\\ \text{rate}\end{matrix}\right\} \end{aligned} \tag{2.29}$$

To proceed further, Eqs. (2.22), (2.23), (2.25) and (2.28) are substituted into Eq. (2.29). Finally, the volume integrals over the whole d^3r are cancelled off, and the final form of the neutron transport equation is given by,

$$\begin{aligned}\frac{1}{v}\frac{\partial\psi}{\partial t} &+ \hat{\mathbf{\Omega}}\cdot\nabla\psi(\mathbf{r},E,\hat{\mathbf{\Omega}},t)+\Sigma_{\mathrm{t}}(\mathbf{r},E)\psi(\mathbf{r},E,\hat{\mathbf{\Omega}},t)\\ &= \int_{4\pi}\int_{0}^{\infty}\Sigma_{\mathrm{s}}(\mathbf{r},E'\rightarrow E,\hat{\mathbf{\Omega}}'\rightarrow\hat{\mathbf{\Omega}})\psi(\mathbf{r},E',\hat{\mathbf{\Omega}}',t)\,dE'\,d\hat{\mathbf{\Omega}}'\\ &+\frac{\chi(E)}{4\pi}\int_{4\pi}\int_{0}^{\infty}\nu(E')\Sigma_{\mathrm{f}}(\mathbf{r},E')\psi(\mathbf{r},E',\hat{\mathbf{\Omega}}',t)\,dE'\,d\hat{\mathbf{\Omega}}'\end{aligned} \tag{2.30}$$

Alas, solving Eq. (2.30) is rather difficult. It is necessary to simplify the form of Eq. (2.30) before any attempts are made to solve it. One of the well-known simplification methods is via the neutron diffusion approximation. Such an approximation is an essential part in reactor theory since it is adequately uncomplicated to enable detailed calculations. The model is sufficiently realistic to provide many more significant concepts arising in the nuclear reactor analysis. The next chapter will focus on the establishment of an approximate representation of the neutron transport equation, which is much easier to work with.

References

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

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

CHAPTER 3
MULTIGROUP METHOD AND CRITICALITY CALCULATION

Presume that one wishes to sustain a stable fission chain reaction and thereby reach a steady rate of fission energy production. Therefore, one must design a nuclear reactor in such a way that the rates of neutron absorption and leakage are compensated by the rate of fission neutron production. In this chapter, the multigroup method will be introduced to simplify and reduce the general transport equation into the multigroup equations. Then, readers will be presented with the most vital calculation in reactor physics, that is, the criticality calculation. At this point, the criticality calculation of a nuclear system will allow us to evaluate the stability of the fission chain reaction. At the end of this chapter, the alternative form of the time-independent multigroup diffusion equations will be established. Also, these equations form a matrix expression which is known as the k-eigenvalue equation and it will be used to accomplish a criticality calculation. Briefly, a criticality calculation at first gathers all parameters related to the reactor design, nuclear fuel properties and the reactor core configuration. At the end of the calculation, nuclear engineers will be able to quantitatively estimate the stability of the reactor using the multiplication factor, k.

3.1 Interaction Probability

Consider a monodirectional and monoenergetic neutron beam with energy E is targeted along the direction $\widehat{\boldsymbol{\Omega}}$ towards a material of thickness R, with an initial beam flux magnitude ϕ_0. Also, the region outside the material is considered vacuum, and a neutron detector is placed at some distance behind the material. Hence, every neutron that interacts in the material is lost from the beam and leaving only neutrons that do not interact to reach the detector.

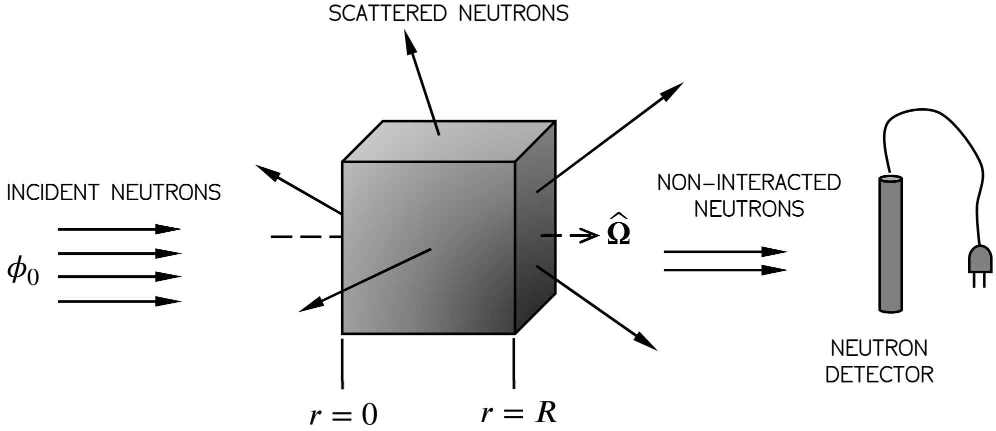


Figure 3.1: Measurement of neutrons that have not interacted with the target.

At this instance, it is convenient to define $\phi(r)$ as the flux of the neutron beam after penetrating the distance r into the material. After travelling at an additional distance dr, the decrease of the flux is given by,

$$-d\phi(r) = \Sigma_{\mathrm{t}}(r)\, \phi(r)\, dr \tag{3.1}$$

Assuming that the target material is homogeneous, and henceforth, the total neutron cross section is constant at all locations within the material,

$$-d\phi(r) = \Sigma_{\mathrm{t}}\ \phi(r)\, dr \tag{3.2}$$

The probability of neutrons that will subsequently interact in the next additional distance dr after penetrating the material at the distance r is given by P_1,

$$P_1(r) = \frac{d\phi(r)}{\phi(r)} \tag{3.3}$$

Hence, rearranging Eq. (3.2) and substituting into Eq. (3.3) yields,

$$P_1(r) = -\Sigma_{\mathrm{t}}\ dr \tag{3.4}$$

Solving the separable differential equation in Eq. (3.2) yields,

$$\phi(r) = \phi_0 e^{-\Sigma_{\mathrm{t}} r} \tag{3.5}$$

The probability of the neutrons to penetrate the material until the distance r without any interactions is given by P_2,

$$P_2(r) = \frac{\phi(r)}{\phi_0} = e^{-\Sigma_t r} \tag{3.6}$$

Next, the probability of a neutron to have its first interaction in the distance dr, $P(r)$, is given by,

$$P(r) = P_1(r) \times P_2(r) = \Sigma_t e^{-\Sigma_t r}\, dr \tag{3.7}$$

Finally, the probability density function of the distance to the next interaction, $p(r)$, is given by,

$$p(r) = \Sigma_t e^{-\Sigma_t r} \tag{3.8}$$

3.2 Fick's Law

As pointed out in the previous section, solving the neutron transport equation is rather cumbersome. However, it is possible to impose certain conditions so that the neutron flux, ϕ, and current, **J**, are related in a simple way. Thus, the relation can be used to simplify the complicated form of the transport equation which will ease the process of solving the equation. Now, it is practical to derive the relationship between ϕ and **J** by calculating the neutron current density at any location within a medium.

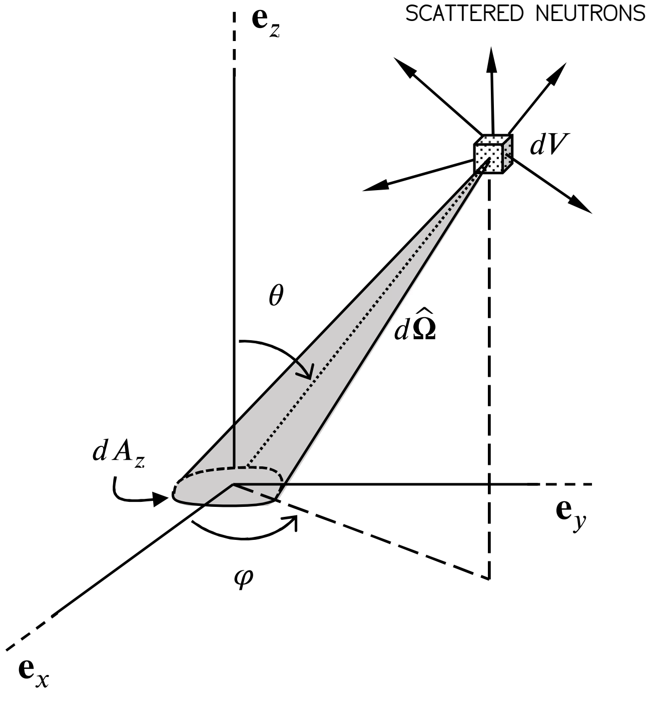


Figure 3.2: An illustration for deriving Fick's law.

The location at which the neutron current density is calculated is defined to be the origin of a coordinate system shown in Fig. 3.2. Most importantly, the three vector components of **J** must be evaluated. It is convenient to begin with J_z and consider the rate at which neutrons flow through the area dA_z spanning over the xy-plane at the origin. Note also that there are no neutron sources exist in the medium. Correspondingly, all neutrons that pass through dA_z have just arrived from a scattering collision. Equally important, all scattering collisions that occur above the xy-plane will always cause some of the outgoing neutrons to flow downward through dA_z and vice versa.

At this point, it is useful to begin calculating the number of scattering collisions that occur per second in the volume element dV at point $\mathbf{r}$. By using the theorems explained in Section 2.4, one can deduce that this number is equal to

$$\Sigma_{\mathrm{s}}\phi(\mathbf{r})\,dV \tag{3.9}$$

Based on the assumption that the scattering process is isotropic in the laboratory frame of reference, the fraction of outgoing neutrons that are scattered in the direction of dA_z is given by

$$\frac{dA_z\cos\theta}{4\pi r^2}\times\Sigma_{\mathrm{s}}\phi(\mathbf{r})\,dV \tag{3.10}$$

Inescapably, a fraction of these scattered neutrons will not succeed in reaching dA_z where they are absorbed en route. As discussed in the previous section, the fraction of neutrons which able to reach dA_z per second is given by,

$$\frac{dA_z\cos\theta}{4\pi r^2}\times\Sigma_{\mathrm{s}}\phi(\mathbf{r})\,dV\times e^{-\Sigma_{\mathrm{t}}r} \tag{3.11}$$

At this instance, the z-component of neutron current density, J_z is just the integral of the fraction given by Eq. (3.11) divided by dS_z over the entire volume. With dV written in spherical coordinates, where, $dV = r^2\sin\theta\,dr\,d\theta\,d\varphi$, thus,

$$J_z=\frac{\Sigma_{\mathrm{s}}}{4\pi}\int_{r=0}^{\infty}\int_{\theta=0}^{\pi}\int_{\varphi=0}^{2\pi}\phi(\mathbf{r})\,e^{-\Sigma_{\mathrm{t}}r}\cos\theta\sin\theta\;d\varphi\;d\theta\;dr \tag{3.12}$$

At this point of calculation, much of the attention was given to the z-component of $\mathbf{J}$, where it is expressed in terms of the equivalent spherical coordinates. Most importantly, the orientation of the spherical coordinate system is symmetric, thus, the form of the right-hand side of Eq. (3.12) also applies to the x-component and y-component of $\mathbf{J}$. Suppose that the component of interest is denoted with $\mu\in\{x,y,z\}$. In general, the three components of $\mathbf{J}$ in a Cartesian coordinate system can be written in the form of,

$$J_\mu=\frac{\Sigma_{\mathrm{s}}}{4\pi}\int_{r=0}^{\infty}\int_{\theta_\mu=0}^{\pi}\int_{\varphi_\mu=0}^{2\pi}\phi(\mathbf{r}_\mu)\,e^{-\Sigma_{\mathrm{t}}r}\cos\theta_\mu\;\sin\theta_\mu\,d\varphi_\mu d\theta_\mu dr \tag{3.13}$$

where θ_μ and φ_μ are the polar angle and the azimuthal angle respectively with the zenith of the spherical coordinate system lies along the μ-axis of the Cartesian coordinate system.

Alas, the integral in Eq. (3.13) cannot be evaluated because $\phi(\mathbf{r}_\mu)$ is unknown. However, if $\phi(\mathbf{r}_\mu)$ varies slowly with positions, it can be expressed in Taylor's series. For the case where z-axis as the zenith of the spherical coordinate system ($\mu \equiv z$), the Taylor's expansion of $\phi(\mathbf{r})$ is given by,

$$\phi(\mathbf{r}) = \phi_0 + x\left(\frac{\partial\phi}{\partial x}\right)_0 + y\left(\frac{\partial\phi}{\partial y}\right)_0 + z\left(\frac{\partial\phi}{\partial z}\right)_0 \cdots \tag{3.14}$$

with $\mathbf{r} = x\mathbf{e}_x + y\mathbf{e}_y + z\mathbf{e}_z$, $x = r\sin\theta\cos\varphi$, $y = r\sin\theta\sin\varphi$ and $z = r\cos\theta$. From Eq. (3.14), it is now clear that the similar form can be used for the other two components of $\mathbf{J}$, i.e. $\mu \equiv x$ and $\mu \equiv y$. In general, it is useful to write $\phi(\mathbf{r}_\mu)$ as,

$$\phi(\mathbf{r}_\mu) = \begin{cases} \phi_0 + r\cos\varphi\left(\frac{\partial\phi}{\partial x}\right)_0 + r\sin\theta\cos\varphi\left(\frac{\partial\phi}{\partial y}\right)_0 + r\sin\theta\sin\varphi\left(\frac{\partial\phi}{\partial z}\right)_0, \mu \equiv x \\ \phi_0 + r\cos\varphi\left(\frac{\partial\phi}{\partial y}\right)_0 + r\sin\theta\cos\varphi\left(\frac{\partial\phi}{\partial z}\right)_0 + r\sin\theta\sin\varphi\left(\frac{\partial\phi}{\partial x}\right)_0, \mu \equiv y \\ \phi_0 + r\cos\varphi\left(\frac{\partial\phi}{\partial z}\right)_0 + r\sin\theta\cos\varphi\left(\frac{\partial\phi}{\partial x}\right)_0 + r\sin\theta\sin\varphi\left(\frac{\partial\phi}{\partial y}\right)_0, \mu \equiv z \end{cases} \tag{3.15}$$

Next, Eq. (3.15) is substituted in Eq. (3.14), and of course, it is found that the terms containing $\cos\varphi$ and $\sin\varphi$ will integrate to zero. Thus, J_μ reduces into,

$$\begin{aligned} J_\mu = \frac{\Sigma_\mathrm{s}}{4\pi}\int_{r=0}^{\infty}\int_{\theta_\mu=0}^{\pi}\int_{\varphi_\mu=0}^{2\pi} \Bigg(&\phi_0 \\ &+ r\cos\varphi_\mu\left(\frac{\partial\phi}{\partial\mu}\right)_0\Bigg)\, e^{-\Sigma_\mathrm{t} r}\cos\theta_\mu\ \sin\theta_\mu\, d\varphi_\mu d\theta_\mu dr \end{aligned} \tag{3.16}$$

Hence, evaluating the definite integrals in Eq. (3.16) yields,

$$J_\mu = -\frac{\Sigma_\mathrm{s}}{3\Sigma_\mathrm{t}^2}\left(\frac{\partial\phi}{\partial\mu}\right)_0. \tag{3.17}$$

Finally, the vector $\mathbf{J}$ is now,

$$\begin{aligned} \mathbf{J} &= J_x\mathbf{e}_x + J_y\mathbf{e}_y + J_z\mathbf{e}_z \\ &= -\frac{\Sigma_\mathrm{s}}{3\Sigma_\mathrm{t}^2}\boldsymbol{\nabla}\phi. \end{aligned} \tag{3.18}$$

In summary, Eq. (3.18) is known as *Fick's law*, which states that the current density vector is proportional to the negative gradient of the flux. It is useful to define the proportionality constant $\boldsymbol{D}$, which is known as the diffusion coefficient where,

$$D = \frac{\Sigma_s}{3\Sigma_t^2} \tag{3.19}$$

With this notation, Fick's law can be written as,

$$\mathbf{J} = -D\,\nabla\phi. \tag{3.20}$$

It should be noticed that the derivation of Fick's law in this section is just a pre-requisite of the construction of good knowledge and foundation for the research and development of a multigroup Monte Carlo method which will be discussed further in Chapter 4. Readers may find that in the Monte Carlo method, the transport behaviour of neutrons is determined stochastically without the use of Fick's law, as in Eq. (3.20). Here, the diffusion of neutrons within a medium is reproduced via the computer simulation of neutron random walks.

So far, much of the attention has been given to the derivation of Fick's law. However, it is crucial to re-examine the conditions in which this law can be expected to be valid. There are a few assumptions made when deriving Fick's law in reactor calculation. However, most of these assumptions can be relaxed because of some reasons that apply to nuclear reactor conditions. A detailed discussion of these assumptions is given by (Lamarsh & Baratta, 1955). However, the diffusion approximation has to be strictly performed under a number of important assumptions:

(a) The neutron diffusing medium must be infinite, to allow the integration of Eq. (3.13) over the entire space. Fortunately, the $e^{-\Sigma_t r}$ term in Eq. (3.13) dies off quickly with distance. Consequently, neutrons streaming from farther distances from the point where **J** is calculated give fewer contributions to **J**, and their effect on the computation of **J** is negligible. Thus, Fick's law is still valid in the interior of a reactor, since the size of a reactor is exceptionally large compared to the diffusion distances of these neutrons. A problem may arise for the case near the outer surface of a reactor because there exists an abrupt change of the material cross section, i.e. between a dense medium (the interior of the reactor) and a vacuous medium (the outer region of the reactor). However, such an issue can be rectified by using a special treatment which is known as the *reciprocal logarithmic derivative*. A detailed reference of the special treatment is given by (Lamarsh & Baratta, 1955) and (Duderstadt & Hamilton, 1976). It should be noted that this issue is not applicable to Monte Carlo method since the neutron transport behaviour is simulated and purely based on statistical observations.

(b) The neutron flux is a slowly varying function of position; the Taylor series of the neutron flux, $\phi(\mathbf{r})$, given by Eq. (3.15) is only expanded up to the first order. Hence, a rapid change of the neutron flux may compromise the accuracy of the calculation, since it can cause the higher spatial derivative terms of the Taylor expansion of $\phi(\mathbf{r})$ to be significant and cannot be neglected.

(c) The neutron flux is a slowly varying function of time; In Eq. (3.13), the neutron flux is assumed to be independent of time. Unfortunately, a neutron needs time to travel from the collision site to the location where **J** is calculated. As previously discussed, only neutrons streaming from closer distances from the point where $\mathbf{J}$ is calculated significantly contributes to $\mathbf{J}$. So, the time taken for neutrons to travel at these close distances is negligible since neutrons travel at high speed. In a nuclear reactor, even the slowest neutrons travel at 1000m/s, thus this assumption can be relaxed.

3.3 Multigroup Method

It is well understood that neutron energies typically encountered in a reactor span a range from 10^{-3} eV to 10^{7} eV. Also, neutron cross sections depend sensitively on energy over most of this range (Duderstadt & Hamilton, 1976). In this section, the continuous neutron energy dependence will be suppressed by assuming that all neutrons can be characterized using a few discrete energy groups. Precisely, the continuous neutron energy range is divided into several energy group intervals. Plus, the energy boundaries of each group interval are defined prior to the neutronic calculation. Subsequently, each individual neutron energy group is assigned to a differential equation known as the *group equation*. Next, a sequence of one-group calculations for each successive energy group is performed.

In the multigroup method, the entire continuous neutron energy within a specific thermal nuclear reactor is divided into G energy groups. Suitably, the highest energy that can be attained by a neutron in a nuclear system is $E_0 = \infty$. A neutron group is assigned with an index, $g \in \{1,2,3, \dots, g-1, g, g+1, \dots, G-1, G\}$ where it has a predefined neutron energy range covering from E_g to E_{g-1}, where $E_{g-1} > E_g$. To derive the diffusion equation for the energy group g, several group constants for the energy group need to be defined.

Right now, it is favourable to simplify the neutron transport equation given by Eq. (2.30) using Fick's law and the multigroup method. It is beneficial to go through each term of the neutron transport equation and simplify each of these terms accordingly. Consider a system, typically an infinitesimal volume dV of a homogeneous fissile material. To begin with, the energy variable in the neutron transport equation is eliminated by integrating Eq. (2.30) over the g^{th} energy group defined within the energy range $\left(E_g, E_{g-1}\right)$. Plus, the neutron direction, $\widehat{\mathbf{\Omega}}$, in the neutron transport equation can be eliminated by integrating the transport equation over the entire neutron directions, i.e. $[0,4\pi)$. Thus,

$$\begin{aligned}
&\underbrace{\int_{4\pi}\int_{E_g}^{E_{g-1}} \frac{1}{v}\frac{\partial}{\partial t}\psi(\mathbf{r},E,\hat{\boldsymbol{\Omega}},t)\,dE\,d\hat{\boldsymbol{\Omega}}}_{\text{Time rate of change}} + \underbrace{\int_{4\pi}\int_{E_g}^{E_{g-1}} \hat{\boldsymbol{\Omega}}\cdot\boldsymbol{\nabla}\psi(\mathbf{r},E,\hat{\boldsymbol{\Omega}},t)\,dE\,d\hat{\boldsymbol{\Omega}}}_{\text{Streaming term}} \\
&+ \underbrace{\int_{4\pi}\int_{E_g}^{E_{g-1}} \Sigma_t(\mathbf{r},E)\psi(\mathbf{r},E,\hat{\boldsymbol{\Omega}},t)\,dE\,d\hat{\boldsymbol{\Omega}}}_{\text{Disappearance term}} \\
&= \underbrace{\int_{4\pi}\int_{E_g}^{E_{g-1}}\int_{4\pi}\int_{0}^{\infty} \Sigma_s(\mathbf{r},E'\rightarrow E,\hat{\boldsymbol{\Omega}}'\rightarrow\hat{\boldsymbol{\Omega}})\psi(\mathbf{r},E',\hat{\boldsymbol{\Omega}}',t)\,dE'\,d\hat{\boldsymbol{\Omega}}'\,dE\,d\hat{\boldsymbol{\Omega}}}_{\text{In-scattering term}} \\
&+ \underbrace{\int_{4\pi}\int_{E_g}^{E_{g-1}}\int_{4\pi}\int_{0}^{\infty} \frac{\chi(E)}{4\pi}\nu(E')\Sigma_f(\mathbf{r},E')\psi(\mathbf{r},E',\hat{\boldsymbol{\Omega}}',t)\,dE'\,d\hat{\boldsymbol{\Omega}}'\,dE\,d\hat{\boldsymbol{\Omega}}}_{\text{Fission term}}
\end{aligned} \tag{3.21}$$

In addition, all integrations with respect to the neutron direction, $\hat{\boldsymbol{\Omega}}$, over the entire solid angle, 4π, will relax our focus on dealing with $\hat{\boldsymbol{\Omega}}$. Remark that the study of the behaviour of neutrons regardless of its direction is in focus instead.

Subsequently, several formal definitions of multigroup constants will be introduced. At this instance, the group flux, ϕ_g, is defined as the flux integrated over the entire energy range defined for the group, g, i.e. $\left(E_g, E_{g-1}\right)$

$$\phi_g(\mathbf{r},t) = \int_{4\pi}\int_{E_g}^{E_{g-1}} \psi(\mathbf{r},E,\hat{\boldsymbol{\Omega}},t)\,dE\,d\hat{\boldsymbol{\Omega}}. \tag{3.22}$$

The *time rate of change term* in Eq. (3.21) can be further simplified into,

$$\begin{aligned}
\left\{\begin{matrix}\text{Time rate}\\ \text{of change}\\ \text{of neutrons}\end{matrix}\right\} &= \int_{4\pi}\int_{E_g}^{E_{g-1}} \frac{1}{v}\frac{\partial}{\partial t}\psi(\mathbf{r},E,\hat{\boldsymbol{\Omega}},t)\,dE\,d\hat{\boldsymbol{\Omega}} \\
&= \frac{1}{v_g}\frac{\partial\phi_g(\mathbf{r},t)}{\partial t}
\end{aligned} \tag{3.23}$$

where the inverse neutron speed characterizing group g is given by

$$\frac{1}{v_g} = \frac{1}{\phi_g(\mathbf{r},t)}\int_{4\pi}\int_{E_g}^{E_{g-1}} \frac{1}{v}\psi(\mathbf{r},E,\hat{\boldsymbol{\Omega}},t)\,dE\,d\hat{\boldsymbol{\Omega}} \tag{3.24}$$

Next, the *streaming term* in Eq. (3.21) can be further simplified in terms of the angular neutron current density, $\mathbf{j}(r, E, \hat{\boldsymbol{\Omega}}, t)$, via Eqs. (2.5), (2.6) and (2.7),

$$
\begin{aligned}
&\int_{4\pi}\int_{E_g}^{E_{g-1}} \hat{\mathbf{\Omega}} \cdot \nabla\psi(\mathbf{r}, E, \hat{\mathbf{\Omega}}, t)\, dE\, d\hat{\mathbf{\Omega}} \\
&\qquad = \int_{4\pi}\int_{E_g}^{E_{g-1}} \nabla \cdot \hat{\mathbf{\Omega}}\psi(\mathbf{r}, E, \hat{\mathbf{\Omega}}, t)\, dE\, d\hat{\mathbf{\Omega}} \\
&\qquad = \int_{4\pi}\int_{E_g}^{E_{g-1}} \nabla \cdot \mathbf{j}(\mathbf{r}, E, \hat{\mathbf{\Omega}}, t)\, dE\, d\hat{\mathbf{\Omega}} \\
&\qquad = \int_{E_g}^{E_{g-1}} \nabla \cdot \mathbf{J}(\mathbf{r}, E, t)\, dE
\end{aligned}
\tag{3.25}
$$

Since Fick’s law establishes the relationship between the flux and the current density, $\mathbf{J}(\mathbf{r}, E, t)$ can be approximated by substituting Eq. (3.20) into Eq. (3.25), giving,

$$
\begin{aligned}
\int_{E_g}^{E_{g-1}} \nabla \cdot \mathbf{J}(\mathbf{r}, E, t)\, dE &= \int_{E_g}^{E_{g-1}} \nabla \cdot \left(-D(\mathbf{r}, E)\nabla\phi(\mathbf{r}, E, t)\right) dE \\
&= -\nabla \cdot \int_{E_g}^{E_{g-1}} D(\mathbf{r}, E)\, \nabla\phi(\mathbf{r}, E, t)\, dE
\end{aligned}
\tag{3.26}
$$

With some simple mathematical manoeuvres, the group diffusion coefficient, $D_g(\mathbf{r}, t)$, is defined as

$$
D_g(\mathbf{r}, t) = \frac{\displaystyle\int_{E_g}^{E_{g-1}} D(\mathbf{r}, E)\nabla\phi(\mathbf{r}, E, t)\, dE}{\displaystyle\int_{E_g}^{E_{g-1}} \nabla\phi(\mathbf{r}, E, t)\, dE}
\tag{3.27}
$$

The group diffusion coefficient as defined in Eq. (3.27) permits us to write the multigroup net leakage term as,

$$
\begin{Bmatrix} \text{Net} \\ \text{leakage} \end{Bmatrix} = -\nabla \cdot D_g(\mathbf{r}, t)\, \nabla\phi_g(\mathbf{r}, t)
\tag{3.28}
$$

Subsequently, the total macroscopic group cross section $\Sigma_{\mathrm{t}}^{g}(\mathbf{r}, t)$ is defined as

$$
\begin{aligned}
\Sigma_{\mathrm{t}}^{g}(\mathbf{r}, t) &= \frac{1}{\phi_g(\mathbf{r}, t)} \int_{4\pi}\int_{E_g}^{E_{g-1}} \Sigma_{\mathrm{t}}(\mathbf{r}, E)\psi(\mathbf{r}, E, \hat{\mathbf{\Omega}}, t)\, dE\;\, d\hat{\mathbf{\Omega}} \\
&= \frac{1}{\phi_g(\mathbf{r}, t)} \int_{E_g}^{E_{g-1}} \Sigma_t(\mathbf{r}, E)\phi(\mathbf{r}, E, t)\, dE
\end{aligned}
\tag{3.29}
$$

Therefore, the rate at which neutrons from energy group g interacts with the homogeneous fissile material and disappear from the energy group g (also known as the disappearance term) can be simply written as,

$$\left\{\begin{array}{c}\text{Group}\\ \text{disappearance}\\ \text{term}\end{array}\right\} = \Sigma_{\mathrm{t}}^{g}(\mathbf{r},t)\ \phi_g(\mathbf{r},t) \tag{3.30}$$

Recall that some neutrons from other energy groups, g', will experience scattering collision with the homogeneous fissile material. Such a scattering collision changes the neutron energy, which also implies the change of the energy group. Consequently, some of these scattered neutrons will fall into group g, and this phenomenon is often coined as in-scattering. Of course, scattering collision can also remove neutrons from energy group g, which is termed as out-scattering. However, it is important to stress that the rate of out-scattering reaction has been included in Eq. (3.30) since it is equivalent to the removal of neutrons from energy group g. To this point, the *macroscopic transfer group cross section* is defined as

$$\begin{aligned}\Sigma_{\mathrm{s}}^{g'\to g}(\mathbf{r},t) &= \frac{1}{\phi_{g'}(\mathbf{r},t)}\int_{4\pi}\int_{E_g}^{E_{g-1}}\int_{4\pi}\int_{E_{g'}}^{E_{g'-1}}\Sigma_{\mathrm{s}}(\mathbf{r},E'\to E,\widehat{\mathbf{\Omega}}'\\ &\quad\to\widehat{\mathbf{\Omega}})\Psi(\mathbf{r},E',\widehat{\mathbf{\Omega}}',t)\,dE'\,d\widehat{\mathbf{\Omega}}'\,dE\,d\widehat{\mathbf{\Omega}}\\ &= \frac{1}{\phi_{g'}(\mathbf{r},t)}\int_{E_{g'}}^{E_{g'-1}}\Sigma_{\mathrm{s}}(\mathbf{r},E'\to E)\phi(\mathbf{r},E',t)\,dE'\end{aligned} \tag{3.31}$$

where $\phi^{g'}(\mathbf{r},t)$ is defined similarly according to Eq. (3.22). Be aware of the special notation, i.e. $g'\to g$, where it indicates the transfer of neutrons from group g' into group g. The in-scattering rate from energy group g' into g can be written as,

$$\left\{\begin{array}{c}\text{Group}\\ \text{in-scattering}\\ \text{term}\end{array}\right\} = \Sigma_{\mathrm{s}}^{g'\to g}(\mathbf{r},t)\ \phi_{g'}(\mathbf{r},t) \tag{3.32}$$

Of course, the contribution of the neutron in group g is caused by the in-scattering of neutrons from all energy groups into the energy group g. Such a total contribution can be represented as the summation of the right-hand-side of Eq. (3.32) over the entire incident neutron energy groups, $g' = \{1,2,3,\dots,G\}$. Thus, the total in-scattering rate can be written as,

$$\begin{aligned}\Sigma_{\mathrm{s}}^{1\to g}(\mathbf{r})\ \phi_1(\mathbf{r},t) + \Sigma_{\mathrm{s}}^{2\to g}(\mathbf{r})\ \phi_2(\mathbf{r},t) + \cdots + \Sigma_{\mathrm{s}}^{G\to g}(\mathbf{r})\ \phi_G(\mathbf{r},t)\\ = \sum_{g'=1}^{G}\Sigma_{\mathrm{s}}^{g'\to g}(\mathbf{r})\ \phi_{g'}(\mathbf{r},t)\end{aligned} \tag{3.33}$$

So far, intense work on deriving the group flux, multigroup inverse neutron speed, multigroup streaming term, multigroup disappearance term and multigroup in-scattering has been completed. In a fissile medium, one must include a mathematical term that accounts for the neutron production due to fission reaction in the neutron transport equation. Originally, such a mathematical term is the fission term which is given by Eq. (2.28). The multigroup form of the fission term can be derived by first considering the production rate of fission neutrons with energy E and direction $\widehat{\mathbf{\Omega}}$ due to fission reactions induced by an incident neutron with energy E' and direction $\widehat{\mathbf{\Omega}}'$. This is given by,

$$S_{\mathrm{f}}(\mathbf{r},E,\widehat{\mathbf{\Omega}},t)=\frac{\chi(E)}{4\pi}\int_{4\pi}\nu(E')\Sigma_{\mathrm{f}}(\mathbf{r},E')\psi(\mathbf{r},E',\widehat{\mathbf{\Omega}}',t)\,d\widehat{\mathbf{\Omega}}' \tag{3.34}$$

In the above equation, the integration with respect to $d\widehat{\mathbf{\Omega}}'$ over the entire solid angle, 4π, simply indicates that $S_{\mathrm{f}}(\mathbf{r},E,\widehat{\mathbf{\Omega}},t)$ considers all possible incident neutron directions. Bear in mind that it is assumed that the fission neutrons are emitted isotropically. Put differently, the probability of ejecting fission neutrons isotropically via fission caused by an incident neutron of any direction is equal. Suppose that isotropic fission is implied, the total birth rate of fission neutrons in energy group g cause by fission reactions induced by incident neutron regardless of their energy and direction is given by,

$$\begin{aligned}S_{\mathrm{f}}(\mathbf{r},E,t)&=\frac{\chi(E)}{4\pi}\int_{4\pi}\int_{4\pi}\nu(E')\Sigma_{\mathrm{f}}(\mathbf{r},E')\psi(\mathbf{r},E',\widehat{\mathbf{\Omega}}',t)\,d\widehat{\mathbf{\Omega}}'\;d\widehat{\mathbf{\Omega}}\\&=\frac{\chi(E)}{4\pi}\nu(E')\Sigma_{\mathrm{f}}(\mathbf{r},E')\int_{4\pi}d\widehat{\mathbf{\Omega}}\;\int_{4\pi}\psi(\mathbf{r},E',\widehat{\mathbf{\Omega}}',t)\,d\widehat{\mathbf{\Omega}}'\\&=\chi(E)\nu(E')\Sigma_{\mathrm{f}}(\mathbf{r},E')\phi(\mathbf{r},E',t)\end{aligned} \tag{3.35}$$

where,

$$\int_{4\pi}d\widehat{\mathbf{\Omega}}\;=4\pi \tag{3.36}$$

The multigroup form of $S_{\mathrm{f}}(\mathbf{r},E,t)$ can be derived by integrating it with respect to the incident neutron energy, E' and the outgoing fission neutron energy E over the entire energy range characterizing group g' and g respectively,

$$\begin{aligned}&\int_{E_g}^{E_{g-1}}dE\,S_{\mathrm{f}}(\mathbf{r},E,t)\\&\qquad=\int_{E_g}^{E_{g-1}}\int_{E_{g'}}^{E_{g'-1}}\chi(E)\nu(E')\Sigma_{\mathrm{f}}(\mathbf{r},E')\phi(\mathbf{r},E',t)\,dE'\,dE\end{aligned} \tag{3.37}$$

To further simplify the group fission term given in Eq. (3.37), several group constants of the fission term are defined. The group fission spectrum is defined as,

$$\chi_g=\int_{E_g}^{E_{g-1}}\chi(E)dE \tag{3.38}$$

and the group neutron production cross section is defined as,

$$\nu_{g'}\Sigma_{\mathrm{f}}^{g'}=\frac{1}{\phi_{g'}(\mathbf{r},t)}\int_{E_{g'}}^{E_{g'-1}}\nu(E')\Sigma_{\mathrm{f}}(\mathbf{r},E')\phi(\mathbf{r},E',t)\,dE' \tag{3.39}$$

Consequently, Eq. (3.37) can be simplified into,

$$\int_{E_g}^{E_{g-1}} S_\mathrm{f}(\mathbf{r}, E, t)\, dE = \chi_g \nu_g \Sigma_\mathrm{f}^{g'}(\mathbf{r})\phi_{g'}(\mathbf{r}, t) \tag{3.40}$$

Correspondingly, the total production of fission neutrons in group g can be obtained by considering the sum of the fission neutrons production induced by incident neutrons from all groups. Thus, using Eq. (3.40), the group fission term can be written as:

$$\begin{aligned} \left\{\begin{array}{c}\text{Group}\\ \text{fission}\\ \text{term}\end{array}\right\} &= \chi_g \nu_1 \Sigma_\mathrm{f}^1(\mathbf{r})\phi_1(\mathbf{r}, t) + \chi_g \nu_2 \Sigma_\mathrm{f}^2(\mathbf{r})\phi_2(\mathbf{r}, t) + \cdots \\ &\quad + \chi_g \nu_G \Sigma_\mathrm{f}^G(\mathbf{r})\phi_G(\mathbf{r}, t) \\ &= \chi_g \sum_{g'=1}^{G} \nu_{g'} \Sigma_\mathrm{f}^{g'}(\mathbf{r})\phi_{g'}(\mathbf{r}, t) \end{aligned} \tag{3.41}$$

Perhaps, the most direct way of deriving the multigroup neutron diffusion equation is to apply the concept of neutron balance to a given energy group. Base on this notion, the mechanisms in which neutrons enter and leave the energy group g are balanced. Mathematically, the neutron balance equation can be written as,

$$\begin{aligned} \boxed{\begin{array}{c}\text{Time rate}\\ \text{of change of}\\ \text{neutrons in}\\ \text{group } g\end{array}} = -\boxed{\begin{array}{c}\text{Change}\\ \text{due to}\\ \text{leakage}\\ \text{(Leakage term)}\end{array}} - \boxed{\begin{array}{c}\text{Disappearance}\\ \text{of neutrons}\\ \text{from group } g\\ \text{(Disappearance term)}\end{array}} \\ + \boxed{\begin{array}{c}\text{Neutrons}\\ \text{scattering into}\\ \text{group g}\\ \text{(In-scattering term)}\end{array}} + \boxed{\begin{array}{c}\text{Appearance of}\\ \text{fission neutrons}\\ \text{in group } g\\ \text{(Fission Term)}\end{array}} \end{aligned} \tag{3.42}$$

Now, Eqs. (3.23), (3.28), (3.30), (3.32) and (3.41) are substituted into Eq. (3.42), which will then yield the general form of the *multigroup equations*,

$$\begin{aligned} \frac{1}{v_g}\frac{\partial \phi_g(\mathbf{r}, t)}{\partial t} &= -\boldsymbol{\nabla} \cdot D_g(\mathbf{r}, t)\, \boldsymbol{\nabla}\phi_g(\mathbf{r}, t) - \Sigma_\mathrm{t}^g(\mathbf{r}, t)\, \phi_g(\mathbf{r}, t) \\ &\quad + \sum_{g'=1}^{G} \Sigma_\mathrm{s}^{g' \to g}(\mathbf{r}, t)\, \phi_{g'}(\mathbf{r}, t) + \chi_g \sum_{g'=1}^{G} \nu_{g'} \Sigma_\mathrm{f}^{g'}(\mathbf{r}, t)\phi_{g'}(\mathbf{r}, t) \\ &\qquad g = 1,2,3, \dots, G \end{aligned} \tag{3.43}$$

Several remarks on these equations are essential and need be clearly stated before proceeding to the next step of reactor calculation. The general form of multigroup diffusion equations given in Eq. (3.43) are reasonably exact since the following group constants were introduced,

$$\frac{1}{v_g}, D_g, \Sigma_{\mathrm{t}}^{g}, \Sigma_{\mathrm{s}}^{g'\to g}, \chi_g, \nu_{g'}, \Sigma_{\mathrm{f}}^{g'}$$

Nevertheless, these group constants are still undetermined and need to be calculated. From the definitions of the group constants given by Eqs. (3.24), (3.27), (3.29), (3.31), (3.38) and (3.39), it is apparent that the flux $\phi(\mathbf{r}, E, t)$ must be known prior to the calculation of the group constants. However, $\phi(\mathbf{r}, E, t)$ is just the function that needs to be solved in the first place by discretizing the neutron energy into groups. This seems that the development of the multigroup method has been quite circular. Note also that the group constants are also dependent on the space and time which makes the problem even more cumbersome. Indeed, the group constants will be rigorously constant only if $\phi(\mathbf{r}, E, t)$ is of the separable form,

$$\phi(\mathbf{r}, E, t) = F(\mathbf{r}, t)G(E) \tag{3.44}$$

For such a scenario, the group constants will reduce to group averages over the flux energy spectrum $G(E)$. Alas, this is not the case in a nuclear reactor where the flux is usually not separable in energy. To rectify the problem, one may attempt to guess and approximate the intragroup flux characterizing each neutron energy group, i.e. $\phi(\mathbf{r}, E, t) \cong \tilde{\phi}(\mathbf{r}, E, t)$. Hence, the group constants can be calculated by replacing $\phi(\mathbf{r}, E, t)$ with $\tilde{\phi}(\mathbf{r}, E, t)$ in their corresponding mathematical definition. For example, the group total neutron cross section can be calculated as averages over these approximate intragroup fluxes,

$$\Sigma_{\mathrm{t}}^{g}(\mathbf{r}, t) = \frac{1}{\phi_g(\mathbf{r}, t)} \int_{E_g}^{E_{g-1}} \Sigma_{\mathrm{t}}(\mathbf{r}, E)\tilde{\phi}(\mathbf{r}, E, t)\, dE \tag{3.45}$$

with

$$\phi_g(\mathbf{r}, t) = \int_{E_g}^{E_{g-1}} \tilde{\phi}(\mathbf{r}, E, t)\, dE. \tag{3.46}$$

In the next section, the strategy of approximating the flux $\phi(\mathbf{r}, E, t)$ will be discussed in detail, and thus, the group constants can be directly calculated.

3.4 Fine Group Constants

Recall that in the previous section, intense work has been accomplished to simplify the neutron transport equation via multigroup diffusion method. Along the simplification process, several group constants were introduced. However, these group constants are still dependent on space, energy and time because their calculation requires the flux, $\phi(\mathbf{r}, E, t)$, to be known beforehand. Unfortunately, the purpose of solving the transport equation is to compute the flux. Thus, the only way to rectify such issue is by first ignoring the space and time variation of the flux, i.e., $\phi(\mathbf{r}, E, t) \sim \phi(E)$. Then, $\phi(E)$ is approximated with the intragroup fluxes – a weighing flux as a function of neutron energy characterizing each of the neutron energy group

g. Above all, one will be able to do very good multigroup calculations with only a few neutron energy groups provided that a good guess of the shape of $\phi(E)$ for a nuclear system is made.

Since the nature of the neutron transport equation is separable in time, the flux $\phi(\mathbf{r}, E, t)$ can be written as

$$\phi(\mathbf{r}, E, t) = F(\mathbf{r}, E)\tau(t) \tag{3.47}$$

As a result, group constants can be averaged over time. In general, the mathematical definitions of the group constants can be generally written as,

$$\Sigma_g = \frac{\displaystyle\int_{E_g}^{E_{g-1}} K\phi(\mathbf{r}, E, t)\, dE}{\displaystyle\int_{E_g}^{E_{g-1}} \phi(\mathbf{r}, E, t)\, dE} \tag{3.48}$$

where K is the integral kernel which maps Eq. (3.48) into various type of group constants, i.e. $K \in \left\{\frac{1}{v(\mathbf{r},E)}, \Sigma_t(\mathbf{r}, E), \Sigma_s(\mathbf{r}, E' \to E), \nu\Sigma_f(\mathbf{r}, E), \right\}$. Suppose that the flux is separable in time, it is convenient to eliminate the time variable in our group constants calculation, since,

$$\begin{aligned} \Sigma_g &= \frac{\displaystyle\int_{E_g}^{E_{g-1}} KF(\mathbf{r}, E)\tau(t)\, dE}{\displaystyle\int_{E_g}^{E_{g-1}} F(\mathbf{r}, E)\tau(t)\, dE} \\ &= \frac{\displaystyle\int_{E_g}^{E_{g-1}} KF(\mathbf{r}, E)\, dE}{\displaystyle\int_{E_g}^{E_{g-1}} F(\mathbf{r}, E)\, dE} \end{aligned} \tag{3.49}$$

Also, from the definition of the group diffusion coefficient,

$$D_g = \frac{\displaystyle\int_{E_g}^{E_{g-1}} D(\mathbf{r}, E)\nabla^2 F(\mathbf{r}, E)\tau(t)\, dE}{\displaystyle\int_{E_g}^{E_{g-1}} \nabla^2 F(\mathbf{r}, E)\tau(t)\, dE} \tag{3.50}$$

Since Laplacian operator, ∇^2, does not operate on the time variable, the definition of the group diffusion coefficient can be written as

$$D_g = \frac{\int_{E_g}^{E_{g-1}} D(\mathbf{r}, E)\nabla^2 F(\mathbf{r}, E) dE}{\int_{E_g}^{E_{g-1}} \nabla^2 F(\mathbf{r}, E)\, dE} \tag{3.51}$$

At this point, it is clear that the group constants are only dependent on position and energy. In the actual reactor calculation practice, one usually works with from two to 20 neutron energy groups. Such few group calculations are only reliable with reasonably accurate estimates of the group constants. Again, accurate estimates of the group constants can only be achieved if the intragroup fluxes, $\phi(E)$, are accurately determined. The widely accepted strategy is to perform two multigroup calculations. In the first multigroup calculation, the spatial and time dependence is neglected or very crudely approximated, and a very finely structured multigroup calculation is performed to calculate the fine spectrum fluxes. The group constants for this fine spectrum calculation are usually pre-calculated and represented as a tabulated multigroup cross section data library. Furthermore, the group constants are averaged over each of the fine energy groups. An example of such a library is the IAEA WIMSD formatted data library. The IAEA WIMSD formatted data library offers 69 and 172 fine group structures for nuclear reactor calculations. Fig. 3.3 shows the total cross section of U-235 in 69 and 172 fine energy groups.

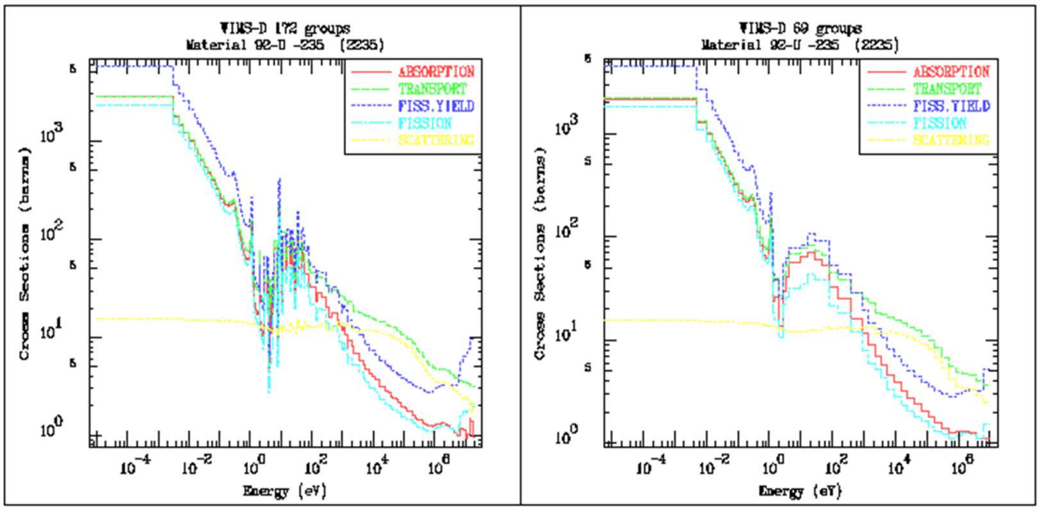


Figure 3.3: Multigroup neutron cross section plot for U-235. (Plot retrieved from International Atomic Energy Agency (IAEA) website, https://www-nds.iaea.org/wimsd/xsplots.htm)

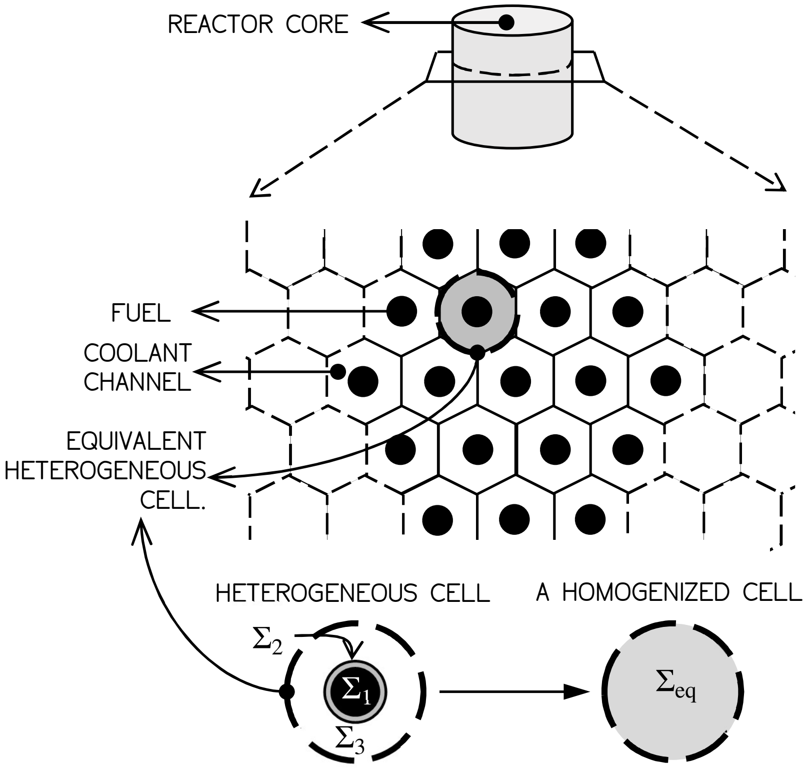


Figure 3.4: An equivalent heterogeneous cell consisting of three different materials, i.e. fissile fuel (Σ_1), fuel cladding (Σ_2) and the coolant channel (Σ_3), is homogenized. The heterogeneous cell is represented with an equivalent homogenized cell with the equivalent cross section Σ_{eq}.

At first sight, a reactor core can be constructed using its basic lattice structures which are known as unit cells. The geometry of each of these unit cells is defined such that the entire reactor core can be build using repetitions of these unit cells. For instance, the unit cells forming a reactor core are illustrated in Fig. 3.4. Essentially, the spatial dependence of the group constants reflects the fact that the geometry is heterogeneous which compose of discrete uniform material zones. Here, a heterogeneous unit cell is made up of a few separated regions of materials. To be more specific, a fuel cell consists of four different types of heterogeneous regions, i.e. the fuel meat, cladding material and the adjacent coolant channel. Note also that these heterogeneous regions are represented with specific group constants characterizing the material contained in each region.

To suppress the spatial dependency of the group constants, the heterogeneous unit cell is replaced with an equivalent homogenized unit cell. Consequently, a homogenized unit cell is defined such that the separated regions are "mixed" thus preserving the integral neutron behaviour. In this context, neutron behaviour is a sequence of events where various neutron-nucleus interactions take place. Thus, after the homogenization process, a set of homogenized group constants characterizing the homogenized material within the unit cell is obtained. A detailed procedure to calculate the homogenized group constants for the case of a TRIGA reactor core will be discussed intensively in Chapter 4.

3.5 Multiplication Factor

Recall that in the previous section, the main working principle of a nuclear reactor is by tapping the energy released in fission reaction via the use of fission chain reaction. Such an essential idea is based on the fact that a fission neutron produced by the previous fission reaction will eventually induce another fission reaction. Thus, the neutrons that induce the fission reactions are known as the chain carriers. In normal nuclear engineering practice, it is desirable to ensure a steady-state chain reaction, which simply means a chain reaction that does not grow or decay away with time. Consequently, a nuclear engineer must design a reactor that is capable of inducing one fission reaction per fission neutrons produced by the previous fission reaction. Plus, certain factors that cause the fission chain reaction to degenerate have to be considered. Specifically, the remaining neutrons produced by the previous fission reaction could either be absorbed by means of disappearance reactions or will leak and escape out from the reactor.

Such a neutron balance requirement can be expressed in mathematical form. To begin with, it is handy to consider the life cycle history of a single neutron. Essentially, the birth of a neutron usually begins as a result of a fission event. Then, the neutron will usually scatter for multiple times within the reactor region until it arrives at the point where its death or disappearance occurs. Also, some of the neutrons that were absorbed by fissionable nuclei will turn out inducing another fission reaction. This leads to the creation of a new generation of fission neutrons. Now, the quantity that describes the balance of neutrons within a nuclear reactor is known as the *effective multiplication factor*, which is defined as,

$$\begin{aligned} k_{\text{eff}} &\equiv \frac{\text{Number of neutrons born in a reactor}}{\text{Number of neutrons loss in a reactor}} \\ &\equiv \frac{\text{Number of neutrons in generation } j}{\text{Number of neutrons in previous generation } j-1} \end{aligned} \tag{3.52}$$

Note that if $k_{\text{eff}} = 1$, the number of neutrons produced in a reactor is equal to the number of neutron loss in a reactor. Consequently, the number of neutrons in any two consecutive fission generations will be the same. Plus, the chain reaction will become time-independent. For the case when $k_{\text{eff}} < 1$, there will be more neutron loss in the reactor since the denominator of Eq. (3.52) is larger than its numerator. Accordingly, the fission chain is expected to die off and the reactor is said to be in the state of subcritical. Finally, if $k_{\text{eff}} > 1$, the chain reaction is expected to grow exponentially, since more fission neutrons are produced during each successive fission generation. Such a condition is coined as supercritical.

In conclusion, the effective multiplication factor plays an extremely important role in assessing the stability of a nuclear reactor. The calculation of k_{eff} characterizing a specific reactor configuration and composition is often the main interest of most nuclear engineers. From this point, readers will find the rest of this book will be focusing on the development of various methods and procedures for performing this calculation.

3.6 Reactor *k*-Eigenvalue Equation

With the available knowledge of neutron transport, it is now suitable to determine the composition and size of a particular reactor so that the reactor is critical, i.e. $k_{\text{eff}} = 1$. Such a determination is crucial in the reactor core designing and management process since nuclear engineers will always desire to have a self-sustaining (critical) and a constant power-producing nuclear reactor. Inevitably, the existence of the time derivative in the multigroup equation given by Eq. (3.43) indicates that the number of neutrons in nuclear reactors is not always balanced over time. In reality, the state of a nuclear reactor always fluctuates over time. Therefore, assessing the state of neutron balance in a particular nuclear reactor using the time-dependent transport equation is not practical. Plus, such a practice can cause difficulties during the process of designing a self-sustaining nuclear reactor.

Practically, determining the criticality of a nuclear reactor with a specific geometry and composition requires the time-averaging of the neutron transport equation. Here, *time-averaging* simply means that the average reactor behaviour over time is observed instead. Note also that a time-averaged transport equation is often called the static eigenvalue equation. One of the approaches to create a static eigenvalue equation is to force the time derivative of the transport equation (continuous form or multigroup form) to zero. Next, a scaling factor, $1/k$, on the fission term of the transport equation is introduced. To comprehend the reason behind the introduction of the scaling factor $1/k$, it is useful to first consider the multigroup diffusion equation given by Eq. (3.43). However, the time derivative of the multigroup diffusion equation is set to zero so that the equation is time-averaged and static. For this reason,

$$\begin{gathered}\frac{1}{v_g}\underbrace{\frac{\partial \phi_g(\mathbf{r},t)}{\partial t}}_{=0} + \nabla \cdot D_g(\mathbf{r})\,\nabla\phi_g(\mathbf{r}) + \Sigma_{\mathrm{t}}^{g}(\mathbf{r})\,\phi_g(\mathbf{r}) \\ = \sum_{g'=1}^{G} \Sigma_{\mathrm{s}}^{g'\to g}(\mathbf{r})\,\phi_{g'}(\mathbf{r}) + \underbrace{\chi_g \sum_{g'=1}^{G} \nu_{g'}\Sigma_{\mathrm{f}}^{g'}(\mathbf{r})\phi_{g'}(\mathbf{r})}_{\text{Fission Term}} \\ g = 1,2,3,\dots,G\end{gathered} \tag{3.53}$$

Recall that the right-hand side of Eq. (3.53) indicates the gain of neutrons within the reactor and the left-hand side of Eq. (3.53) indicates the loss of neutrons from the reactor. Setting $\partial\phi_g/\partial t = 0$ forces the net gain of neutrons to be equal with the net loss of neutrons from the reactor, and such condition only possible when the reactor is critical. As a result, Eq. (3.53) has no general solution, unless the exact combination of the core composition, geometry and group constants such that the reactor is critical is just happened to hit. The only way to alleviate this issue is to introduce the $1/k$ factor to the fission term of Eq. (3.53). Henceforth, this will mathematically scale the fission term so that the equation satisfies for any reactor condition, even if the reactor core is not critical, i.e. subcritical or supercritical. Thus,

$$\begin{gathered}\nabla \cdot D_g(\mathbf{r})\,\nabla\phi_g(\mathbf{r}) + \Sigma_\mathrm{t}^g(\mathbf{r})\,\phi_g(\mathbf{r}) \\ = \sum_{g'=1}^{G} \Sigma_\mathrm{s}^{g'\to g}(\mathbf{r})\,\phi_{g'}(\mathbf{r}) + \frac{1}{k}\chi_g \sum_{g'=1}^{G} \nu_{g'}\Sigma_\mathrm{f}^{g'}(\mathbf{r})\phi_{g'}(\mathbf{r}) \\ g = 1,2,3,\dots,G\end{gathered} \tag{3.54}$$

For simplicity, an operator $\widehat{\mathrm{T}}$ is introduced such that,

$$\begin{aligned}\widehat{\mathrm{T}}_g\phi_g &\equiv \left(\nabla \cdot D_g\,\nabla + \Sigma_\mathrm{t}^g\right)\phi_g \\ &= \nabla \cdot D_g\,\nabla\phi_g + \Sigma_\mathrm{t}^g\,\phi_g\end{aligned} \tag{3.55}$$

Finally, Eq. (3.54) can be written into its equivalent matrix form:

$$\begin{gathered}\mathrm{diag}\left(\widehat{\mathrm{T}}_1,\widehat{\mathrm{T}}_2,\dots,\widehat{\mathrm{T}}_G\right)\begin{pmatrix}\phi_1\\ \phi_2\\ \vdots\\ \phi_g\\ \vdots\\ \phi_G\end{pmatrix} \\ = \begin{pmatrix}\Sigma_\mathrm{s}^{1\to1} & \dots & \Sigma_\mathrm{s}^{G\to1}\\ \Sigma_\mathrm{s}^{1\to2} & \cdots & \Sigma_\mathrm{s}^{G\to2}\\ \vdots & \ddots & \vdots\\ \Sigma_\mathrm{s}^{1\to G} & \cdots & \Sigma_\mathrm{s}^{G\to G}\end{pmatrix}\begin{pmatrix}\phi_1\\ \phi_2\\ \vdots\\ \phi_g\\ \vdots\\ \phi_G\end{pmatrix} + \frac{1}{k}\begin{pmatrix}\chi_1\nu_1\Sigma_\mathrm{f}^1 & \dots & \chi_1\nu_G\Sigma_\mathrm{f}^G\\ \chi_2\nu_1\Sigma_\mathrm{f}^1 & \cdots & \chi_2\nu_G\Sigma_\mathrm{f}^G\\ \vdots & \ddots & \vdots\\ \chi_G\nu_1\Sigma_\mathrm{f}^1 & \cdots & \chi_G\nu_G\Sigma_\mathrm{f}^G\end{pmatrix}\begin{pmatrix}\phi_1\\ \phi_2\\ \vdots\\ \phi_g\\ \vdots\\ \phi_G\end{pmatrix}\end{gathered} \tag{3.56}$$

It is also convenient to simplify the form of Eq. (3.56) by introducing several matrix operators, **T**, **S** and **F**, such that

$$\mathbf{T}\boldsymbol{\Phi} = \mathbf{S}\boldsymbol{\Phi} + \frac{1}{k}\mathbf{F}\boldsymbol{\Phi} \tag{3.57}$$

where,

$$\mathbf{T} \equiv \mathrm{diag}\left(\widehat{\mathrm{T}}_1,\widehat{\mathrm{T}}_2,\dots,\widehat{\mathrm{T}}_G\right) \tag{3.58}$$

$$\mathbf{S} \equiv \begin{pmatrix}\Sigma_\mathrm{s}^{1\to1} & \dots & \Sigma_\mathrm{s}^{G\to1}\\ \Sigma_\mathrm{s}^{1\to2} & \cdots & \Sigma_\mathrm{s}^{G\to2}\\ \vdots & \ddots & \vdots\\ \Sigma_\mathrm{s}^{1\to G} & \cdots & \Sigma_\mathrm{s}^{G\to G}\end{pmatrix} \tag{3.59}$$

and

$$\mathbf{F} \equiv \begin{pmatrix} \chi_1 \nu_1 \Sigma_{\mathrm{f}}^1 & \cdots & \chi_1 \nu_G \Sigma_{\mathrm{f}}^G \\ \chi_2 \nu_1 \Sigma_{\mathrm{f}}^1 & \cdots & \chi_2 \nu_G \Sigma_{\mathrm{f}}^G \\ \vdots & \ddots & \vdots \\ \chi_G \nu_1 \Sigma_{\mathrm{f}}^1 & \cdots & \chi_G \nu_G \Sigma_{\mathrm{f}}^G \end{pmatrix} \tag{3.60}$$

Also, the group flux vector, Φ, is given by,

$$\Phi \equiv (\phi_1 \quad \phi_2 \quad \cdots \quad \phi_G)^{\mathrm{T}} \tag{3.61}$$

Next, it is suitable to re-arrange Eq. (3.57) into,

$$(\mathbf{T} - \mathbf{S})^{-1}\mathbf{F}\Phi = k\Phi \tag{3.62}$$

If the above matrix equation is carefully analysed, the operator $(\mathbf{T} - \mathbf{S})^{-1}\mathbf{F}$ is a square matrix, which has the dimension of $G \times G$. So, Eq. (3.62) is actually an eigen-equation with Φ, a column vector identified as the eigenvector and the scalar value k as the eigenvalue. Recall that the whole point of neutron transport theory is to solve the neutron flux, which in this case the eigenvector Φ. Plus, the existence of the eigenvalue, k, in Eq. (3.62) allow nuclear engineers to use it as a tool for searching the right combination of reactor composition and geometry such that the reactor is critical.

At this level, the readers might question on how to solve the eigen-equation in Eq. (3.62) so that Φ and k are determined. One must also note that the reactor composition and geometry will change the characteristics of the square matrix operator $(\mathbf{T} - \mathbf{S})^{-1}\mathbf{F}$. Thus, solving the eigen-equation yields a specific value of k and a specific group flux vector, Φ. Essentially, the solution of the eigenvalue problem given by Eq. (3.62) can be accomplished by using the standard common technique in numerical analysis known as the *power iteration method*.

Recall that in a nuclear reactor, neutrons serve as the fission chain reaction carrier. Additionally, a neutron from the previous generation may induce another fission reaction, creating neutrons of the next fission generation. To begin with, it is customary to introduce an integer denoting the current fission generation number, j. Firstly, notice that if the group flux vector of the neutron population from the previous fission generation, $\Phi^{(j-1)}$ is known, then it is possible to determine the group flux vector of the current fission generation, $\Phi^{(j)}$ using Eq. (3.62), where,

$$\Phi^{(j)} = \frac{1}{k^{(j-1)}}(\mathbf{T} - \mathbf{S})^{-1}\mathbf{F}\Phi^{(j-1)} \tag{3.63}$$

Unfortunately, the true value of k is unknown and the true group flux vector of the previous generation is not known. Consequently, the issue is resolved by estimating the initial guess of the group flux vector, $\Phi^{(0)}$ and the initial guess of the effective multiplication factor, $k^{(0)}$. Thus, $\Phi^{(0)}$ and $k^{(0)}$ are used to obtain $\Phi^{(1)}$ using Eq. (3.63) whereas $k^{(1)}$ is estimated using an equation that will be derived later. This successive generation iteration continues until a sufficiently large number of iterations j so that the $\Phi^{(j)}$ and $k^{(j)}$ converge to their corresponding true value. The convergence of $\Phi^{(j)}$ and $k^{(j)}$ can be proven mathematically. Throughout the iteration, the value of $k^{(j)}$ and $\Phi^{(j)}$ are self-adjusted until the combination of these quantities satisfies the eigen-equation in Eq. (3.63). Eventually, this successive generation iteration will guarantee to converge regardless of the value of $\Phi^{(0)}$ and $k^{(0)}$ prescribed during the starting point of the iteration. For a large number of iterations,

$$\Phi^{(j)} \cong \frac{1}{k^{(j)}}(\mathbf{T}-\mathbf{S})^{-1}\mathbf{F}\Phi^{(j)} \tag{3.64}$$

Thus, if Eq. (3.64) is integrated over the entire space, it is reasonable to define the current estimate of the effective multiplication factor, $k^{(j)}$ as,

$$k^{(j)} \cong \frac{\int (\mathbf{T}-\mathbf{S})^{-1}\mathbf{F}\Phi^{(j)} d^3r}{\int \Phi^{(j)}\, d^3r} \tag{3.65}$$

Finally, by using the above formula it is now possible to compute a new guess of the effective multiplication factor, $k^{(j)}$. In summary, the iterative algorithm of finding the solution of the eigenvalue problem deterministically is given in Fig. 3.5.

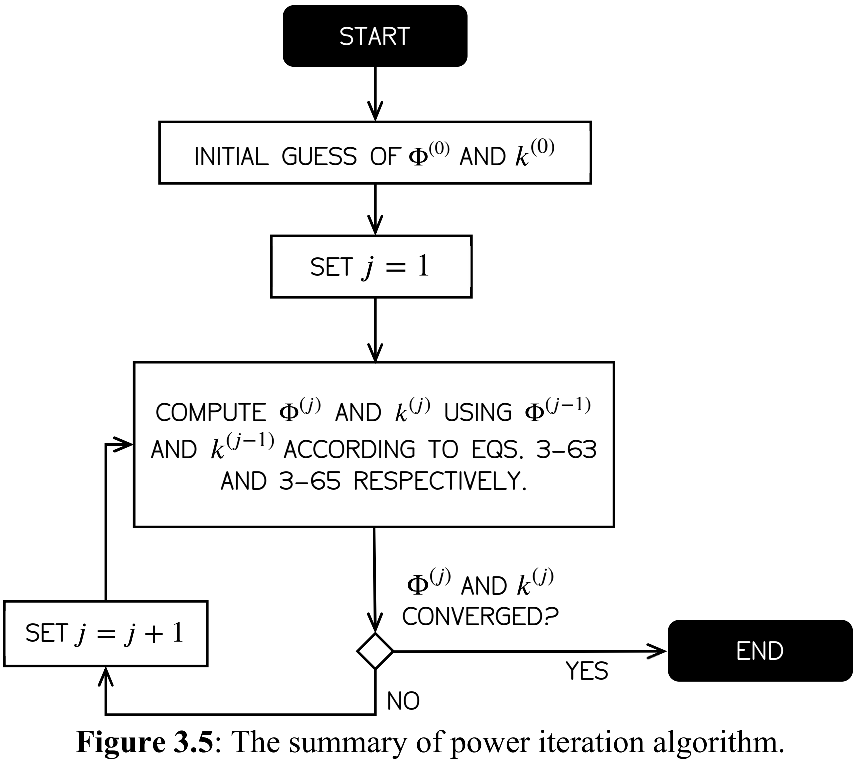

START
INITIAL GUESS OF $\Phi^{(0)}$ AND $k^{(0)}$
SET $j = 1$
COMPUTE $\Phi^{(j)}$ AND $k^{(j)}$ USING $\Phi^{(j-1)}$ AND $k^{(j-1)}$ ACCORDING TO EQS. 3-63 AND 3-65 RESPECTIVELY.
$\Phi^{(j)}$ AND $k^{(j)}$ CONVERGED?
YES
END
NO
SET $j = j + 1$

Figure 3.5: The summary of power iteration algorithm.

References

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

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

CHAPTER 4
THEORIES AND LAWS IMPLEMENTED IN TRIMON[1]

In the previous chapters, most of the neutron transport theorems have been presented in detail, and these theorems will become the foundation of the development of a multigroup Monte Carlo code for TRIGA reactors. This chapter outlines the details of the development of TRIMON (**TRI**GA **Mon**te Carlo Code), a next-generation reactor code that integrates diffusion-theory-type group cross sections into the Monte Carlo method for TRIGA reactors. Also, this hybrid combination speeds up stochastic simulations via homogenization of complex local core regions. TRIGA reactors are currently installed in 24 different countries, therefore, a robust core management code must be written for their safety analysis. Also, TRIMON is written in Fortran90 programming language. In fact, Fortran90 is chosen due to its simple and elegant modular programming paradigm that enables the developers and users to emphasize on the theoretical implementation of the code without the need of advanced programming knowledge. TRIMON incorporates critical features that improve certain functionalities that are less optimal in most state-of-the-art Monte Carlo codes such as direct integration of local fuel burnup in core calculation, sophisticated reactor core design considerations and simulation time improvements in complex core configurations. TRIMON also excludes the intricate jargon related to the core geometry and tally specifications required by most general multi-purpose Monte Carlo codes during the process of translating TRIGA core problems.

[1] This chapter is adapted from the following published research article: Omar, M. R., Karim, J. A., & Yoon, T. L. (2019). *The development of a multigroup Monte Carlo code for TRIGA reactors.* Nuclear Engineering and Design, 342.

4.1 Introduction to TRIGA Reactors

TRIGA is a commercial research reactor built by General Atomics, USA. Now, the reactor has been installed in 24 different countries. The reactor has been used for many diverse applications such as radioisotopes production, non-destructive testing, research on the properties of matter and for education and training. The reactor is a pool-typed water reactor and the reactor core is loaded with hydride fuel-moderator element, specifically U-ZrH. Most neutron moderations take place in the fuel element itself and the neutron moderation is mainly due to H in H-Zr (Henry, Tiselj, & Snoj, 2017). The reactor utilises demineralized water as coolant and moderator, where the loaded fuels are cooled by the flow of the coolant through the reactor core through natural convection or by forced cooling which depends on the reactor design.

TRIGA reactors are well recognised for its built-in safety characteristic due to a physical property of U-ZrH fuel. Here, TRIGA reactors have large prompt negative temperature coefficient. This implies that it is adequate to control an unexpected large insertion of positive reactivity to the reactor core. The fuel meat is a solid, homogeneous alloy of U-ZrH with the uranium enriched to 20% U-235. Also, the fuel meat is clad by a 0.051cm thick aluminium or stainless steel (SUS304) can.

Reaktor TRIGA Puspati (RTP) is a 1 MWth research reactor that has been installed in 1982 at Malaysian Nuclear Agency, Bangi, Malaysia. RTP core is a cylindrical-shaped core holding 127 designated core locations to accommodate fuel elements and other non-fuel elements such as control rods and irradiation facilities. The reactor core and the reflector assembly are mounted at the bottom of an aluminium tank situated inside the concrete shielding. The reactor core and experimental facilities are enclosed by a high-density concrete shielding. The reflector is made up of graphite and the reactor assembly is equipped with four boron carbide control rods. To provide vertical shielding, water is filled about 5m above the reactor core. Each element is arranged in seven concentric rings designated as Ring-A, Ring-B, …, Ring-G with 1, 6, 12, 18, 24, 30 and 36 core locations respectively.

4.2 TRIGA Core Unit Cells and Core Meshing

Theoretically, a reactor core can be built using its primary lattice structures which are identified as unit cells. The geometry of the unit cells is specified such that the complete reactor core can be formed using copies of these unit cells. For example, the unit cells forming a TRIGA core are illustrated in Fig. 4.1. A heterogeneous unit cell is composed of some separated regions of materials. In contrast, a homogenized unit cell is assigned such that the separated regions are “blended” while maintaining the integral neutron behaviour. In this context, neutron behaviour is a series of events where numerous neutron-nucleus interactions take place.

Essentially, a homogenized macroscopic neutron cross section, $\overline{\Sigma}$, is defined such that when they are used in the calculation of a homogenized unit cell, the net neutron leakage, the total neutron absorption and the total reaction rate remain the same as obtained in the calculation of the heterogeneous unit cell (International Atomic Energy Agency (IAEA), 1980). In TRIMON, each unit cell represents a homogenized reactor sub-region with constant neutron cross section. Customarily, when tracing the random walks of a neutron history, the current unit cell which holds the neutron is mapped so that the value of $\overline{\Sigma}$ for the cell can be retrieved and used for simulating the succeeding nuclear events.

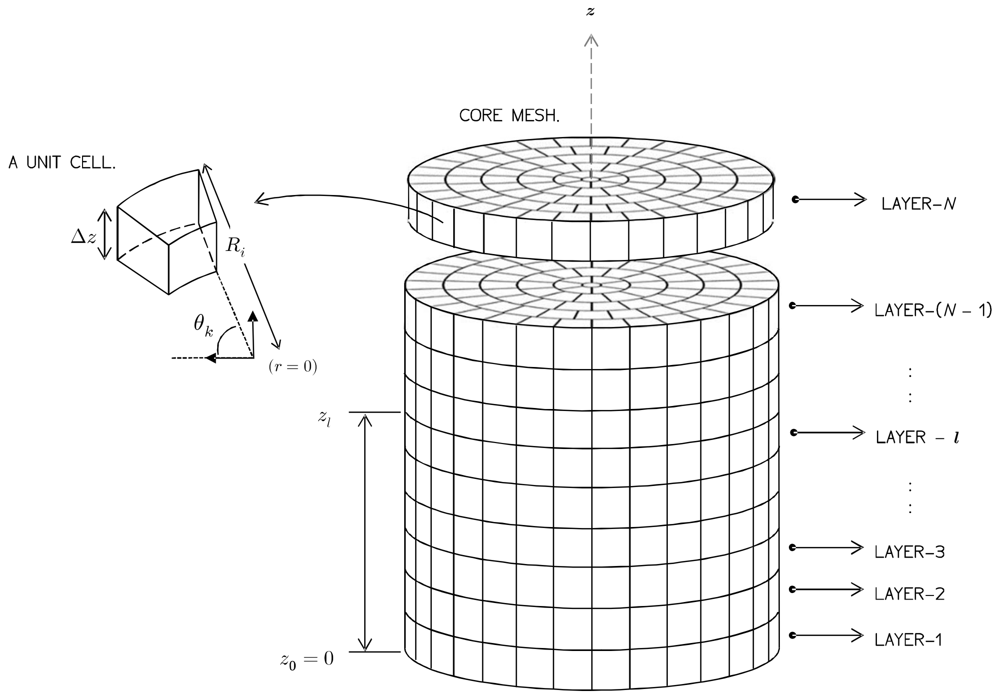


Figure 4.1: An assembly of unit cells forming a TRIGA reactor core.

To apply a similar concept to a TRIGA core, a unit cell is identified using cell indices $\langle i, k, l\rangle$. The radial position of the unit cell, R_i is determined using the radial index, i. The angular bearing of the unit cell, $\theta_{i,k}$ can be determined using combinations of the radial index and the angular index, $\langle i, k\rangle$. The cylindrical reactor core is sliced into several layers of equal thickness, Δz. Dividing the core into several layers will identify the z-axis position of the unit cell, z_l. A core layer can be determined using the core layer index, l. The position, $\mathbf{P}$, of a unit cell in the cylindrical coordinate system can be mapped using the set of cell indices $\langle i, k, l\rangle$. The relationship is defined as,

$$\mathbf{P} = R_i\, \mathbf{e}_r + \frac{\pi}{6}\frac{2k-1}{i-1}\mathbf{e}_\theta + l\Delta z\, \mathbf{e}_z, \tag{4.1}$$

where $\mathbf{e}_r$, $\mathbf{e}_\theta$ and $\mathbf{e}_z$ are the basis vectors that define the cylindrical coordinates.

4.3 Cell Homogenization

In common, the homogenization method starts with the transport calculation at the heterogeneous unit cell level. Subsequently, a heterogeneous unit cell is represented using its equivalent homogeneous unit cell of equal volume. Normally, a unit cell that includes fuel materials is assigned as a fuel cell. In a typical TRIGA core, a single fuel cell comprises a set of fuel regions plus the surrounding coolant water gap. A Zr rod is positioned at the centre of the fuel element (see Fig. 4.2). In TRIMON, only the active part of the fuel element is considered since it is the central part that drives the reactor core power generation.

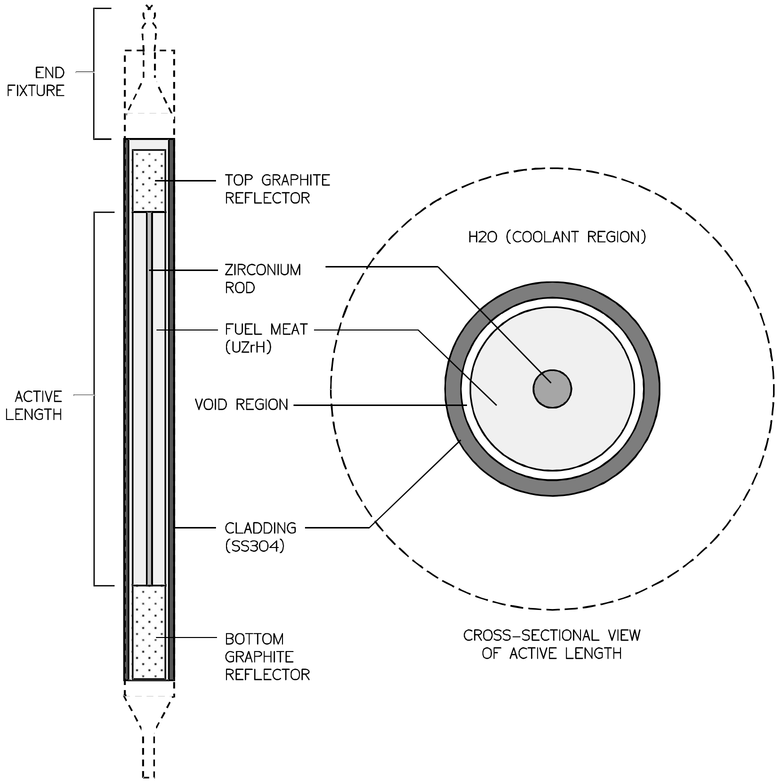


Figure 4.2: A schematic diagram of a TRIGA standard fuel element.

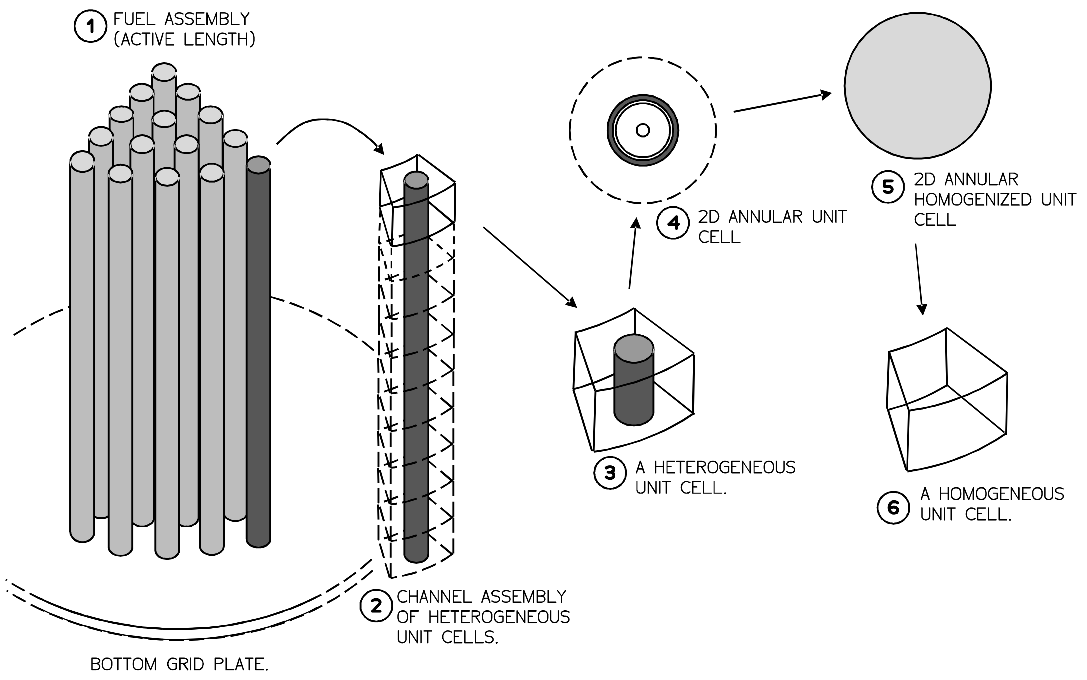

1 FUEL ASSEMBLY (ACTIVE LENGTH)
2 CHANNEL ASSEMBLY OF HETEROGENEOUS UNIT CELLS.
3 A HETEROGENEOUS UNIT CELL.
4 2D ANNULAR UNIT CELL
5 2D ANNULAR HOMOGENIZED UNIT CELL
6 A HOMOGENEOUS UNIT CELL.
BOTTOM GRID PLATE.

Figure 4.3: An overview of the steps involved in the procedure of homogenizing a unit cell.

Examine the illustration of a TRIGA core shown in Fig. 4.3. Suppose that a heterogeneous unit cell is taken from the core. For an annular TRIGA reactor core, one can effectively substitute a heterogeneous unit cell with a similar heterogeneous annular cell of equal volume. Following, the annular unit cell is accurately modelled in two-dimensional geometry to approximate the heterogeneous unit cell. At this level, the transport calculation of the two-dimensional model is performed to get the reference flux spectrum for the next homogenization step.

Neutrons will experience isotropic scattering as they arrive the cell boundary of the two-dimensional model (Trkov & Ravnik, 1994). Therefore, the transport problem at the unit cell level is indistinguishable to one involving a huge reactor core which is formed by many identical unit cells. Besides, the heterogeneous modelling method intends to achieve the best estimate of the original reactor core.

Next, the homogenization method continues with the use of the Effective Diffusion Homogenization (EDH) method (Trkov & Ravnik, 1994) for obtaining the homogenized neutron cross sections. Essentially, EDH method gives the approximate treatment of the radial leakage of the unit cell. Aside from maintaining the total reaction rates, the radial leakage in both homogeneous and heterogeneous unit cells is also conserved. So, this helps to decouple transport calculations of the unit cell from its surroundings. In essence, TRIMON employs the homogenization steps outlined by (Trkov & Ravnik, 1994).

Finally, a set of homogenized group neutron cross sections is obtained from the unit cell. The unit cell homogenization method is repeated for all unit cells within the reactor core. The calculated homogenized group neutron cross sections are grouped and compiled into a TRIGA cross section file (*.txs* file) and serve as a lookup table of the homogenized cross sections for the next full core Monte Carlo simulation.

4.4 Fuel Burnup Effect

In TRIMON, fuel burnup accumulation due to previous core operations is not ignored during cross section data pre-processing. Fuel depletion correction of neutron cross sections implemented in TRIMON does not need any additional fuel depletion codes as existing fuel depletion empirical relations are used. Fuel depletion calculation in TRIMON is based on the pre-defined correlations of the remaining mass fraction of U-235 in per cent, *b*, and its equivalent fuel burnup in MWd. Also, each predefined correlation is defined for a specific TRIGA fuel type, i.e. standard U-ZrH fuel types such as FE08 for 8.5%wt U, FE12 for 12%wt U and FE20 for 20%wt U.

In TRIMON, a fuel element is equally sliced axially into several fuel unit cells, where each unit cell holds a specific burnup level, *b*. The burnup, *b*, for each fuel unit cell in per cent U-235 is given by the general power series correlation,

$$b = \sum_{j=1}^{N} \beta_j p_{\text{cell}}^{j} (\Delta t)^j \tag{4.2}$$

where N is the order of the series, β_j are the coefficients of the series, p_{cell} is the fission power deposited in the cell, and Δt is the burnup increment in days. In TRIMON, the pre-set values of β_j were determined based on the WIMS calculations done by (A. Persic, Slavic, Ravnik, & Zagar, 1998) and the results of the calculation for each different standard U-ZrH fuel types are shown in Fig. 4.4. Plus, each type of fuel is assigned to a specific set of β_j. It is important to note that the values of β_j can be prescribed by the user through the main code input file. During each cell calculation, the fuel density and enrichment are corrected according to the prescribed value of b so that the number density of the fuel region within the cell can re-calculated.

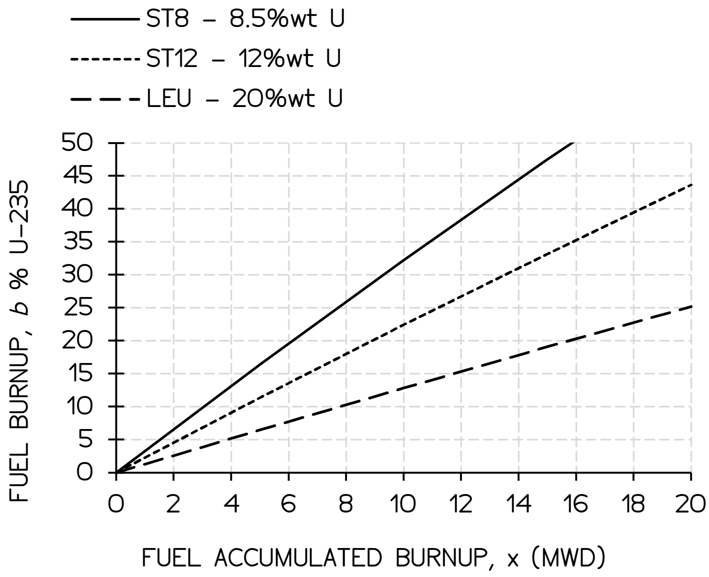


Figure 4.4: Fuel burnup in % U-235 correlations for different standard U-ZrH fuel types, 8.5%wt (ST8), 12%wt (ST12) and 20%wt (LEU). (A. Persic et al., 1998)

4.5 Fuel Temperature Effect

There is a clear correlation between the average fuel burnup and fuel temperature because the total fission energy liberated by the fuel material is largely exchanged into thermal energy. Inescapably, fuel temperature instigates the Doppler broadening of the neutron cross section's resonance region (Carter & Cashwell, 1975), therefore it greatly affects the heterogeneous cell's flux spectrum for the purpose of homogenization calculation. Principally, the temperature of a fuel cell is dependent on its position inside the core. For a TRIGA reactor, the temperature of a fuel cell is stemmed based on the empirical formula recommended by (Peršič et al., 2017),

$$T_{\text{cell}} = T_{\text{cool}} + \sum_{n=1}^{N} a_n \alpha_r \alpha_z \left\{ \frac{m}{M} \left(1 - \frac{b}{100}\right) P \right\}^n \tag{4.3}$$

where:

T_{cool} is the temperature of the coolant,
N is the power series order,
a_n is the coefficient of the power series,
α_r is the radial power form factor in the fuel cell location,
α_z is the axial power form factor at the fuel cell location,
m is the mass of Uranium in the fuel rod,
M is the total mass of Uranium within the reactor core,
b is the fuel rod burnup in per cent, and,
P is the nominal core power.

The fuel cladding temperature is provided by the average of T_{cool} and T_{cell}. The values of constants, a_n, are verified experimentally and needs to be indicated by the user in the code input. In TRIMON, Eq. (4.3) can be expressed as a piecewise function, such that a different set of a_n is designated to a different cell power interval.

4.6 Power Form Factors

The power form factor, α, of a reactor core portrays the profile of the power distribution inside a reactor core. In fact, it regulates the fuel temperature distribution relative to the hottest spot within a reactor core. Bear in mind that fuel temperature Doppler broadens a neutron cross section's resonance region, thus, α is a notable tool for scrutinizing the accuracy of the core simulation results. In the extent of this work, there are two types of power form factor to be pondered – the radial power form factor, α_r and the axial power form factor, α_z.

Meanwhile, TRIMON provides three-dimensional power distribution, the radial and axial power form factors can also be determined using TRIMON by the use of the algorithm summarized in Fig. 4.5. Notice also that power peaking factors can be calculated according to the straightforward method outlined by (Snoj & Ravnik, 2008).

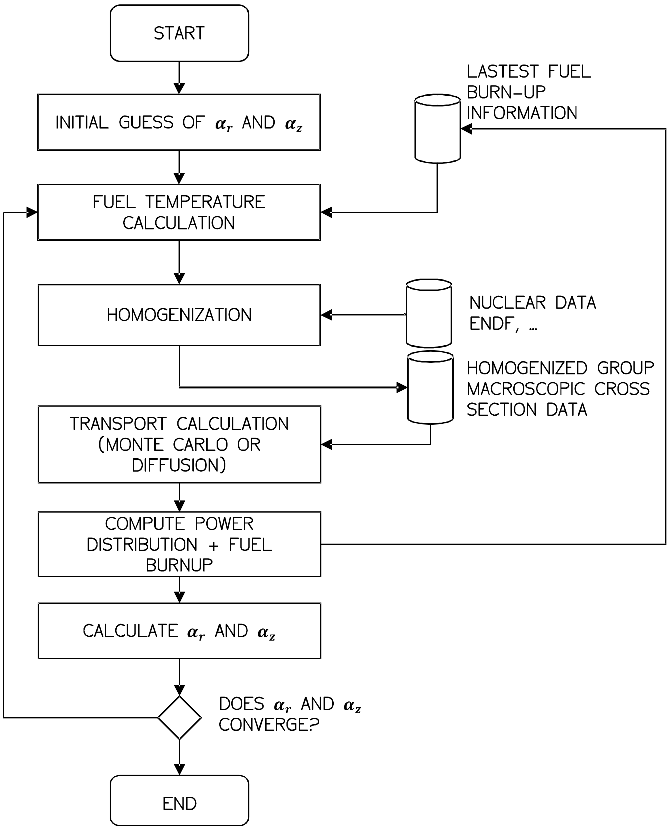

START
INITIAL GUESS OF α_r AND α_z
LASTEST FUEL BURN-UP INFORMATION
FUEL TEMPERATURE CALCULATION
HOMOGENIZATION
NUCLEAR DATA ENDF, ...
HOMOGENIZED GROUP MACROSCOPIC CROSS SECTION DATA
TRANSPORT CALCULATION (MONTE CARLO OR DIFFUSION)
COMPUTE POWER DISTRIBUTION + FUEL BURNUP
CALCULATE α_r AND α_z
DOES α_r AND α_z CONVERGE?
END

Figure 4.5: Simplified steps involved in the calculation of power peaking factors in TRIMON.

4.7 Homogenized Multigroup Monte Carlo Method

The Monte Carlo method is a technique of unravelling a deterministic problem by a stochastic approach by the use of neutron random walks. A number M of independent observations (e.g. neutron histories) are accumulated and the result is obtained from the averaged observation. The Monte Carlo method is frequently applied to deterministic problems that are hard to solve by deterministic methods. The main benefits of the Monte Carlo method include its straightforwardness where the transport equation does not have to be formulated to obtain the neutron flux in the reactor. And also, it applies to complicated problems without simplifications, for instance, it can model an exact complicated nuclear reactor core geometry.

In the Monte Carlo method, each neutron is simulated from the birth of a neutron until the death event of the neutron (i.e. due to a disappearance reaction or leakage). However, the Monte Carlo method presented in this work is distinctive from the usual method, where homogenized group neutron cross-section data is employed to enhance the simulation efficiency. In the conventional Monte Carlo method, non-homogenized neutron cross section data are used, which in fact increasing the complexity of the simulation. According to the homogenization theory discussed in Chapter 3, a non-homogenized or heterogeneous region may comprise of many different types of materials with different cross section values. Thus, in the conventional method, more stochastic calculations need to be accomplished to accommodate all of the different types of materials. However, a homogenized region comprised of a single pseudo-material with a single group neutron cross section will effectively reduce the number of stochastic calculations compared with the conventional Monte Carlo method.

4.7.1 Overview of Monte Carlo Method Implemented in TRIMON

In TRIMON, neutrons are anticipated to be travelling in a straight line in the direction $\widehat{\mathbf{\Omega}} = (u, v, w)$ from an initial position, $\mathbf{r}_0$. Also, it assumes that the neutron is only generated by a fission reaction. Other types of neutron production are disregarded, and their effects are insignificant. For occasion, nearly all $(\mathrm{n}, x\mathrm{n})$ reactions happen at high incident neutron energy, normally beyond 10MeV. Consequently, such a reaction is atypical since a huge fraction of neutrons in a thermal reactor holds an energy regime of less than 10MeV. Neutron production reaction caused by other sorts of particles is also abandoned since these particles are not tracked.

As a general rule, when a neutron journeys within a homogenized unit cell, there is a likelihood that it will strike a nucleus of the homogenized unit cell material. Throughout this collision, there are only three types of reactions counted – neutron absorption, neutron elastic scattering and fission reaction. It is sensible to bring to mind that the Monte Carlo method designated in this research utilizes homogenized neutron cross sections. Then, it is crucial to highlight that all neutron behaviours within the homogenized unit cell are virtual.

Nevertheless, when these virtual behaviours are integrated over the whole volume of the unit cell, they bear a resemblance to the overall behaviour of the neutrons when the heterogeneous neutron cross sections are used in the simulation. Astonishingly, a similar idea is spotted in the delta tracking method in Monte Carlo explained by (Leppänen, 2010). In delta tracking method, the concept of virtual collision is employed, and it effectively homogenized the material total cross section in a way that the total neutron path length within a whole geometry is conserved.

4.7.2 Distance to Next Collision and Random Walks

As a rule, a neutron is set off at a position $\mathbf{r}_0$ and stops at the point of the next collision event, $\mathbf{r}_\mathrm{c}$. Afterwards, the current unit cell, $\langle i, k, l\rangle$, is mapped where the position $\mathbf{r}_0$ is embedded inside $\langle i, k, l\rangle$. Consequently, the distance to the next collision, $\|\mathbf{r}_\mathrm{c} - \mathbf{r}_0\|$, can be calculated,

$$\|\mathbf{r}_c - \mathbf{r}_0\| = -\frac{\ln \zeta}{\overline{\Sigma}_{\mathrm{t},ikl}} \tag{4.4}$$

where $\overline{\Sigma}_{\mathrm{t},ikl}$ is the total homogenized neutron cross section of the unit cell $\langle i, k, l\rangle$ and ζ is a random number where $\zeta \in [0,1)$.

Trailing the transport process further, the distance to the nearest boundary of the current unit cell, $\|\mathbf{r}_b - \mathbf{r}_0\|$, along the neutron trajectory direction, $\widehat{\mathbf{\Omega}}$, is calculated. If the distance to the next collision event is larger than the distance to the nearest cell boundary, i.e., $\|\mathbf{r}_c - \mathbf{r}_0\| > \|\mathbf{r}_b - \mathbf{r}_0\|$, subsequently, the neutron is transported to the nearest boundary. Or else, the neutron is transported to the next collision location, $\mathbf{r}_c$. To sum up, the final location of the neutron is given by,

$$\mathbf{r} = \mathbf{r}_0 + \min(\|\mathbf{r}_c - \mathbf{r}_0\|, \|\mathbf{r}_b - \mathbf{r}_0\|)\,\widehat{\mathbf{\Omega}} \tag{4.5}$$

At the collision site, the neutron will progress for a neutron–nuclear interaction. Contrariwise, if the neutron is transported to the cell boundary, the distance to the next collision will be calculated using the total neutron cross section of the neighbouring unit cell. This process repeats until the neutron is absorbed by a nucleus or leak out of the reactor core. As soon as the neutron is absorbed, the simulation ends and the next neutron from the fission source bank is simulated.

4.7.3 Distance to the Nearest Cell Boundary

At first, a set of boundary surfaces, β, that characterize the boundary of a homogeneous unit cell is taken into consideration. Easily, the distance of a neutron to each boundary surface $b \in \beta$ along the neutron flight direction $\widehat{\mathbf{\Omega}}$ is given by $\|\mathbf{r}_b - \mathbf{r}_0\|$. Consider a neutron at $\mathbf{r}_0$ is moving in the direction $\widehat{\mathbf{\Omega}}$. To calculate $\|\mathbf{r}_b - \mathbf{r}_0\|$, one must solve the surface equation of the boundary of the unit cell,

$$f_s\left(\mathbf{r}_0 + \|\mathbf{r}_b - \mathbf{r}_0\|\widehat{\mathbf{\Omega}}\right) = 0 \tag{4.6}$$

where $f_s(\mathbf{r})$ is the surface equation of b.

It is essential to mention that the solution to Eq. (4.6) can be more than one and can be real or complex. Undoubtedly, if the solutions are all complex, therefore the direction of flight will never cross with the surface. Then again, a real solution which is less than zero implies that the surface is behind the neutron flight path. Preferably, a positive real solution is always chosen. To finish, the minimum of the positive solutions is selected to be the value of $\|\mathbf{r}_b - \mathbf{r}_0\|$.

4.7.4 Sampling a Reaction at the Collision Site

TRIMON distinguishes three major neutron-nucleus interactions – neutron capture, neutron elastic scattering and fission. In addition, neutron absorption includes entirely other types of reaction such as (n, p), (n, α), etc. Moreover, all of these reactions involve capturing a neutron to give off secondary particles except neutrons. Focusing on fission reaction, it is regarded as an absorption reaction because a nucleus absorbs a neutron before dividing into two daughter nuclei in the course of fission event.

In effect, the homogenized material is supposed to contain a single type of nucleus with a set of homogenized group cross sections that is constant throughout the unit cell. Consequently, one does not need to sample any nuclide of collision. First of all, the total macroscopic cross section of the unit cell is retrieved from the homogenized group neutron cross section data. In the second place, if a neutron in energy group g experiences collision with a nucleus within a unit cell, the first step is to decide whether the neutron is absorbed by the nucleus. To finish, this procedure is based on the algorithm outlined by (Romano & Forget, 2013),

1. A random number, ζ, is obtained from the random number generator.
2. The neutron is killed if the following condition is met,

$$\zeta < \frac{\overline{\Sigma}_{\mathrm{a}}^{g} - \overline{\Sigma}_{\mathrm{f}}^{g}}{\overline{\Sigma}_{\mathrm{t}}(g)} \tag{4.7}$$

where $\overline{\Sigma}_{\mathrm{a}}^{g}$ is the homogenized absorption cross section, $\overline{\Sigma}_{\mathrm{f}}^{g}$ is the homogenized fission cross section and $\overline{\Sigma}_{\mathrm{t}}^{g}$ is the homogenized total macroscopic cross section the unit cell. Finally, the neutron history tracking is terminated and the next neutron in the fission bank is simulated.

4.7.5 Multi-Group Scattering

Basically, the neutron energy and direction are transformed after a neutron scattering event. Therefore, the outgoing energy group g' and the outgoing direction $\widehat{\mathbf{\Omega}}'$ of the neutron following a scattering event is calculated. If a homogeneous group cross section data employs G neutron groups, thus the group scattering cross section from the incident energy group g to all existing outgoing energy groups are queried from the homogenized cross section data table.

At the initial stage of processing a group scattering event, a random number $\zeta \in [0,1)$ is selected from the random number generator. Then, the outgoing energy group of the neutron is chosen using the inverse sampling method, such that the value of ζ is compared to the cumulative distribution function, $F(g \to g')$, of the outgoing energy group, g', for the given incoming energy group, g. The cumulative distribution function can be stated as,

$$F_g(g \to g') = \sum_{1 \le \gamma \le g'} \frac{\overline{\Sigma}_\mathrm{s}^{g \to \gamma}}{\overline{\Sigma}_\mathrm{t}^{g} - \overline{\Sigma}_\mathrm{a}^{g}} \tag{4.8}$$

where $\overline{\Sigma}_\mathrm{s}$ is the homogenized scattering cross section, $\overline{\Sigma}_\mathrm{t}$ is the homogenized total neutron cross section and $\overline{\Sigma}_\mathrm{a}$ is the homogenized absorption cross section. Suitably, the term $\overline{\Sigma}_\mathrm{t} - \overline{\Sigma}_\mathrm{a}$ is the total homogenized scattering cross section, in any case of any outgoing energy group. A similar method can be done to pick out the outgoing angle of the neutron. Contrariwise, one can sample the value of μ by making use of the following relation if the scattering direction is expected to be isotropic,

$$\mu = 2\zeta - 1. \tag{4.9}$$

Automatically, the outgoing neutron direction can be calculated given that the outgoing energy and scattering cosine have been determined. The scattering cosine μ denotes the cosine of the angle between incident neutron direction, $\widehat{\mathbf{\Omega}}$, and the outgoing neutron direction $\widehat{\mathbf{\Omega}}'$. The scattering cosine is expressed as,

$$\mu = \widehat{\mathbf{\Omega}}' \cdot \widehat{\mathbf{\Omega}} \tag{4.10}$$

Accordingly, it is probable to determine $\widehat{\mathbf{\Omega}}'$ by sampling an azimuthal angle $\phi \in [0,2\pi)$ from the uniform distribution. If the components of $\widehat{\mathbf{\Omega}}'$ and $\widehat{\mathbf{\Omega}}$ are given as (u', v', w') and (u, v, w) respectively, the relationship between $\widehat{\mathbf{\Omega}}'$ and $\widehat{\mathbf{\Omega}}$ is given by (Romano & Forget, 2013),

$$u' = \mu u + \sqrt{\frac{1-\mu^2}{1-w^2}}(uw\cos\phi - v\sin\phi) \tag{4.11}$$

$$v' = \mu v + \sqrt{\frac{1-\mu^2}{1-w^2}}(vw\cos\phi - u\sin\phi) \tag{4.12}$$

$$w' = \mu w + \sqrt{1-\mu^2}\sqrt{1-w^2}\cos\phi \tag{4.13}$$

4.7.6 Fission Reaction

Despite the fact that fission is considered as an absorption reaction (A. Persic et al., 1998), TRIMON deals with fission similarly to inelastic scattering. This is because fission neutrons are set off as the outcome of the fission event. At the early step of a fission process, the average number of fission neutrons, ν_g, is estimated based on the incoming neutron energy group, g. In particular, the correlation relation between υ_g and g is given by (A. Persic et al., 1998)

$$\nu_g = 2.55 - 0.11\frac{g-1}{\max(1, G-1)} \tag{4.14}$$

where G is the total number of energy groups used in the simulation. After that, the total number of neutrons to be emanated after the fission event, ν, is predicted using the method summarized by (Romano & Forget, 2013). Then again, homogenized cross sections are used instead. The value of ν can be determined using,

$$\nu = \frac{W \nu_g \overline{\Sigma}_\mathrm{f}}{k_{n-1} \overline{\Sigma}_\mathrm{t}} \tag{4.15}$$

where W is the neutron weight and k_{n-1} is the value of k_eff from the previous fission cycle. Here, the value of ν is rounded to the nearest integer.

Later, the outgoing energy of these fission neutrons is sampled. It is appropriate to make known to the cumulative fission energy group spectrum, $F(g')$, which is defined as the cumulative probability of having energy group g' as the outgoing fission neutron energy group. In most cases, one can sample the outgoing energy of a fission neutron from Watt Distribution. In summary, $F(g')$ is expressed as,

$$F(g') = \sum_{\gamma \le g'} \int_\gamma \mathrm{d}E' c\, \mathrm{e}^{-aE'} \sinh \sqrt{bE'} \tag{4.16}$$

where $c = 0.453$, $a = 1.036$, $b = 2.29$ (Duderstadt & Hamilton, 1976) and E' is the outgoing energy variable. Notice that the integration limits are the energy boundaries that define the entire outgoing energy group γ. Consequently, a random number, ζ is obtained from the random number generator and the outgoing energy group is determined using the inverse sampling method (Ross & Ross, 2013) where the value of ζ is compared to $F(g')$ and the corresponding value of g' is determined.

The finishing step is to sample the outgoing direction, $\widehat{\mathbf{\Omega}}'$, of the fission neutron. In TRIMON, the outgoing angle of a fission neutron is isotropic at all instances. Certainly, the algorithm used to sample the outgoing direction of a neutron after isotropic scattering is used for predicting the outgoing direction of a fission neutron.

4.7.7 Cell Flux Tally Scoring and Calculation of Fuel Element Power

TRIMON make use of track length estimators when collecting flux tallies of an entire volume of a unit cell. Nevertheless, tallies collection is not done for a volume lesser than the volume of the unit cell due to unit cell homogenization. Actually, all collisions that occur inside a homogenized unit cell are virtual, thus, tallies accumulation for a volume smaller than the unit cell is improper.

According to (Romano & Forget, 2013), the derivation of track length flux estimator begins with the volume integrated flux for a specific neutron energy group,

$$V_{\text{cell}}\,\phi_g = \iiint \left[\int_{E_g}^{E_{g-1}} \psi(\mathbf{r}, \hat{\mathbf{\Omega}}, E, t)\, dE \right] d\mathbf{\Omega}\; dr\; dt \tag{4.17}$$

where ψ is the angular flux and V_{cell} is the volume of the unit cell of concern. Observe also that ψ can be rewritten in terms of the neutron density function, $N(\mathbf{r}, E, t)$, therefore,

$$V_{\text{cell}}\,\phi_g = \iint \left[\int_{E_g}^{E_{g-1}} vN(\mathbf{r}, E, t)\, dE \right] dr\; dt, \tag{4.18}$$

where v is the average neutron speed. Using the basic classical mechanics' formula, $dt = dl/v$, yields,

$$V_{\text{cell}}\,\phi_g = \iint \left[\int_{E_g}^{E_{g-1}} N(\mathbf{r}, E, t)\, dE \right] dr\; dl\;. \tag{4.19}$$

From Eq. (4.19), the $\int dl$ term indicates that the total distance travelled by any neutrons in the entire cell must be calculated. The remaining terms indicate that the number of neutrons in energy group g at any position within the unit cell needs to be accumulated. Consequently, the track length estimator of the relative flux for neutron energy group g is given by,

$$\phi_g(i, k, l) = \frac{1}{W} \sum_{k \in \lambda} w_k(g) l_k, \tag{4.20}$$

where λ is the set of all tracks within the unit cell $\langle i, k, l \rangle$.

In the cylindrical coordinate system, the power radiated by a fuel element, P_{el}, is given as (A. Persic et al., 1998),

$$P_{\text{el}} = p_0 \iiint_{V_{\text{el}}} \phi(r, \theta, z)\, \overline{\Sigma}_{\text{f}}(r, \theta, z)\; dr\; d\theta\; dz, \tag{4.21}$$

where p_0 is the normalization factor, ϕ is the neutron flux and $\overline{\Sigma}_{\text{f}}$ is the homogenized fission cross section. The normalization factor can be calculated using,

$$p_0 = P \left[\iiint_{V_{\text{core}}} \phi(r, \theta, z)\, \overline{\Sigma}_{\text{f}}(r, \theta, z)\; dr\; d\theta\; dz \right]^{-1}. \tag{4.22}$$

Here, P is the nominal power of the reactor core in kW. Recall that a unit cell can be identified using the cell indices $\langle i, k, l \rangle$. A fuel element occupies the total volume of unit cells which have the same i and k values. Thus, the volume integral in Eqs. (4.21) and (4.22) can be numerically evaluated using,

$$P_{\text{el}} = p(i, k) = p_0 \sum_{l=1}^{n_l} \sum_{\forall g} \phi_g(i, k, l)\, \overline{\Sigma}_{\text{f}}^{g}(i, k, l), \tag{4.23}$$

where n_l is the total number of core layers, and

$$p_0 = P \left[\sum_{\forall (i,k,l)} \sum_{\forall g} \phi_g(i, k, l)\, \overline{\Sigma}_{\text{f}}^{g}(i, k, l) \right]^{-1}. \tag{4.24}$$

Note also that in Eq. (4.23), the summations include all neutron energy groups for all unit cells within the fuel element channel. On the other hand, the summations in Eq. (4.24) include all neutron energy groups for all unit cells within the reactor core.

4.8 Monte Carlo Criticality Calculation in TRIMON

In general, criticality calculation, or sometimes known as eigenvalue calculation is a transport simulation of neutrons studying the ability of a system, i.e. a nuclear reactor core, to sustain a fission chain reaction. In essence, TRIMON employs the Monte Carlo power iteration method to perform the criticality calculation of a TRIGA reactor. Recall that the Monte Carlo power iteration method necessitates history tracking of a finite number of neutrons from one generation to another. Conveniently, these generations are recognized as MC cycles. If fission happens during history tracking, the location of the fission site, the outgoing energy group and direction of the fission neutron and the weight of the neutron are kept for the use in the next MC cycle.

Since most Monte Carlo programmers represent neutron transport quantities using stacks, queues and arrays, the mathematical analysis incorporated in this book is expressed in terms of finite discrete phase space using vectors and matrices. The neutron phase space is discretized into cells, which is termed as phase space cells. To this point, any neutron that exits the problem phase space is considered escaped. Plus, any functions of $\mathbf{r}$, $\widehat{\mathbf{\Omega}}$ and E become vectors and any operators become square matrices. Most importantly, the order of all matrices and vectors is equal to the number of phase-space cells.

To begin with, it is convenient to define $\Phi^{(j)}$ as the expected number of neutrons in the corresponding phase-space cells during MC cycle j. Next, the eigen-equation in Eq. (3.63) can be further simplified into,

$$\begin{aligned}\Phi^{(j)} &= \frac{1}{k^{(j-1)}}(\mathbf{T}-\mathbf{S})^{-1}\mathbf{F}\Phi^{(j-1)} \\ &= \frac{1}{k^{(j-1)}}\mathbf{R}\Phi^{(j-1)}\end{aligned} \tag{4.25}$$

with $\mathbf{T}$, $\mathbf{S}$, $\mathbf{F}$ and $\mathbf{R}$ are square matrix operators with $\mathbf{R} \equiv (\mathbf{T}-\mathbf{S})^{-1}\mathbf{F}$, $k^{(j-1)}$ is the current estimate of the effective multiplication factor, and j is the iteration cycle index such that $j \geq 1$.

In this sense, the neutron source distribution of the current MC cycle, $\Phi^{(j)}$, can be determined by evaluating the matrix operation of the right-hand side of Eq. (4.25). Here, $\Phi^{(j-1)}$ is the neutron source distribution obtained from the previous MC cycle, $j-1$. In the Monte Carlo method, the right-hand side of Eq. (4.25) can be 'solved' by tracking neutrons that are selected from $\Phi^{(j-1)}$, starting from their birth location until their death after disappearance reactions. In this context, neutron disappearance reactions include fission and neutron capture by a nucleus.

To begin with, suppose M neutrons are randomly selected from $\Phi^{(j-1)}$ and initiated. Also, M is known as the neutron batch size. Each neutron is tracked from its starting point until its death due to escape, Russian roulette and so on. In addition, the total number of neutrons initiated during each MC cycle is always normalized to M. When simulating each of these neutrons, the collision site is determined, and the neutron is transported to the collision site. During each collision, m duplicates of the collision site are stored in $\Phi^{(j)}$ as the probable fission site and m is given by (X-5 Monte Carlo Team, 2005),

$$m = \frac{w}{k^{(j-1)}} \frac{\nu\Sigma_\mathrm{f}}{\Sigma_\mathrm{t}} + \zeta \tag{4.26}$$

where w is the particle weight of the neutron and ζ is a random number in [0, 1). Equally important, the value of $k^{(j)}$ can be estimated using the collision estimator where the $k^{(j)}$ estimator is accumulated during each collision i (X-5 Monte Carlo Team, 2005),

$$k_\mathrm{col}^{(j)} = \frac{1}{M} \sum_i w_i \nu \frac{\Sigma_\mathrm{f}}{\Sigma_\mathrm{t}}. \tag{4.27}$$

There are also other types of $k^{(j)}$ estimators, e.g. the absorption estimator and track-length estimator. However, the discussion of these estimators is beyond the scope of this paper. Finally, $\Phi^{(j)}$ is assigned as the fission neutron source for the next MC cycle.

The power iteration method is always prescribed with the initial guess of the neutron source distribution, $\Phi^{(0)}$. Briefly, as the iteration progresses from one MC cycle to another, $\Phi^{(j)}$ will stochastically converge to an equilibrium state which is also known as the stationary state. When stationarity is implied, tallies such as neutron flux, neutron lifetime and the effective multiplication factor can be accumulated, and their averages can be calculated at the end of the iterations.

It is important to ensure that the value of k_eff and the fission site distribution, $\Phi^{(j)}$, converge before any tally accumulation is made. In fact, $\Phi^{(j)}$ converges slower than that of k_eff (Cho & Chang, 2009). For k_eff the convergence can be observed from the plot of k_eff versus the number of fission cycles. The procedure outlined by (Romano & Forget, 2013) is applied to check whether $\Phi^{(j)}$ has converged, where the fraction of fission source sites that are present in each unit cell is calculated,

$$S_i = \frac{n_i}{n} \tag{4.28}$$

where n_i is the number of fission source sites in the i^th unit cell and n is the total number of fission source sites within the reactor core. Next, the Shannon entropy of Φ is calculated,

$$H_\mathrm{src}\{\Phi\} = -\sum_{i=1}^{n} S_i \ \log_2 S_i \tag{4.29}$$

It is worthwhile to note that the convergence of H_src implies the convergence of Φ.

4.9 Code Design and Application

Principally, TRIMON is meant to simulate neutron transport problems in a TRIGA reactor core. Therefore, the material compositions of the reactor core are predefined in the code. And also, users do not need to identify the material compositions in the code input. The newly developed code recognizes core material compositions according to the core configuration, where each core channels are designated to various types of elements. A core channel may perhaps comprise of a fuel element or any various types of non-fuel element. The list of elements recognized by TRIMON is given in Table 4.1.

Principally, the TRIMON code comprises of two built-in modules. These modules are identified as WIMS Integrated TRIGA Cell Homogenization (WITCH) module and the Homogenized Group Monte Carlo (HGMC) module. WITCH module is intended to generate a homogenized neutron cross section lookup table (via a *.txs* file). Whereas the HGMC module is intended to perform the Monte Carlo simulation using the generated lookup table. It also produces the essential simulation outputs such as the power distribution, flux distribution and fuel burnups.

In relation with local fuel burnup consideration, the current accumulated burnup of fuel cells is acquired from the local burnup lookup table (the *zburn.out* file). Here, the lookup table is interpreted by WITCH module throughout the course of preparing the homogenized neutron cross section. Notice also that the local burnup table is updated after each burnup calculation.

Table 4.1: List of core channel elements.

Channel Element	**Description**
`FE08, FE12, FE20`	19.9% enriched $UZrH_{1.6}$ fuel element with 8.5%wt, 12%wt and 20%wt of uranium respectively; stainless steel (SS304) cladding. The fuel element is surrounded by coolant water.
`CHN1`	Irradiation Channel Type-I: An empty Al tube. CHN1 be used as an approximate model of a transient control rod.
`CHN2`	Irradiation Channel Type-II: Half void, half water in Al tube.
`CHN3`	Irradiation Channel Type-III: Full water in Al tube. Can be used to represent the central thimble of the reactor core.
`GRAP`	Graphite Element: Graphite in Al tube.
`COOL`	Coolant water (without Al tube)

The general workflow of TRIMON code is depicted in Fig. 4.6. TRIMON call for two input files to work. The case to be solved is given by the user through a formatted text file, *main.inp*, which contains various input cards. The second input file, *fuel_inventory.inp*, contains fuel elements information used in the calculation.

TRIMON code is set as a package which comprises of two different Windows™ program (*.exe*) files, where both programs execute WITCH and HGMC modules independently. These program files were created after the compilation of the source codes using the GNU Fortran compiler. The execution of these programs is not error-free since runtime errors may arise due to incorrect usage of input cards. Opportunely, the code is intended to alert users on particular error encounters through a comprehensible error message.

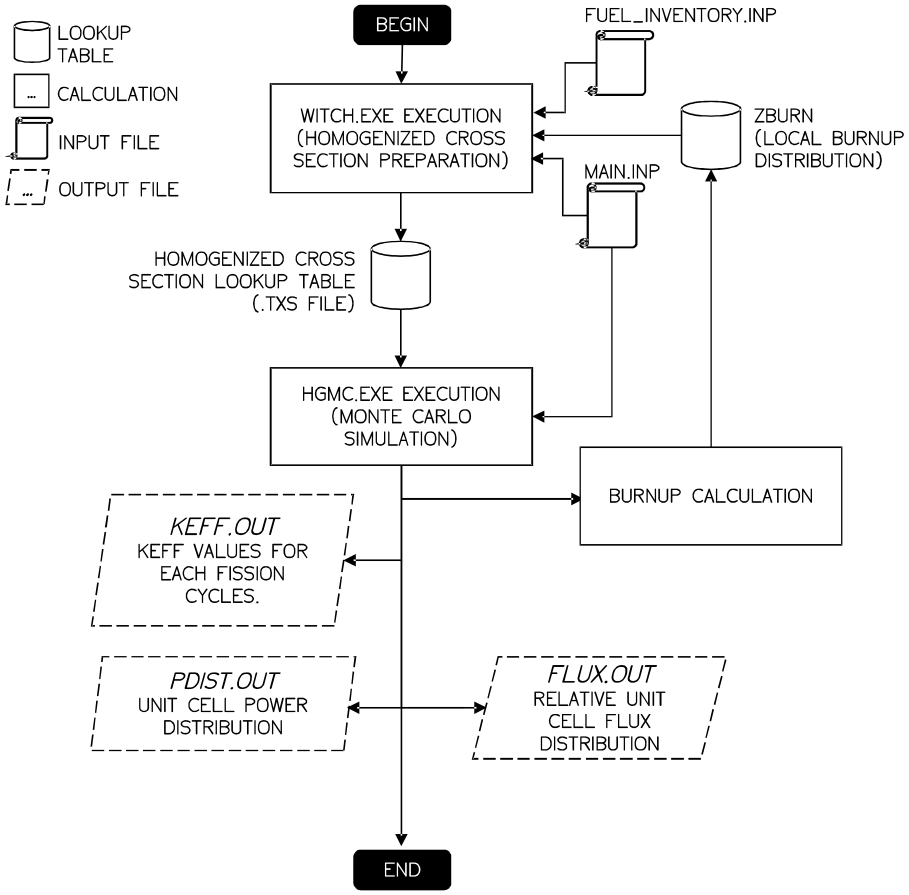

LOOKUP TABLE
CALCULATION
INPUT FILE
OUTPUT FILE
BEGIN
FUEL_INVENTORY.INP
WITCH.EXE EXECUTION (HOMOGENIZED CROSS SECTION PREPARATION)
ZBURN (LOCAL BURNUP DISTRIBUTION)
MAIN.INP
HOMOGENIZED CROSS SECTION LOOKUP TABLE (.TXS FILE)
HGMC.EXE EXECUTION (MONTE CARLO SIMULATION)
BURNUP CALCULATION
KEFF.OUT
KEFF VALUES FOR EACH FISSION CYCLES.
PDIST.OUT
UNIT CELL POWER DISTRIBUTION
FLUX.OUT
RELATIVE UNIT CELL FLUX DISTRIBUTION
END

Figure 4.6: TRIMON code workflow.

4.10 TRIMON Validation and Benchmark Analysis

Unavoidably, it is compulsory for a newly developed neutron transport code to get validated in order to ensure its consistency in producing decent simulation results. At this point, TRIMON validation and tests were done in two ways. This includes comparing the code numerical output results with the actual operational reactor core data recorded in the reactor logbook and also by comparing the code simulation output results with the simulation output results produced by any well-established reactor design code such as MCNP. It is important to stress that TRIMON implements a method which is yet established by any other Monte Carlists, thus, thorough simulation validation and tests must be done before releasing it to the users.

4.10.1 RTP Approach to Criticality Benchmark

In this core benchmark problem, the number of fuel elements inserted into Malaysian Puspati TRIGA Reactor (RTP) core was incremented and the corresponding value of k_{eff} was measured. The measurement was made for each subsequent number of fuel rods until the reactor core reaches criticality. The core configuration at the moment of criticality is designated as Core-0 configuration. Fig. 4.7 shows the comparisons of the results obtained from measurements, MCNP, TRIGLAV and TRIMON.

In the simulation, the problem conditions were set so that they reproduce the conditions of the reactor core during the actual measurements. The temperature of the coolant was set at 293K, the reactor core was xenon-free, and the power of the reactor is at negligible thermal power (<0.01kW). In this experiment, all of the loaded fuels are of 8.5%wt fresh fuels (zero burnup level).

A total of 30000 neutron histories were simulated per fission cycle. The criticality calculation was done for 200 cycles with 30 skipped cycles. The number of cycles was chosen to ensure that the relative errors of the averaged tallies obtained using MCNP and TRIMON falls below 0.1%. It is important to note that (Brown, 2011) recommends using more than 10000 neutron histories for all simulations to avoid significant bias in k_{eff} and any local tallies in MCNP.

The plot of k_{eff} versus the number of fuel elements exhibit significant agreement between the simulation results with the experimental results at high fuel element count. The plot also exhibits strong agreement between TRIMON, MCNP and TRIGLAV. TRIMON predicts that the criticality is achieved after 66 fuel elements are inserted in the reactor core where the critical core k_{eff} is 1.00095±0.00015. The validity of the simulation result is supported by the experimental plot where criticality was finally achieved after adding a similar number of fuel elements where the measured value of k_{eff} is 1.001.

The difference between the value of k_{eff} obtained from TRIMON and the actual experiment at the point of criticality is ~5 pcm. Since the systematic error of TRIMON is 0.00015 (~15 pcm), the predicted value of k_{eff} is well within the expected systematic error interval.

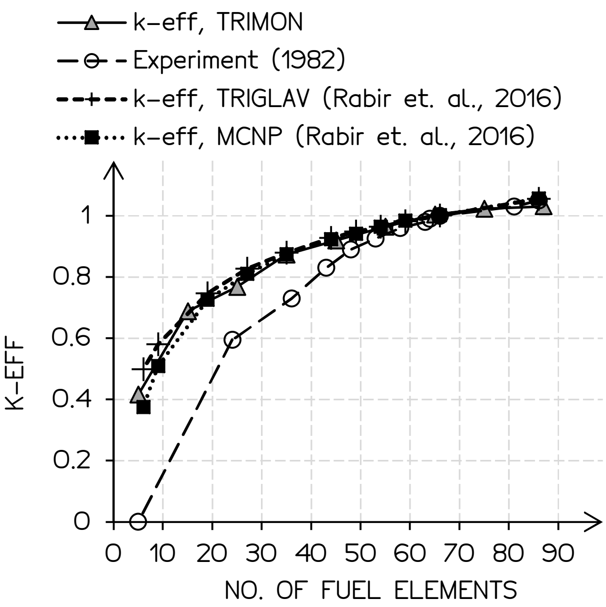


Figure 4.7: Approach to criticality curve.

Inevitably, there is a significant departure of the measured values of k_{eff} from simulated values of k_{eff} at lower fuel elements count. Specifically, this departure is caused by a smaller number of signals received by the detector as a result of its relatively large distance from the active core region. As the number of fuels increased, the active core region is becoming bigger and closer to the detector which is located outside the reactor. As a result, the measurements' uncertainty was significantly reduced.

4.10.2 TRIMON k_{eff} Comparison with Measured RTP Operational Core Data

In the operational core benchmark problem, the mixed core configurations of RTP which consists of 8.5%wt and 12%wt UZrH fuel elements were considered. To assess the reliability of TRIMON, the results of the criticality calculations the first six operational core configurations were compared with the measured operational data obtained from the operational logbook of RTP. In these criticality calculations, 30000 neutron histories were used for each fission cycle and 200 fission cycles with 30 skipped cycles were performed using TRIMON. For additional information, the reactor normally operates six (6) hours per day for four days a week. The details of the operation histories of RTP for the first six operational cores are given in Table 4.2. The summary of the results of the comparison is given in Table 4.3.

Table 4.2: Histories of the first six RTP operational cores.

Core	**Average Power (kW)**	**Total Burnup Given (MWd)**	**Accumulated Burnup (MWd)**	**Operating Hours (days)**
Core-1 (C1)	455.015	43.4	43.4	95.3
Core-2 (C2)	600.580	45.8	89.2	76.4
Core-3 (C3)	674.223	42.6	131.8	63.1
Core-4 (C4)	727.700	87.1	218.9	119.7
Core-5 (C5)	706.565	19.6	238.5	27.8
Core-6 (C6)	736.874	78.0	316.5	105.8

Table 4.3: Summary of criticality calculations using TRIMON for Core-1 to Core-6. Measured k_{eff} obtained from the RTP operational logbook is also given. Difference between calculated and measured, Δk_{eff}, is also given.

Core	k_{eff} **TRIMON**	k_{eff} **Measured**	Δk_{eff}
Core-1 (C1) BOC	1.05190 ± 0.00028	1.05312 ± 0.00038	0.00122 ± 0.00047
Core-1 (C1) EOC	1.03310 ± 0.00030	1.03095 ± 0.00037	0.00215 ± 0.00048
Core-2 (C2) BOC	1.05701 ± 0.00043	1.05377 ± 0.00039	0.00324 ± 0.00058
Core-2 (C2) EOC	1.03702 ± 0.00038	1.04004 ± 0.00038	0.00298 ± 0.00054
Core-3 (C3) BOC	1.05432 ± 0.00026	1.05323 ± 0.00039	0.00109 ± 0.00047
Core-3 (C3) EOC	1.04258 ± 0.00039	1.04193 ± 0.00038	0.00065 ± 0.00055
Core-4 (C4) BOC	1.06005 ± 0.00044	1.05657 ± 0.00039	0.00343 ± 0.00059
Core-4 (C4) EOC	1.03624 ± 0.00043	1.03883 ± 0.00038	0.00259 ± 0.00057
Core-5 (C5) BOC	1.05102 ± 0.00049	1.05183 ± 0.00039	0.00081 ± 0.00063
Core-5 (C5) EOC	1.04499 ± 0.00046	1.04882 ± 0.00039	0.00368 ± 0.00060
Core-6 (C6) BOC	1.04506 ± 0.00043	1.04867 ± 0.00038	0.00361 ± 0.00057
Core-6 (C6) EOC	1.03978 ± 0.00048	1.03936 ± 0.00038	0.00042 ± 0.00061

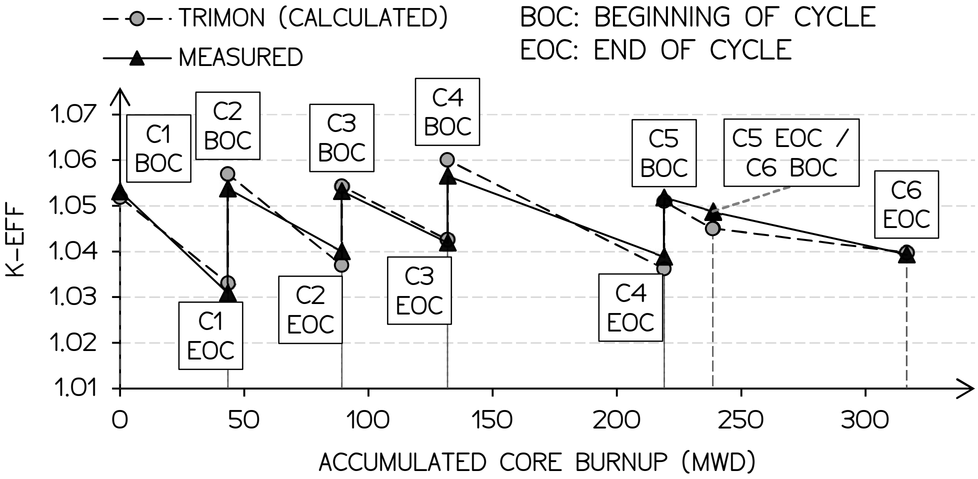


Figure 4.8: Core effective multiplication factors, k_{eff}, at zero core power versus accumulated core burnup in MWd from Core-1 to Core-6 (C1–C6).

The sawtooth-shaped plot depicted in Fig. 4.8 indicates a reduction of k_{eff} at the end of each operational core cycle (EOC) for both measured and calculated values. Then, the value of k_{eff} is restored every time the core is reconfigured at the beginning of cycle (BOC). During fuel reconfiguration, the existing high-burnup fuels may be replaced with new fresh fuels, or the same fuels loaded in the core may be reshuffled with the existing low-burnup fuels.

Each core configuration is assigned to a specific core identification, i.e. Core-*N* for the *N*th operational core. When a specific core configuration has reached its end-of-cycle and reconfigured, the new core configuration is assigned to a new identification, i.e. Core-(*N*+1). It is important to note that the four control rods are assumed to be fully withdrawn. Essentially, the fuel follower control rod model is approximated to a standard 8.5%wt fuel model and the transient rod model is approximated to an empty aluminium tube model.

Here, the simulations of the operational cores using MCNP is not present in this section since incorporating fuel burnup effect in MCNP calculation can be cumbersome and time-consuming. However, the MCNP result of the critical core (Core-0) and the first two operational cores (Core-1 and Core-2) will be presented in the next section. Some work done by (Chiesa et al., 2016; Rabir et al., 2017) proposed extra treatment to the MCNP simulation model in order to account the fuel burnup effect. On the other hand, TRIMON offers instant core burnup consideration without the need for extra work on the simulation model. Fig. 4.9 shows the local fuel burnup calculation results produced by TRIMON.

For all simulated core configurations, the measured multiplication factors were compared with the result obtained using TRIMON. Interestingly, the average percentage difference of k_{eff} is less than 1% $\Delta k/k$, thus, indicating that TRIMON is a reliable tool to be used for core management analysis of RTP. For all configurations, the temperature of the coolant was set at 298K and the reactor was xenon-free. All fuel elements are of 8.5%wt and 20%wt, 19.9% enriched standard UZrH fuel.

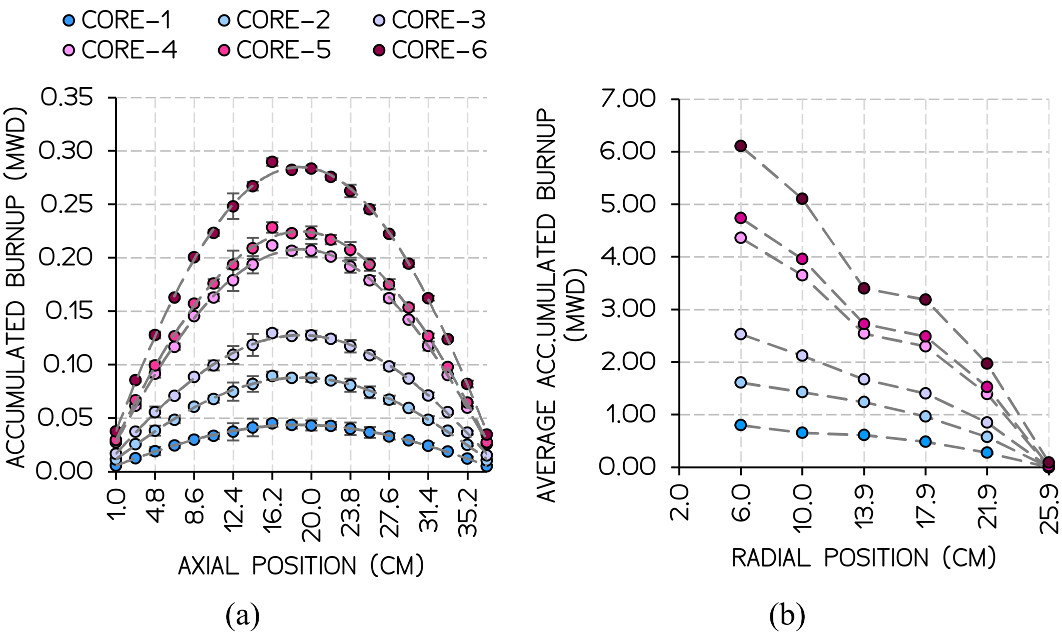


Figure 4.9: (a) Locally accumulated burnup in MWd along the axial length of a standard 8.5%wt fuel calculated using TRIMON. The fuel remains loaded in the reactor core throughout the six operational core configurations (C1–C6). (b) Average accumulated burnup at different core radial positions.

4.10.3 Comparison of TRIMON with MCNP Benchmark of RTP

In this benchmark problem, the criticality calculations of RTP critical core (Core-0), first operational core (Core-1) and burned core (Core-2) were performed using TRIMON and MCNP. Various tally results obtained using TRIMON and MCNP were compared for each of the three cores. In these criticality calculations, the tally results include the effective multiplication factor, fission reaction rate, total reaction rate, total flux and thermal flux. In addition, the performance of both TRIMON and MCNP in terms of simulation time and fission source convergence were also compared.

Essentially, these comparisons were made to assess the ability of TRIMON to reproduce the simulation results obtained using MCNP. Plus, this section reports the evaluation of the gain brought by TRIMON compared to the local information lost due to the use of homogenized neutron cross section data. The core location map of RTP core is shown in Fig. 4.10. The configuration of Core-0, Core-1 and Core-2 is depicted in Fig. 4.11 respectively.

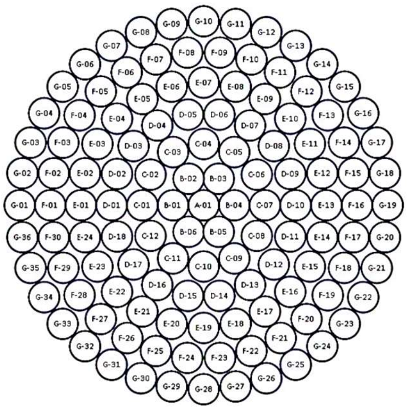
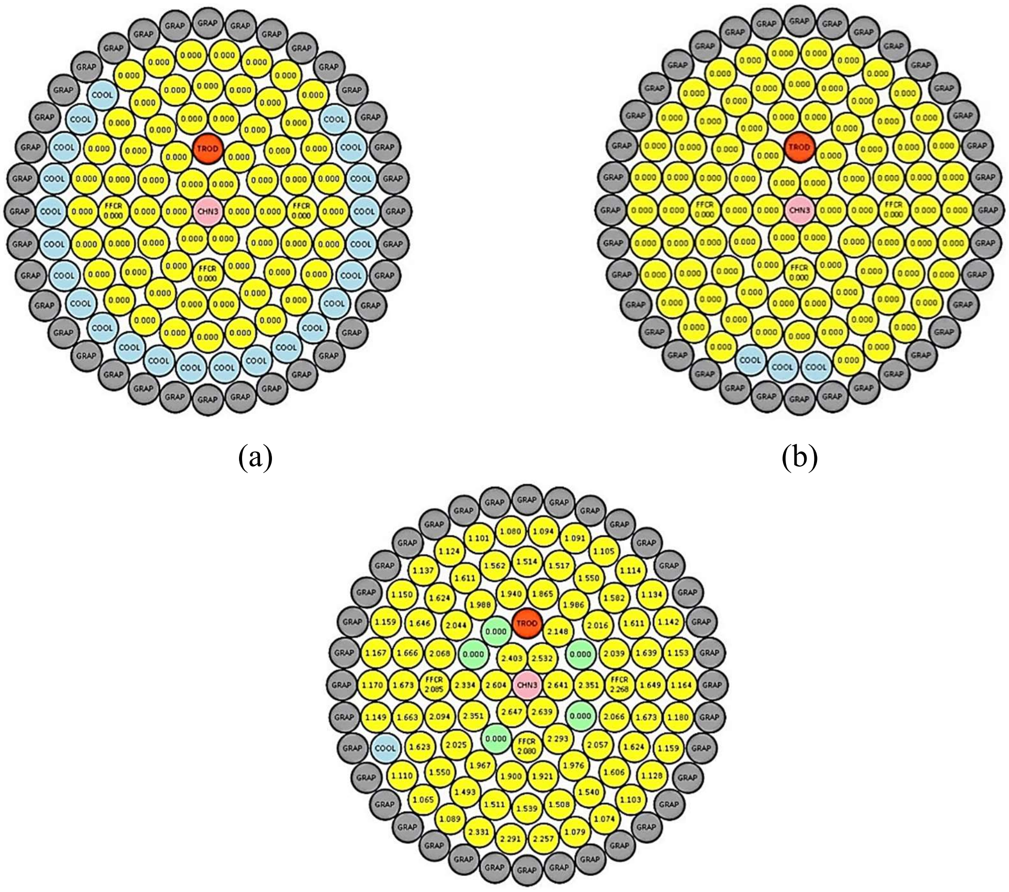


Figure 4.10: A 7-ring RTP core location identification map. A transient control rod (TROD) is located at C-04. Fuel follower control rods (FFCR) are located at D-01, D-10 and C-10.



(a) (b)

(c)

Figure 4.11: RTP operational core configurations (a) Critical core (Core-0) loaded with 66 fresh 8.5%wt UZrH fuels (yellow); (b) First operational core (Core-1) loaded with 86 fresh 8.5%wt UZrH fuels (yellow); (c) Second operational core (Core-2) loaded with 83 spent 8.5%wt UZrH fuels (yellow) and 5 fresh 12%wt UZrH fuels (green). Note: Numeric label indicates the fuel burnup in percent U-235.

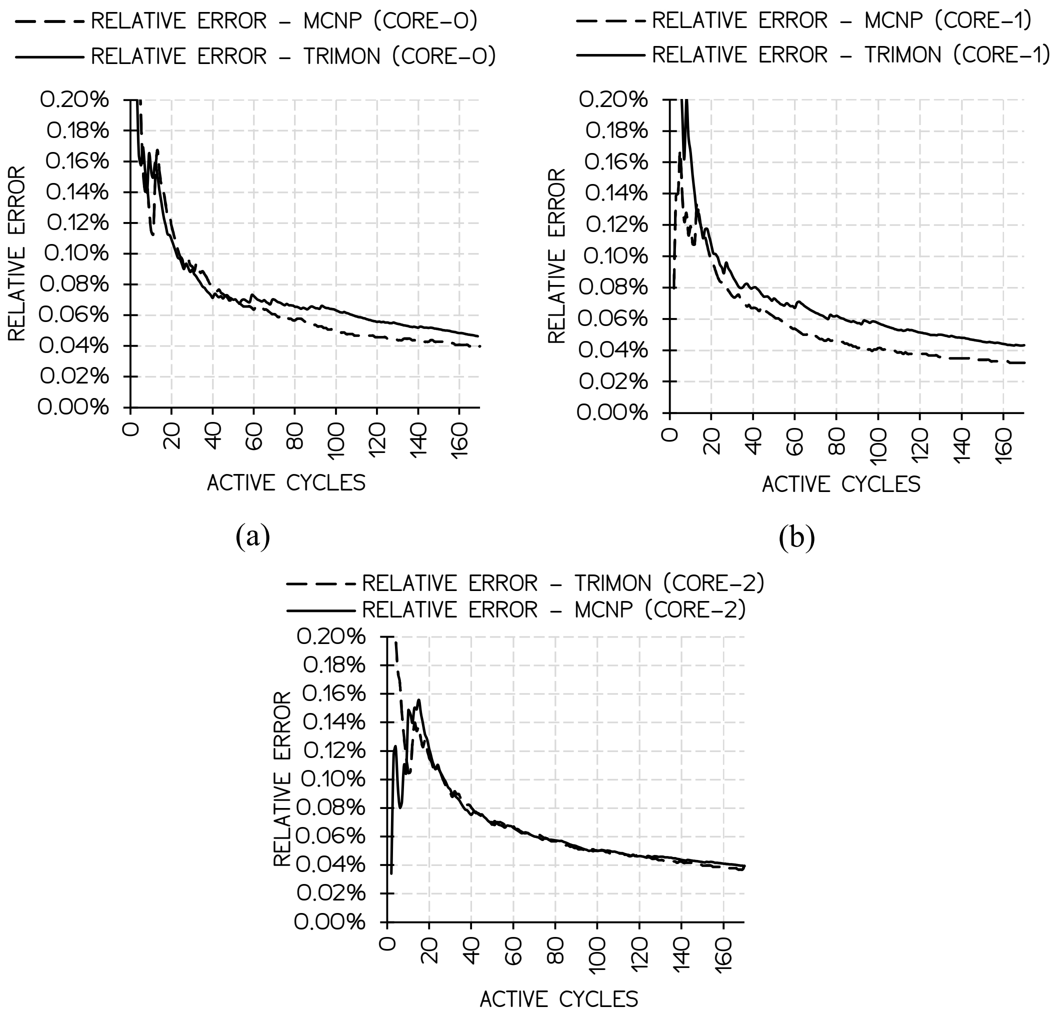
Again, the control rods are assumed to be fully withdrawn, and this applies to all TRIMON and MCNP simulations of Core-0, Core-1 and Core-2. Here, all fuel follower control rods (FFCR) were replaced with standard 8.5%wt UZrH fuels, the transient control rod (TROD) was approximated using a void aluminium tube model and the central thimble of the reactor was replaced with a full water-filled aluminium tube model (CHN3).

Table 4.4 summarizes the effective multiplication factors, k_{eff}, and the total simulation time of the criticality calculations performed using TRIMON and MCNP for the three core configurations. For each criticality calculation, the number of neutron histories per cycle was increased to 50000 compared to the calculations done in the previous sections. This was done to further reduce tally bias in TRIMON and MCNP especially in mixed fuels and burned cores so that comparisons can be made at better accuracy. These calculations were performed using 200 fission cycles. The evolution of the relative error of k_{eff} obtained using TRIMON and MCNP over 170 active cycles for the three cores is shown in Fig. 4.12. It can be observed that TRIMON impose a slightly extra uncertainty due to the use of homogenized cross section data. However, these extra uncertainties are much smaller compared to the extra simulation time consumed by MCNP. Fig. 4.13 shows the plot of CPU wall clock in minutes versus fission cycle. Plots in Fig. 4.13 can be interpreted as the total time required to accomplish a certain number of fission cycles.

The number of skip cycles was set to 30 for both TRIMON and MCNP to ensure that the fission source distribution entropy completely converges with minimum standard deviation. Hence, active fission cycles begin at cycle 31 where tallies accumulation and k_{eff} averaging was started.

Table 4.4: Effective multiplication factors and total CPU time for completing 200 fission cycles, 50000 neutrons/cycle.

Core	k_{eff} (TRIMON)	k_{eff} (MCNP)	Δk_{eff}	CPU Time (MCNP)	CPU Time (TRIMON)
Core-0	1.00388 ± 0.00046	1.00411 ± 0.00040	0.00023 ± 0.00061	45.6 mins	13.7 mins
Core-1	1.05154 ± 0.00046	1.05971 ± 0.00034	0.00817 ± 0.00057	40.4 mins	11.9 mins
Core-2	1.05752 ± 0.00042	1.05653 ± 0.00039	0.00099 ± 0.00057	273.6 mins	12.54 mins



(a)

(b)

(c)

Figure 4.12: Relative error evolution for (a) Core-0, (b) Core-1 and (c) Core-2.

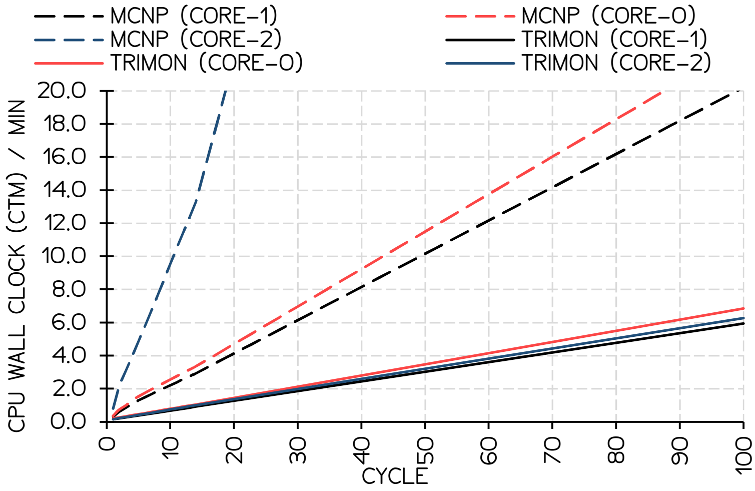


Figure 4.13: CPU wall clock of the first 100 cycles using TRIMON and MCNP.

In this benchmark problem, both TRIMON and MCNP codes were executed using the same machine. Both codes were also executed once at a time using Intel Core-i7 CPU @2.8GHz speed. It can be deduced that TRIMON has a better computational speed. Table 4.5 shows the figure of merit (FOM) of all reactor core calculations. Larger FOM is preferred because it means that less computational time is required to achieve a certain level of error, or in other words, the computation is more efficient.

Table 4.5: Figure of merit (FOM) of TRIMON and MCNP.

Core Configuration	**FOM = 1 / (CPU Time × Relative Error^2)**	
	TRIMON	**MCNP**
Core-0	411293	98075
Core-1	517744	154784
Core-2	600876	39878

Table 4.6: Convergence results for Core-0, Core-1 and Core-2 using TRIMON and MCNP.

	Core-0	**Core-1**	**Core-2**
Converged Cycle (TRIMON)	3	3	4
Converged Cycle (MCNP)	9	8	8
Converged Time (TRIMON)	20.0s	19.0s	25.1s
Converged Time (MCNP)	142.2s	112.2s	516.6s
Converged $\boldsymbol{H}_{\mathbf{src}}$ (TRIMON)	10.120 ± 0.001	10.480 ± 0.001	10.460 ± 0.001
Converged $\boldsymbol{H}_{\mathbf{src}}$ (MCNP)	8.640 ± 0.002	8.697 ± 0.001	9.350 ± 0.001

In the MCNP model of Core-0, Core-1 and Core-2, the reaction rates, total flux and thermal flux tallies were scored over the entire fuel rod cells plus the surrounding coolant water cell. Whereas in TRIMON, the tally volume of these quantities is integrated over all homogenized unit cells (as shown in Fig. 4.1) that build the entire fuel rod plus the surrounding coolant water. In TRIMON, scoring tallies over a region smaller than a unit cell is prohibited since TRIMON utilized spatially homogenized neutron cross section data. Thus, the same tally criteria are imposed to non-fuel cells such as coolant cell, graphite element cell etc.

The accumulated tallies obtained using TRIMON and MCNP were then normalized to a total of one reaction for reaction rates, a total of one neutron for the total flux and a total of one thermal neutron for the thermal flux. The normalization of these quantities was done to make the comparison simple and perceivable. The plot of the total flux, thermal flux, total reaction rate, and fission rate for Core-0, Core-1 and Core-2 are shown in Figs. 14.14-14.16 respectively. The relative differences in per cent of these quantities are also shown in these figures.

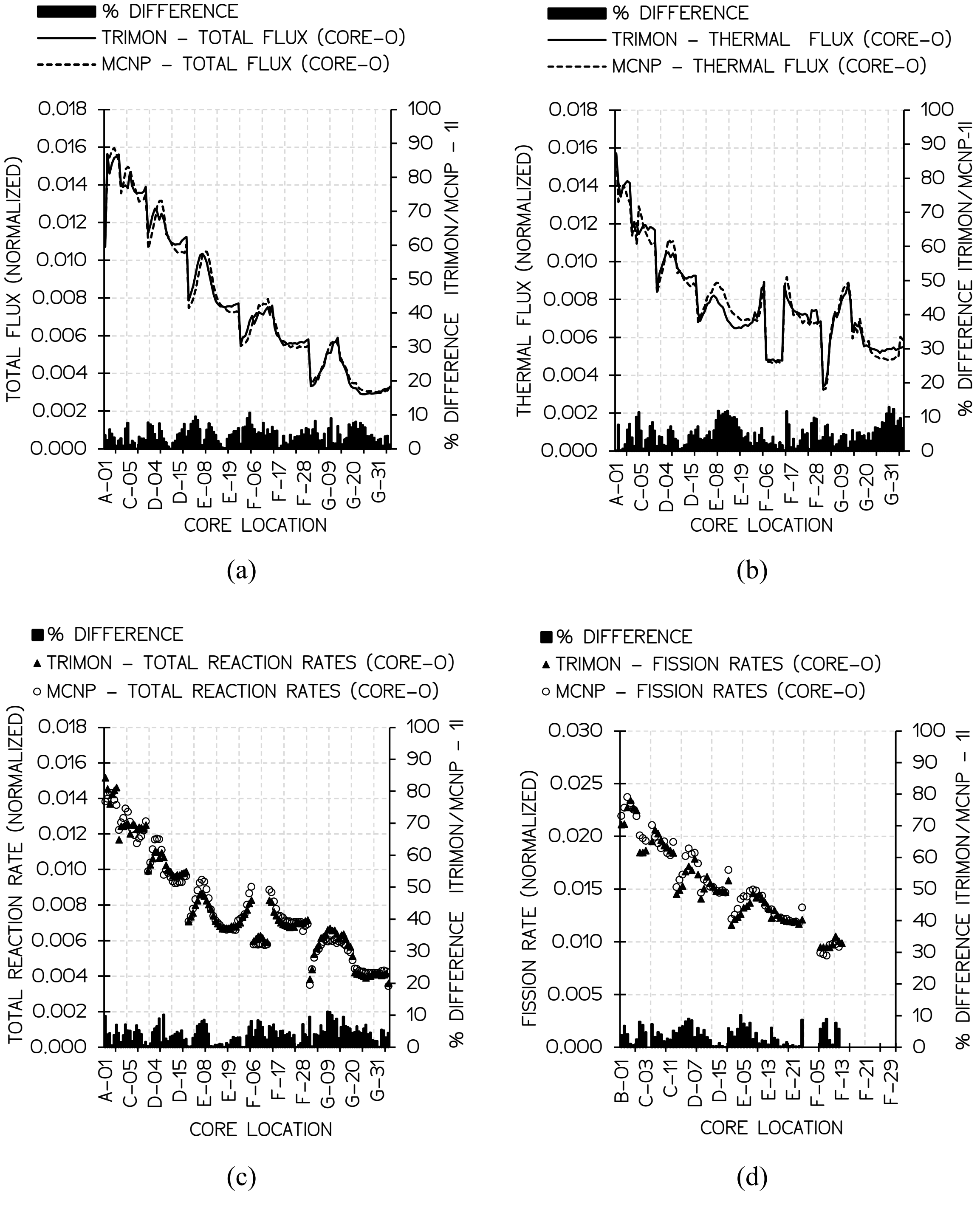


Figure 4.14: The plots of (a) total flux, (b) thermal flux, (c) total reaction rate and (d) fission rates across the entire core locations of Core-0. Note: The fluxes and the reaction rates are normalised to one neutron and one reaction respectively.

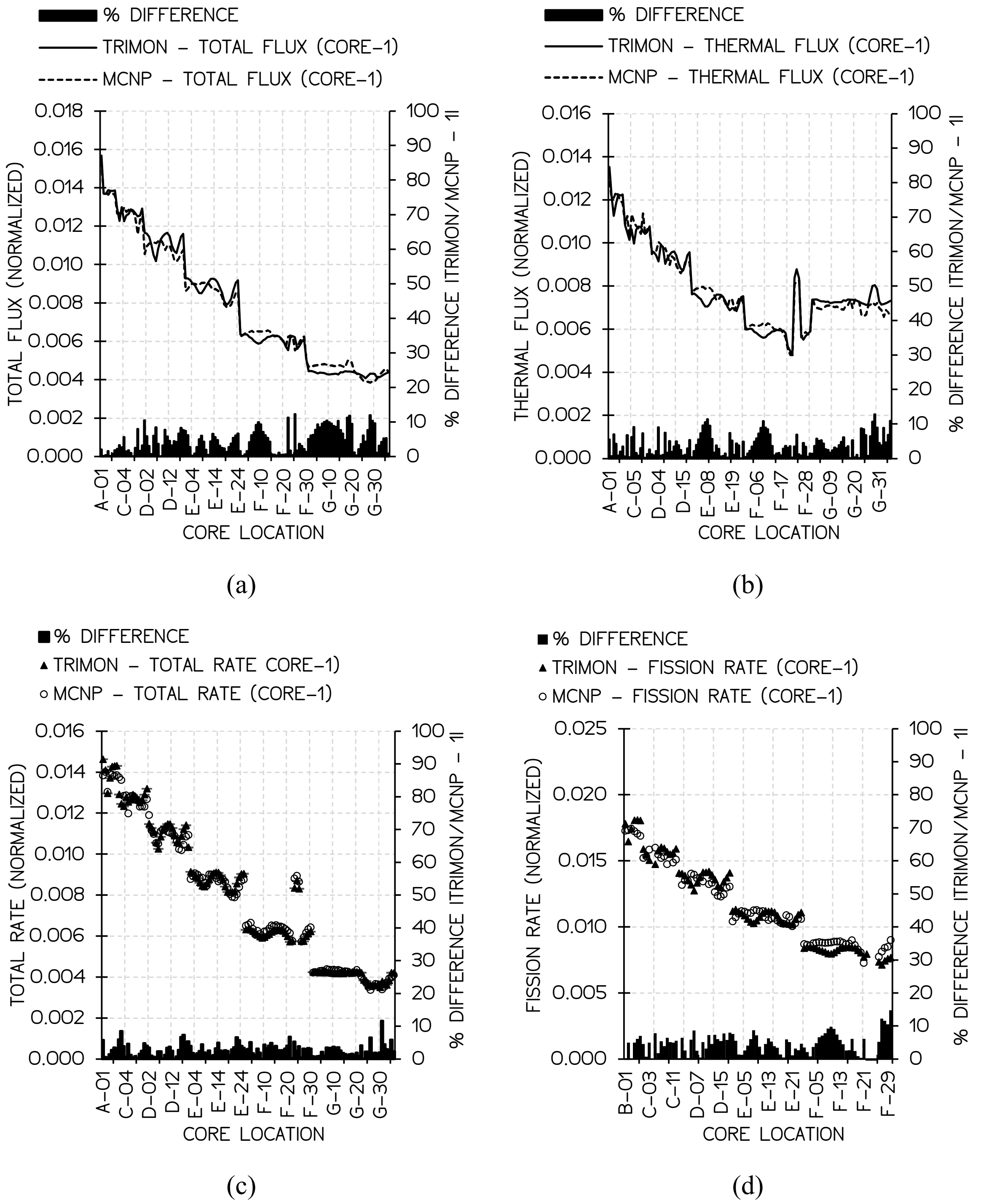
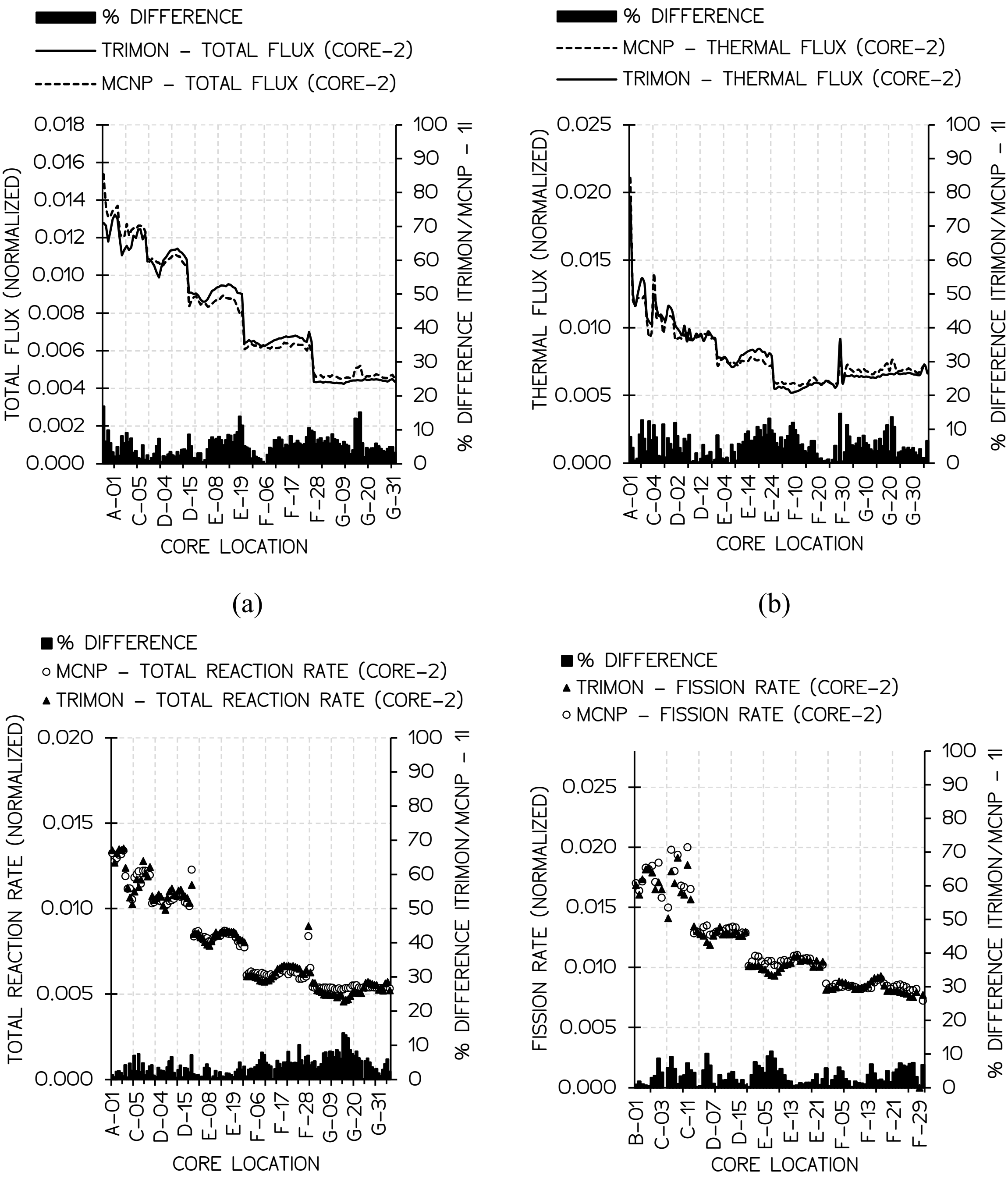


Figure 4.15: The plots of (a) total flux, (b) thermal flux, (c) total reaction rate and (d) fission rates across the entire core locations of Core-1. Note: The fluxes and the reaction rates are normalised to one neutron and one reaction respectively.



(a)

(b)

(c)

(d)

Figure 4.16: The plots of (a) total flux, (b) thermal flux, (c) total reaction rate and (d) fission rates across the entire core locations of Core-2. Note: The fluxes and the reaction rates are normalised to one neutron and one reaction respectively.

4.11 Discussions

Aside from demonstrating the feasibility of using homogenized cross section data in the Monte Carlo method, TRIMON also intends to offer robust core calculations to the end-users. As a result, TRIMON permits users to operate calculations with the consideration of local core burnups.

Fundamentally, the first step to integrating homogenized group data in the Monte Carlo method is simply to reduce the complexity of the stochastic simulation. On the other hand, simulating neutron transport processes in an actual reactor using a point-wise energy cross section data require a detailed reactor core model which requires more computational time. In most instances, a detailed core model is not necessary because the reflection of the entire core behaviour is more significant.

Authors in (Rabir et al., 2016) demonstrated that thousands of MCNP criticality calculation cycles were required to converge at a good estimation of k_{eff} values. The work done by (Wang et al., 2015) also emphasized on speeding up the convergence of source distribution and k_{eff} due to the similar slow convergence problem. Notably, TRIMON addressed this issue by introducing the homogenized cross section in the Monte Carlo method to reduce the spatial variation of neutron cross sections within the reactor core. Hence, it indirectly coarsens the spatial variation of the source distribution, thus relaxing the complexity of the source converging process. However, this attempt sacrifices the tally resolution where the highest resolution is only up to a unit cell level.

Conventional point-wise energy Monte Carlo method requires spatially dependent neutron cross section data, where the detail of cross section variations across the unit cell volume must be known. However, the case is different for homogenized unit cells. Recall that a homogenized unit cell is the spatial average of a heterogeneous unit cell, V, containing various material regions of different neutron cross sections. In this perspective, the volume integrated flux and the net reaction rate in V are the same in both heterogeneous unit cell and its equivalent homogeneous unit cell. As a result, the true k_{eff} value of both heterogeneous and homogeneous unit cells is preserved.

In fact, all sequences of collision events that occur within a heterogeneous unit cell volume are not statistically identical with the collision events that occur within the equivalent homogeneous unit cell volume. Moreover, all collision events within a homogeneous unit cell volume are virtual, however, these virtual collisions preserve the total neutron energy deposited inside the entire geometry of the unit cell. Plus, the integral of the neutron track lengths between these virtual collision events that occur within the entire unit cell volume is also preserved. Due to the implication of these virtual events, tallies should be strictly scored over the entire volume of the unit cell and scoring should not be done in the smaller sub-region of the unit cell. Fortunately, the neutron behaviour over the entire cell is of much greater importance, thus the implication of the virtual collision events is negligible.

References

A. Persic, Slavic, S., Ravnik, M., & Zagar, T. (1998). *TRIGLAV - A Program Package for Research Reactor Calculations*.

Brown, F. B. (2011). "K-effective of the World" and Other Concerns for Monte Carlo Eigenvalue Calculations. *Progress in Nuclear Science and Technology*. https://doi.org/10.15669/pnst.2.738

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

Chiesa, D., Clemenza, M., Pozzi, S., Previtali, E., Sisti, M., Alloni, D., … Sartori, A. (2016). Fuel burnup analysis of the TRIGA Mark II reactor at the University of Pavia. *Annals of Nuclear Energy*, *96*, 270–276. https://doi.org/10.1016/j.anucene.2016.06.008

Cho, N. Z., & Chang, J. (2009). Some outstanding problems in neutron transport computation. *Nuclear Engineering and Technology*, *41*(4), 381–390. https://doi.org/10.5516/NET.2009.41.4.381

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

Henry, R., Tiselj, I., & Snoj, L. (2017). CFD/Monte-Carlo neutron transport coupling scheme, application to TRIGA reactor. *Annals of Nuclear Energy*, *110*, 36–47. https://doi.org/10.1016/J.ANUCENE.2017.06.018

International Atomic Energy Agency (IAEA). (1980). A new approach to homogenization and group condensation. *Specialists' Meeting on Homogenization Methods in Reactor Physics*, *231*(November 1978), 303–322.

Leppänen, J. (2010). Performance of Woodcock delta-tracking in lattice physics applications using the Serpent Monte Carlo reactor physics burnup calculation code. *Annals of Nuclear Energy*, *37*(5), 715–722. https://doi.org/10.1016/J.ANUCENE.2010.01.011

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

Rabir, M. H. B., Mohamed Zin, M. R. B., Karim, J. B. A., Jalal Bayar, A. M. B., Usang, M. D. A., Mustafa, M. K. B., … Jalil, M. H. B. (2017). Neutronics calculation of RTP core. In *AIP Conference Proceedings* (Vol. 1799). https://doi.org/10.1063/1.4972907

Rabir, M. H., Md Zin, M. R., & Usang, M. D. (2016). Neutron flux and power in RTP core-15. *AIP Conference Proceedings*, *1704*(1), 50018. https://doi.org/10.1063/1.4940114

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

Ross, S., & Ross, S. (2013). Generating Discrete Random Variables. *Simulation*, 47–68. https://doi.org/10.1016/B978-0-12-415825-2.00004-8

Snoj, L., & Ravnik, M. (2008). Power peakings in mixed TRIGA cores. *Nuclear Engineering and Design*, *238*(9), 2473–2479. https://doi.org/10.1016/j.nucengdes.2008.02.005

Trkov, A., & Ravnik, M. (1994). Effective diffusion homogenization of cross sections for pressurized water reactor core calculations. *Nuclear Science and Engineering*, *116*(2), 86–95. Retrieved from http://www.scopus.com/inward/record.url?eid=2-s2.0-0028377664&partnerID=tZOtx3y1

Wang, K., Li, Z., She, D., Liang, J., Xu, Q., Qiu, Y., … Yu, G. (2015). RMC - A Monte Carlo code for reactor core analysis. *Annals of Nuclear Energy*, *82*, 121–129. https://doi.org/10.1016/j.anucene.2014.08.048

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

PART II:
TRIMON USER’S MANUAL

By M. R. Omar

Contents

1. Introduction

1.1 Overview

TRIMON is a software package that runs Monte Carlo simulation of neutron random walk movements inside a TRIGA reactor.

TRIMON helps physicists and students to analyse and visualize neutronic behaviour such as the effect of reactor power feedback on the core reactivity and the effect of nuclear fuel burnup on core reactivity. TRIMON could also be used to analyse the change in core reactivity after modifying the fuel configuration, including the fuel arrangements and the fuel type being used.

1.2 Licensing

The current version of TRIMON is free to use for educational and commercial purpose. However, all research works that use TRIMON must cite the following article:

- Omar, M. R., Karim, J. A., & Yoon, T. L. (2019). The development of a multigroup Monte Carlo code for TRIGA reactors. *Nuclear Engineering and Design*, 342.

2. Installing TRIMON

At the user level, TRIMON installation is an easy job since all component files including the required executables are automatically copied to the application data folder of the current user. The installation procedure only requires the user to run the installer package *trimon-installer.msi*. Please refer to the current TRIMON website on how to download the installer package.

2.1 TRIMON Installation Using *trimon-installer.msi*

The steps required to install TRIMON in your machine is given as follows:

- Step 1: Run *trimon-installer.msi*.
- Step 2: Go through the installation wizard by clicking the [***Next***] button.
- Step 3: Specify the location to install TRIMON component files by clicking the [***Browse***...] button. By default, TRIMON component files will be copied and saved in the user's application data folder (the *appdata* folder). Then, click the [***Next***] button.
- Step 4: Confirm the installation by clicking the [***Install***] button. If the user wishes to cancel the installation, click the [***Cancel***] button. Otherwise, click the [***Back***] button if the user wishes to modify the install location of TRIMON.

- Step 5: Click the [***Finish***] button to end the installation. By default, the *Launch TRIMON simulation package* checkbox is checked. This means that TRIMON program will be executed once the user clicks the [***Finish***] button.

2.2 Uninstalling TRIMON

To uninstall TRIMON, it can be done via the Windows Control Panel. Click the *Windows Start* button → search for *Control Panel* → under the *Programs* category, click *Uninstall a Program*. Next, a list of installed programs will appear, and the user is required to search for *TRIMON Simulation Package* in the list. Finally, by clicking *TRIMON Simulation Package*, the user will be directed to the official Windows uninstall wizard.

2.3 Installation Requirements

TRIMON Simulation Package works on any Windows operating system with a pre-installed .NET Framework 2.0 package. If .NET Framework 2.0 is not installed in the user's machine, the user is required to visit the official Microsoft website to get the references on how to install .NET Framework 2.0. In general, .NET Framework 2.0 is readily pre-installed in Windows 7, 8, 8.1 and 10. Guidelines on installing .NET Framework 2.0 in your Windows machine can be found at https://dotnet.microsoft.com/download/dotnet-framework.

3. Using TRIMON Simulation Package

In general, TRIMON Simulation Package is a collection of programs that run various tasks such as

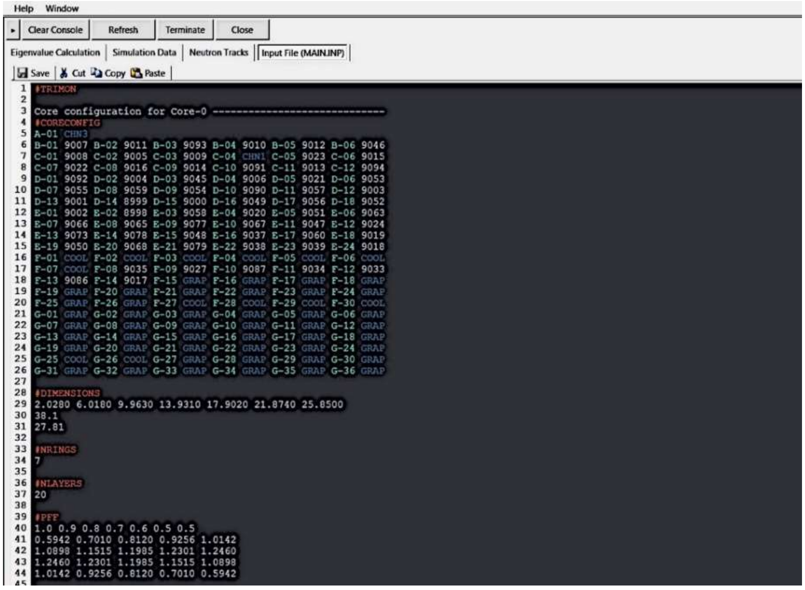
(a) Preparation of simulation input files using *trimon-gui.exe*, the main graphical user interface.
(b) Homogenized neutron cross section table (*.txs* file) preparation using *LIBREADER.EXE*.
(c) Monte Carlo simulation using *HGMC.EXE*.
(d) Fuel Burnup Calculation using *HGMC.EXE*.

TRIMON requires two input files to work and these input files contain the information of the Monte Carlo simulation problem defined by the user. The first input file is a formatted text file designated as *main.inp*. In *main.inp* file, various important simulation definitions are stored, including the geometrical size of the reactor, the arrangement of fuel channels and various non-fuel channels such as the control rod location, coolant, and instrumentation facilities. Plus, the simulation settings are also kept in *main.inp*, for example, the neutron energy group structure, reactor power, coolant temperature, fuel power-temperature correlation parameters, burnup correlation and criticality calculation parameters (number fission cycles, number of inactive cycles and the initial guess of k_{eff}).

The second input file is designated as *fuel_inventory.inp*. It contains a list of the UZrH fuels available that will be used for the current simulation job. Practically, *fuel_inventory.inp* should contain the list of all fuels that are available in the fuel storage facility of the reactor. The user can transfer a fuel pin from the fuel inventory to the reactor core, or vice versa. The characteristic information of each fuel pin are stored in every single line of *fuel_inventory.inp*. Each line must contain the following characteristic information – the fuel identification number (an integer 0001-9999), the fuel type (FE08, FE12 or FE20), the mass of Uranium in grams, the fuel enrichment in percent, fuel burnup in MWd, fuel utilization in per cent, and the initial burnup state of the fuel in per cent. However, the users are not encouraged to modify any of these parameters, except the fuel pin's identification number and its corresponding fuel type.

3.1 Preparation of the Main Input File – *main.inp*

To modify *main.inp*, on *TRIMON Dashboard* window, click [***Input File (MAIN.INP)***] tab. A text editor with colourful text will appear and the user can edit the pre-existing content of *main.inp*. To save the edited *main.inp* file, click the [***Save***] button on the top toolbar of the editor. Once the edited content of *main.inp* has been saved, the [***Save***] button appears disabled. If the content of *main.inp* is further modified using the text editor and yet to be saved, the [***Save***] button will be enabled again.



```
#TRIMON

Core configuration for Core-0 ----------------------------
#CORECONFIG
A-01 CHN3
B-01 9007 B-02 9011 B-03 9093 B-04 9010 B-05 9012 B-06 9046
C-01 9008 C-02 9005 C-03 9009 C-04 CHN1 C-05 9023 C-06 9015
C-07 9022 C-08 9016 C-09 9014 C-10 9091 C-11 9013 C-12 9094
D-01 9092 D-02 9004 D-03 9045 D-04 9006 D-05 9021 D-06 9053
D-07 9055 D-08 9059 D-09 9054 D-10 9090 D-11 9057 D-12 9003
D-13 9001 D-14 8999 D-15 9000 D-16 9049 D-17 9056 D-18 9052
E-01 9002 E-02 8998 E-03 9058 E-04 9020 E-05 9051 E-06 9063
E-07 9066 E-08 9065 E-09 9077 E-10 9067 E-11 9047 E-12 9024
E-13 9073 E-14 9078 E-15 9048 E-16 9037 E-17 9060 E-18 9019
E-19 9050 E-20 9068 E-21 9079 E-22 9038 E-23 9039 E-24 9018
F-01 COOL F-02 COOL F-03 COOL F-04 COOL F-05 COOL F-06 COOL
F-07 COOL F-08 9035 F-09 9027 F-10 9087 F-11 9034 F-12 9033
F-13 9086 F-14 9017 F-15 GRAP F-16 GRAP F-17 GRAP F-18 GRAP
F-19 GRAP F-20 GRAP F-21 GRAP F-22 GRAP F-23 GRAP F-24 GRAP
F-25 GRAP F-26 GRAP F-27 COOL F-28 COOL F-29 COOL F-30 COOL
G-01 GRAP G-02 GRAP G-03 GRAP G-04 GRAP G-05 GRAP G-06 GRAP
G-07 GRAP G-08 GRAP G-09 GRAP G-10 GRAP G-11 GRAP G-12 GRAP
G-13 GRAP G-14 GRAP G-15 GRAP G-16 GRAP G-17 GRAP G-18 GRAP
G-19 GRAP G-20 GRAP G-21 GRAP G-22 GRAP G-23 GRAP G-24 GRAP
G-25 COOL G-26 COOL G-27 GRAP G-28 GRAP G-29 GRAP G-30 GRAP
G-31 GRAP G-32 GRAP G-33 GRAP G-34 GRAP G-35 GRAP G-36 GRAP

#DIMENSIONS
2.0280 6.0180 9.9630 13.9310 17.9020 21.8740 25.8500
38.1
27.81

#NRINGS
7

#NLAYERS
20

#PFF
1.0 0.9 0.8 0.7 0.6 0.5 0.5
0.5942 0.7010 0.8120 0.9256 1.0142
1.0898 1.1515 1.1985 1.2301 1.2460
1.2460 1.2301 1.1985 1.1515 1.0898
1.0142 0.9256 0.8120 0.7010 0.5942
```

Figure 3.1-1: The *main.inp* text editor.

Each simulation information is represented using a formatted array of strings segments. Each segment comprises a few lines containing various parameters and the first line of a segment will always begin with a keyword. Also, the first character of a keyword must be the hashtag symbol, '#', i.e. `#POWER`, `#TCOOL` and etc. In the text editor, keywords will appear with red characters. For example, if the user wishes to set the reactor coolant temperature to 300K, the following segment must be added to *main.inp*:

```
#TCOOL
300.0
```

In summary, TRIMON requires thirteen (13) different data segments in order to work and the summary of the syntax of each segment is given in Table 3.1-1. Most importantly, *main.inp* will always begin with `#TRIMON` segment on its first line:

```
#TRIMON
Comments can be added here…

#KEYWORD1
…

#KEYWORD2
…
```

TRIMON reads *main.inp* line by line, which means if there exist two segments with a similar keyword, only the first found occurrence is considered. The remaining duplicates of the segment are ignored by TRIMON. Therefore, it is important to ensure that no duplicates of segments exist in *main.inp* to avoid ambiguity.

Table 3.1-1: A list of syntaxes for all data segments defined in TRIMON.

Segments	Syntax	Description
KRUN	#KRUN [seed] [nhistories] [ncycle] [nskip] [keffguess]	To specify k-eff calculation parameters. [seed] is an integer number between 0-999999. [nhistories] is an integer number between 3000-6000000 specifying the number of neutron histories to be simulated in a single fission cycle. [ncycle] is an integer number > 0 specifying the total number of fission cycles to be simulated throughout the criticality calculation. [nskip] is the number if skip cycle to allow the simulation converges before scoring tallies. **Note**: [nskip] < [ncycle]. is the initial guess value of the effective multiplication factor.
DIMENSIONS	#DIMENSIONS [rA] [rB] [rC] … [core-critical-height]	To specify reactor core ring radius and core critical height [cm]. [rA],[rB], …, [rC] are the outer radiuses of ring A, ring B, …, ring G, respectively. [core-critical-height] is the core critical height of the simulated reactor core.
NRINGS	#NRINGS [number-of-core-rings]	To specify the number of core rings. Only two valid values are accepted – 6 or 7.
NLAYERS	#NLAYERS [number-of-core-layers]	To specify the number of core layers.
PFF	#PFF [arA] [arB] … [ah1] [ah2] …	To specify the initial guess of radial power form factors for each core ring and axial power form factors for each core layers. See equation (4.3), *Chapter 4: Theory and Laws Implemented In TRIMON*. [arA],[arB], …,[arG] are the radial power form factors of Ring A, Ring B, …, Ring G, respectively. [ah1],[ah2],… are the axial power form factors of Layer 1, Layer 2 , …, respectively.

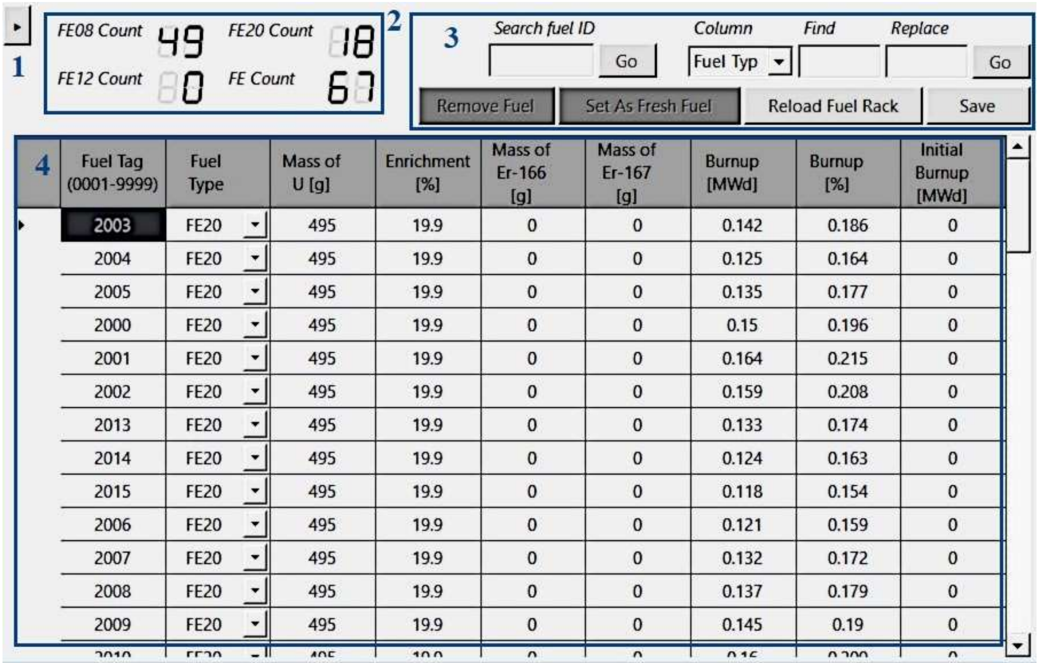
MACROGROUP	#MACROGROUP [ngroup] [g1] [g2] …	To specify the total number of neutron energy macro-groups in the simulation; and to specify the last micro-group number for each macro-group. In TRIMON, there are a total of 32 micro-groups. [ngroup] is the total number of neutron energy groups. [g1] [g2] are the last micro-group numbers. There should be [ngroup]+1 last micro-group number in the list.
TCOOL	#TCOOL [temperature]	To specify the coolant water temperature in [K].
POWER	#POWER [nominal-power]	To specify the nominal core power in [kW].
TEMPREL	#TEMPREL [n08] [Pmin] [Pmax] [a0] [a1] … [a_n08] [n12] [Pmin] [Pmax] [a0] [a1] … [a_n12] [n20] [Pmin] [Pmax] [a0] [a1] … [a_n20]	To specify the coefficients of the fuel cell temperature relation polynomial given by equation (4.3), *Chapter 4: Theory and Laws Implemented In TRIMON*. Note that [n08], [n12] and [n20] are the degree of the polynomial; [Pmin] and [Pmax] are the fuel cell power boundaries defined by the temperature relation; and [a0], [a1]… are the corresponding coefficients.
BURNUP	#BURNUP [dt] ST8 [nj] [beta1] [beta2] … ST12 [nj] [beta1] [beta2] … ST20 [nj] [beta1] [beta2] …	To specify the burnup increment [dt] in [days]; to specify the degree of power series, [nj], and the beta coefficients, [beta1], [beta2] …, in equation (4.2), *Chapter 4: Theory and Laws Implemented In TRIMON.* for each different types of fuel elements.

3.2 Preparation of the Fuel Inventory Input file – *fuel_inventory.inp*

Fortunately, the preparation of *fuel_inventory.inp* requires no pre-defined syntaxes and its content can be modified using the given graphical user interface. To modify *fuel_inventory.inp*: activate the *Core Configuration* window by either clicking the title bar of the window or by simply selecting [***Window***] → [***Core Configuration***] on the main menu strip. Once the *Core Configuration* window is focused, several new options will appear on the main menu strip, i.e. [***Loading***], [***Fuel***], [***Simulation***] and [***Configuration Display***]. Next, choose [***Fuel***] → [***Fuel Inventory Input***] on the main menu strip. Lastly, the *Fuel Inventory Input* window will appear. The user can modify the fuel properties of each fuel pin in the current inventory by the use of the *Fuel Inventory Input* window. The summary of the available controls in the *Fuel Inventory Input* window is illustrated in Fig. 3.2-1.

- The role of the [►] button (1) is to maximize the *Fuel Inventory Input* window so that the user gains a wider view of the window.
- The *Fuel Count* panel (2) displays the number of fuel pina stored in the current fuel inventory of each/all fuel type(s) including the FE08, FE12 and FE20.

- The *Inventory Management* Panel (3) contains a set of controls which enable the user to search fuel pin(s), remove fuel pin(s), setting a fuel pin as a fresh fuel, reloading fuel pins rack and save the current fuel pins inventory.



FE08 Count 49
FE20 Count 18
FE12 Count 0
FE Count 67

Fuel Tag (0001-9999)	Fuel Type	Mass of U [g]	Enrichment [%]	Mass of Er-166 [g]	Mass of Er-167 [g]	Burnup [MWd]	Burnup [%]	Initial Burnup [MWd]
2003	FE20	495	19.9	0	0	0.142	0.186	0
2004	FE20	495	19.9	0	0	0.125	0.164	0
2005	FE20	495	19.9	0	0	0.135	0.177	0
2000	FE20	495	19.9	0	0	0.15	0.196	0
2001	FE20	495	19.9	0	0	0.164	0.215	0
2002	FE20	495	19.9	0	0	0.159	0.208	0
2013	FE20	495	19.9	0	0	0.133	0.174	0
2014	FE20	495	19.9	0	0	0.124	0.163	0
2015	FE20	495	19.9	0	0	0.118	0.154	0
2006	FE20	495	19.9	0	0	0.121	0.159	0
2007	FE20	495	19.9	0	0	0.132	0.172	0
2008	FE20	495	19.9	0	0	0.137	0.179	0
2009	FE20	495	19.9	0	0	0.145	0.19	0

Figure 3.2-1: The *fuel_inventory.inp* editor. (**1**) *Window Resize* button; (**2**) *Fuel Count* panel; (**3**) *Inventory Management* panel; and fuel pins rack (**4**).

3.2.1 Adding a New Fuel to the Fuel Inventory

To add a new fuel, scroll to the end of the rack (**4**) and double click the empty cell at the end of the *Fuel Tag* column to activate the edit mode of the cell. In the *Fuel Tag* cell, key in the identification number of the new fuel pin (an integer ranging from 0001 to 9999). Next, on the same row of the fuel rack, select the appropriate fuel type listed in the dropdown control embedded in the *Fuel Type* cell. Once the fuel type is selected i.e. FE08/FE12/FE20, the remaining columns on the same row will automatically be filled with the pre-defined values corresponding to the selected fuel type. If the new fuel pin is a fresh fuel (zero burnup level with 0% utilization), these remaining columns should be left with the pre-defined values. In certain cases, the user may want to modify these fields accordingly. For example, if the burnup level of the fuel pin is 5%, the user is required to modify the corresponding *Burnup [%]* cell field to 5.0.

3.2.2 Removing Existing Fuels from the Fuel Inventory

To remove one or more of the existing fuel pins in the fuel inventory rack, CTRL-click one or more rows containing the fuels to be removed → click the [***Remove Fuel***] button. Most importantly, the entire fuel row(s) must be selected to enable the [***Remove Fuel***] button. If there is no fuel row selected, the [***Remove Fuel***] button is disabled by default.

3.2.3 Setting an Existing Fuel as a Fresh Fuel

To set one or more fuel pin(s) as fresh fuel, CTRL-Click one or more rows containing the desired fuel pin(s) → click the [***Set As Fresh Fuel***] button. Most importantly, the entire fuel row(s) must be selected to enable the [***Set As Fresh Fuel***] button. If there is no fuel row selected, the [***Set As Fresh Fuel***] button is disabled by default.

3.2.4 Reloading the Fuel Inventory Table

For some instances, the user may want to reload the fuel rack (**4**) so that the fuel rack only displays the current fuel inventory list. To reload the fuel rack, click the [***Reload Fuel Rack***] button. **Caution**: Any unsaved new fuel pins or unsaved fuel pin edits will be discarded after reloading the fuel rack.

3.2.5 Saving Changes Made to the Fuel Inventory

To save the newly added fuel pin(s) or the edited fuel pin(s), click the [***Save***] button.

3.2.6 Search Fuels and Replacing Fuel Information

To search for one or more fuel rows containing the information of a fuel pin with a specific identification number, type the fuel identification number in the *Search Fuel ID* textbox → click the [***Go***] button. The system will automatically highlight the entire row with the specified fuel pin identification number.

To search for the fuel row(s) by property type and replace the current property value with a new value: select the desired property type listed in the *Column* dropdown list → type the desired property value to be searched → type the new property value in the *Replace* textbox → click the [***Go***]. **Caution**: this will replace the property value of all found rows.

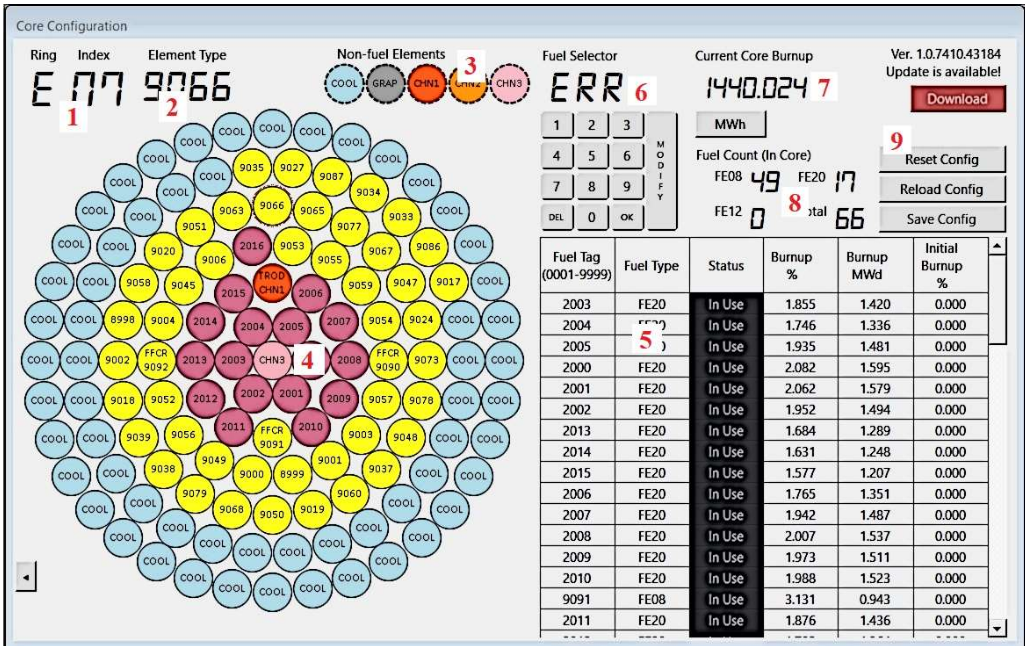
3.3 Preparation of the Core Configuration

Principally, TRIMON is meant to be used to simulate neutron transport problems in a TRIGA reactor core. Therefore the material compositions of the reactor core are predefined in the code. Plus, the user does not require to identify the material compositions in the code input. Most importantly, TRIMON recognizes core material compositions according to the core configuration, where each core channels are designated to accommodate various types of elements. A core channel may comprise of a fuel element or any various types of non-fuel element. The list of elements recognized by TRIMON is given in Table 3.3-1.

Technically, the core configuration of the problem can be set by modifying the `#CORECONFIG` segment of the main.inp. However, such a job may cost an extra bit of time and effort, which in fact making the process of setting up the problem definition troublesome. To ease the process of assigning an appropriate core channel (i.e. FE08/FE12/FE20/GRAP/COOL) to each core sites (i.e. A-01, B-01, B-02 …), the user is encouraged to modify the core configuration via the *Core Configuration* window, see Fig 3.3-1.

Table 3.3-1: List of channel elements supported by TRIMON.

Channel Element	Description
`FE08, FE12, FE20`	Enriched UZrH1.6 fuel element with 8.5%wt, 12%wt and 20%wt of uranium respectively; stainless steel (SS304) cladding. The fuel element is surrounded by coolant water.
`CHN1`	Irradiation Channel Type-I: An empty Al tube. CHN1 can be used as an approximate model of a transient control rod.
`CHN2`	Irradiation Channel Type-II: Half void, half water in Al tube.
`CHN3`	Irradiation Channel Type-III: Full water in Al tube. Can be used to represent the central thimble of the reactor core.
`GRAP`	Graphite in Al tube.
`COOL`	Coolant water.



Ver. 1.0.7410.43184
Update is available!

Fuel Selector: ERR

Current Core Burnup: 1440.024 MWh

Fuel Count (In Core)
FE08 49 FE20 17
FE12 0 Total 66

Fuel Tag (0001-9999)	Fuel Type	Status	Burnup %	Burnup MWd	Initial Burnup %
2003	FE20	In Use	1.855	1.420	0.000
2004	FE20	In Use	1.746	1.336	0.000
2005	FE20	In Use	1.935	1.481	0.000
2000	FE20	In Use	2.082	1.595	0.000
2001	FE20	In Use	2.062	1.579	0.000
2002	FE20	In Use	1.952	1.494	0.000
2013	FE20	In Use	1.684	1.289	0.000
2014	FE20	In Use	1.631	1.248	0.000
2015	FE20	In Use	1.577	1.207	0.000
2006	FE20	In Use	1.765	1.351	0.000
2007	FE20	In Use	1.942	1.487	0.000
2008	FE20	In Use	2.007	1.537	0.000
2009	FE20	In Use	1.973	1.511	0.000
2010	FE20	In Use	1.988	1.523	0.000
9091	FE08	In Use	3.131	0.943	0.000
2011	FE20	In Use	1.876	1.436	0.000

Fig. 3.3-1: The *Core Configuration* window. (1) ring-index display (2) element type display (3) non-fuel element selector (4) reactor core channel display (5) fuel inventory rack (6) fuel element selector (7) core total burnup display (8) in-core fuel count display (9) configuration management buttons.

The ring-index display shows the channel site identification (i.e. A-01, B-01 … G-36) of the currently selected channel. A selected channel in the reactor core channel display (**4**) is indicated with a red border around it. If a channel in (**4**) is selected, the ring-index display (**1**) and the element type display will show the corresponding site identification and the corresponding element type, respectively.

3.3.1 Inserting a non-fuel element into a core channel

To help the user, it is advised to toggle the labelling of the reactor core channel display to *Channel ID* display mode. In *Channel ID* display mode, the channel site identification of each core channel is displayed. This task is optional. An experienced user may not need such a task since the user is already familiar with the channel site identification position on the reactor core channel display. To enable the *Channel ID* display mode, select [***Configuration Display***] → [***Display Text***] → [***Channel ID***] on the main menu strip. **Tips**: TRIMON also provides several configuration display options to the user. For example, if the user wishes to display the fuel burnup value on all fuel channels, select [***Configuration Display***] → [***Display Text***] → [***Burnup***]. In the *Burnup* display mode, each fuel channel is labelled with its corresponding fuel burnup level in per cent.

For instance, our objective is to insert a graphite element into the core channel site G-36. Inserting a non-fuel element into a core channel can be done in two ways. The first is by simply clicking the G-36 core channel → right-click → select [***Insert GRAP***] on the context menu strip. The second is by toggling the *Insert GRAP* mode by clicking the *GRAP* element on the fuel element selector (**3**) → click the G-36 core channel → disable the *Insert GRAP* mode by re-clicking the *GRAP* element on the fuel element selector. The advantage of the second method is that it enables multiple non-fuel element insertions.

3.3.2 Inserting a fuel element into a core channel

Suppose that the user wishes to insert an 8.5%wt UZrH fuel pin with fuel pin identification number 9015 into the core channel site B-02. Here, assume that fuel pin 9015 has already been stored in the fuel inventory record. To add a new fuel pin into the fuel inventory record, please refer Section 3.2.1. Using the fuel selector (6) on the core configuration window, click the [***9***] button → click the [***0***] button → click the [***1***] button → click the [***5***] button → and click the [***OK***] button. The system will automatically highlight fuel 9015 in the fuel inventory rack (**5**). To delete a mistakenly pressed digit, click the [***Del***] button. Alternatively, with the *Core Configuration* window in focus, press the keyboard buttons: 9, 0, 1 and 5 instead of pressing the numeric buttons on the fuel selector (**6**). **Note**: If the specified fuel identification number could not be found in the fuel inventory rack, the fuel selector (**6**) panel will display *ERR*. The selected fuel can be inserted into the core provided that the selected fuel appears to available in the fuel inventory rack (**5**). The third column of the selected table row will indicate whether the fuel is *IN-USE* or *AVAILABLE*.

Next, select the core channel site B-02 on the reactor core channel display until 'B-06' appear on the ring-index display (**1**) → right-click the highlighted fuel inventory rack table row → click [***Insert Fuel 9015 into channel B-02***].

3.3.3 Removing a fuel pin

A fuel pin can be removed from a reactor channel (**4**) by selecting the fuel pin's channel site → replace the channel site with a coolant element (*COOL*) using the method outlined in Section 3.3.1.

3.3.4 On-the-fly modifications of fuel properties

Traditionally, the user can modify fuel pin properties by selecting [***Fuel***] → [***Fuel Inventory Input***] from the main menu strip. Alternatively, it is possible to do such modifications by using the fuel selector → select the desired fuel using the fuel selector (**6**) → click the [***MODIFY***] button on the fuel selector (**6**). A *Fuel Inventory Input* window will appear with the specified fuel pin information being pre-highlighted on the fuel inventory rack. The user may proceed for the modification by using the steps outlined in Section 3.2.6.

3.3.5 Displaying the total core burnup

The total core burnup of the current state is displayed on the total core burnup display (**7**). This value is equals to the sum of all numbers listed on the fifth column of the fuel inventory table (**5**). The total core burnup unit can be modified into MWd/MWh/kWh by using unit toggle button on the total core burnup display (**7**).

3.3.6 In-Core Fuel Count

The in-core fuel count display (**8**) gives the information on the number of fuels that are being loaded into the reactor core according to the following categories:

(a) FE08 Count – The number of 8.5%wt UZrH fuels loaded in the reactor core
(b) FE12 Count – The number of 12%wt UZrH fuels loaded in the reactor core
(c) FE20 Count – The number of 20%wt UZrH fuels loaded in the reactor core
(d) Total – The total number of UZrH fuels loaded in the reactor core

This is an important feature that helps the user to ensure the correct number of mixed fuels are loaded into the reactor core.

3.3.7 Saving fuel configuration

If the user has finished setting up the fuel configuration, it is important to save the configuration in the *main.inp* file. To do this, click the [***Save Config***] button (**9**). **Caution**: If the core configuration is not saved, the future simulation will not consider any newly modified core configuration. Plus, the unsaved core configuration might not be able to get recovered in the future, unless the user has clicked the [***Save Config***] button.

3.3.8 Resetting/Reloading core configuration

If the user wants to reset the reactor core, that is, to remove all available fuel pins and non-fuel elements from the reactor core: press the [***Reset Config***] button (**9**). This will replace all core channel sites with *COOL* elements.

For some instance, the user needs to cancel his/her unsaved core configuration modifications due to some reasons. In order to reload the reactor core with the previously saved core configuration, click the [***Reload Config***] button (**9**).

3.4 Running the Monte Carlo Simulation

3.4.1 Setting up the Monte Carlo simulation parameters

In this section, it is important that the user must have a good knowledge of Monte Carlo criticality calculation. To learn more about Monte Carlo criticality calculation, please refer *Chapter 4: Theory and Laws Implemented in TRIMON* from the theoretical reference of TRIMON. To run the Monte Carlo simulation of the problem defined by the user in the *main.inp* and *fuel_inventory.inp* files, the following MC parameters must be understood carefully:

(a) Random Number Seed – A seed is a positive integer ranging from 0-9999999. The user can choose any number within this range. To ensure reproducibility of the simulation, the user is encouraged to use the same random number seed for all related computational problems. This seed value will be used by the random number generator module of HGMC.EXE to generate random number sequences during the Monte Carlo simulation

(b) Total Neutron Histories – This is the total number of neutrons to be simulated in a single fission cycle. The user must prescribe an integer number ranging from 3000-6000000.

(c) Total Number of Fission Cycle – This is the total number of fission cycle to be repeated during the course of criticality calculation. The user must prescribe an integer number ranging from 1-99999.

(d) Total number of Skip Cycles – This is the total number of idle fission cycles, such that the simulation tallies are not accumulated during these skip cycles. The user must prescribe an integer value greater than 0 and less than the total number of fission cycle (c).

(e) Initial Guess of k_{eff} – In most criticality calculations, it is compulsory to prescribe the initial guess of k_{eff} during the starting point of the calculation. The user is advised to prescribe a value which is close to the final calculated k_{eff}. This number is usually between 0.0 to 2.0. The closer the initial guess of k_{eff} to the final calculated k_{eff}, the faster the convergence of the calculation.

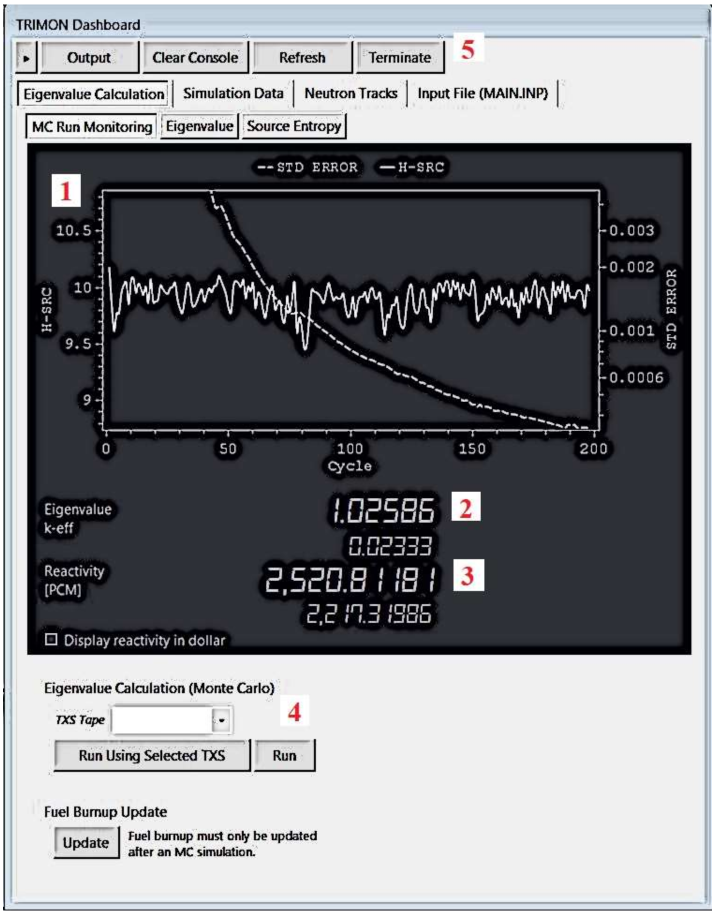
These MC calculation parameters can be set up by modifying the `#KRUN` segment of the *main.inp* file. To do so, please follow the steps outlined in Section 3.1. For instance, the following #KRUN segment

```
#KRUN
920909 100000 2000
0.8
```

indicates that the random number seed is 920909, the total number of neutron histories is 100000, the total number of fission cycles is 2000, the total number of skip cycles is 500 and the initial guess of k_{eff} is 0.8.

3.4.2 Eigenvalue Calculation Tab

The *Eigenvalue Calculation* tab consists of all relevant controls that assist the user on executing Monte Carlo criticality calculations. *Eigenvalue* tab can be found on the *TRIMON Dashboard* window.



Fuel burnup must only be updated after an MC simulation.

Fig. 3.4-1: Basic Criticality Calculation (Eigenvalue Calculation) controls in TRIMON Dashboard window. (1) Fission source entropy (*H-SRC*) and standard error (*STD ERROR*) monitoring display, (2) *Eigenvalue* display (3) core excess reactivity display (4) Monte Carlo simulation execution control, and (5) Monte Carlo simulation job termination.

To begin the execution of Monte Carlo criticality calculation, on the Eigenvalue Calculation (Monte Carlo) panel (4), click the [***Run***] button. This will invoke the *LIBREADER.EXE* program to generate the homogenized neutron cross section data table (*.txs* file) and then the neutron cross section table will be used by *HGMC.EXE* to execute the Monte Carlo criticality calculation. If the user wishes to use the pre-existing neutron cross section data table: Select the desired *.txs* file from the *TXS Tape* drop down list → click the [***Run Using Selected TXS***] button. In this mode, the *LIBREADER.EXE* program is not invoked by the system since no new neutron cross section table will be generated because a pre-existing *.txs* is being used. However, the *TXS Tape* dropdown control only lists the *.txs* files that are located within the same directory of *HGMC.EXE*. If the user wishes to run the critical height search mode, CTRL-click the [***Run***] button. In the critical height search mode, TRIMON will search for the core height that yields criticality when the current core configuration set by the user is used. This step is important during the first-time calculation setup for a specific reactor. A different reactor has a different critical height therefore TRIMON needs to make buckling adjustment via the adjustment of the core height. To run the critical height search mode using an existing *.txs* tape, CTRL-click the [***Run Using Selected TXS***] button.

During the progress of the criticality calculation, the effective multiplication factor calculated during the previous fission cycle is displayed at the *Eigenvalue* display (2). Therefore, the user will notice that the displayed value keeps on changing from one fission cycle to another. Also, the corresponding core excess reactivity is also kept updated from one fission cycle to another in the *Reactivity* display (**3**). The user can also evaluate the fission source convergence trend and the current k-eff standard error by monitoring the live plot of the fission source entropy (*H-SRC*) and standard error (*STD ERROR*) (**1**).

3.4.3 Terminating the current running simulation

The user can also force to terminate the Monte Carlo criticality calculation by clicking the [**T*erminate***] button (**5**). However, the user is encouraged to complete the calculation since terminating a Monte Carlo simulation prematurely leads to an incomplete display of simulation results. Therefore, it is best for the user to thoroughly finalize the problem definition via *main.inp* and *fuel_inventory.inp* before attempting to run the simulation.

3.4.4 Burnup Calculation

Burnup calculation will be performed after TRIMON has completed the Monte Carlo criticality calculation. At the end of the criticality calculation, the fuel power tally result is compiled and stored in *PDIST.OUT* file. The burnup module will then update the current burnup values for each of the fuels by using the fuel power distribution data stored in *PDIST.OUT*. The burnup calculation may also require the operational time interval of the current core in days, and this must be prescribed by the user via the `#BURNUP` segment of the *main.inp* file. For instance, the following `#BURNUP` segment indicates that the operational time interval of the current core is 100.0 days.

```
#BURNUP
100.0
FE08 3 3.33 -0.0102  -0.0000525
FE12 3 2.30 -0.00570 -0.00000748
FE20 3 1.31 -0.00263  0.00000520
```

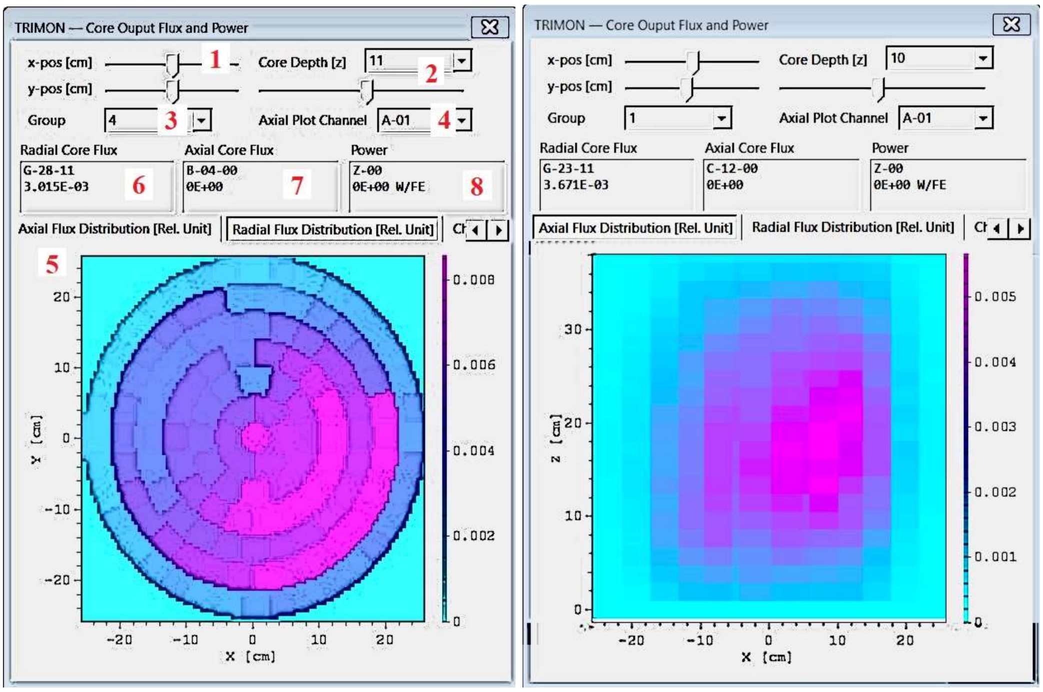
At the end of the burnup calculation, *fuel_inventory.out* output file is generated. The content of *fuel_inventory.out* is similar to *fuel_inventory.inp*, however, the former contains the updated burnup level of each fuels. If the user wish to re-do the burnup calculation using a different operational time interval, the user is required to update the operational time interval defined in the `#BURNUP` segment → return to *Eigenvalue Calculation* tab → click the [***Update***] button under the *Burnup Calculation* panel (**6**).

3.5 Simulation Output

The simulation output is a set of information measured during the course of the Monte Carlo simulation. In TRIMON, a set of neutronic quantities were tallied and accumulated during the criticality calculation. These neutronic quantities include:

(a) Neutron Flux Distribution
(b) Full Core Power Distribution
(c) Reaction Rates

Once the criticality calculation ended, the corresponding simulation output files will be generated by the system and these output files are ready to get evaluated by TRIMON. Suppose that the user has executed the criticality calculation and the simulation has ended without any issue. To view the simulation results, focus on *TRIMON Dashboard* window → click the [***Output***] button → *Simulation Output* window will appear in the interface.



TRIMON — Core Ouput Flux and Power

x-pos [cm]
y-pos [cm]
Core Depth [z]: 11
Group: 4
Axial Plot Channel: A-01

Radial Core Flux	Axial Core Flux	Power
G-28-11 3.015E-03	B-04-00 0E+00	Z-00 0E+00 W/FE

Axial Flux Distribution [Rel. Unit] | Radial Flux Distribution [Rel. Unit] | Ch

TRIMON — Core Ouput Flux and Power

x-pos [cm]
y-pos [cm]
Core Depth [z]: 10
Group: 1
Axial Plot Channel: A-01

Radial Core Flux	Axial Core Flux	Power
G-23-11 3.671E-03	C-12-00 0E+00	Z-00 0E+00 W/FE

Axial Flux Distribution [Rel. Unit] | Radial Flux Distribution [Rel. Unit] | Ch

Fig. 3.5-1: Basic simulation output controls *in Core Output Flux and Power* window. (1) Plotting plane adjustor (2) core depth adjustor (3) neutron energy group selector (4) channel selector (5) plotting tab control (6) radial flux value display (7) axial flux value display (8) fuel power display.

3.5.1 *Viewing the Axial Flux Distribution*

TRIMON has the capability of computing the axial flux distribution of a TRIGA reactor core. This means that the flux profile along the height of the reactor can be analysed by the user for better analysis of the neutronic behaviour inside the reactor. To view the axial flux distribution plot, select the *Axial Flux Distribution [Rel. Unit]* tab (5).

By default, the plot plane is the *xz*-plane at $y = 0$. The user can translate the *xz*-plane position along the *y*-axis by sliding the *y-pos [cm]* trackbar on the plotting plane adjustor panel (1). The plot will be automatically updated according to the *y* value given by the trackbar. Also, sliding the *y-pos [cm]* trackbar will reset the *x-pos [cm]* trackbar value to $x = 0$ and vice versa. Similarly, if the user wishes to set the plot plane to the *yz*-plane at $x = 3.0$, the user needs to adjust the *x-pos [cm]* trackbar until the label on the status bar shows $x = 3.0$.

Note also that when the mouse hovers on the plot view (5), the *Axial Core Flux* display will show the flux value at the mouse position. Here, the display will show the location identification tag in the form of {*Channel ID*}-{*Layer*} where {*Channel ID*} is the channel identification (i.e. A-01, B-01, …, G-36), {*Layer*} is a two-digit number representing the core layer identification number.

3.5.2 *Viewing the Radial Flux Distribution*

TRIMON also has the capability of computing the radial flux distribution of a TRIGA reactor core. This means that the flux profile along the radius of the reactor can be analysed by the user. To view the radial flux distribution plot, select the *Radial Flux Distribution [Rel. Unit]* tab on the plotting tab control (5).

By default, the plot plane is the *xy*-plane at *z* = 0. The user can also translate the *xy*-plane position along the *z*-axis by sliding trackbar on the *Core Depth* adjustor panel (**2**). The plot will be automatically updated according to the *z* value given by the trackbar. The current value of *z* in cm is displayed on the status strip.

Note also that when the mouse pointer hovers on the plot view (**5**), the *Radial Core Flux* display will show the flux value at the mouse position. The display will also show the location identification tag in the form of {*Channel ID*}-{*Layer*} where {*Channel ID*} is the channel identification (i.e. A-01, B-01, …, G-36), {*Layer*} is a two-digit number representing the core layer identification number.

3.5.3 Viewing the Channel Axial Flux Distribution

TRIMON also features the capability of visualizing the flux variation along a single channel. For instance, this feature allows the user to analyse the flux distribution along the fuel length of a fuel pin loaded into, say, the channel B-06. To view the channel axial flux distribution plot, select the *Channel Axial Flux Distribution [Rel. Unit]* tab on the plotting tab control (**5**). The user can view the axial flux of various channels by simply selecting the channel site identification listed in *Channel Selector* drop down list (**4**).

3.5.4 Viewing the Power Distribution

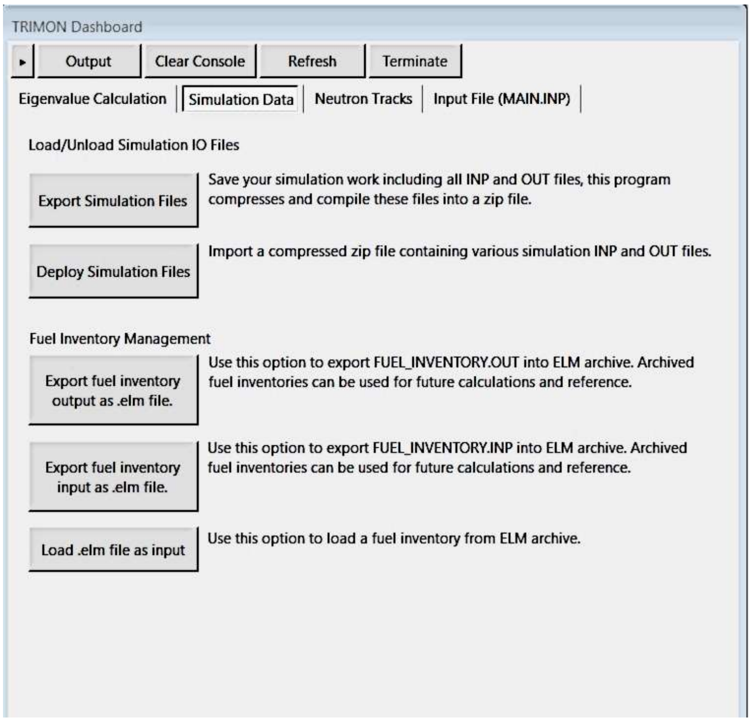
It is also possible to analyse the fuel power of fuels that were loaded into the TRIGA reactor core using TRIMON. This means that the fuel power distribution profile along the radius of the reactor can be analysed by the user for better analysis of the fuel economy. To view the power distribution plot, select *Power Distribution [Rel. Unit]* tab on the plot view tab control (**5**).

By default, the plot plane is the *xy*-plane. Note also that when the mouse hovers on the plot view (**5**), the *Power* display will flash the power per fuel element (per fe) value at the mouse curser position. The display will also show the location identification tag in the form of {*Channel ID*}-{*Layer*} where {*Channel ID*} is the channel identification (i.e. A-01, B-01, …, G-36), {*Layer*} is a two-digit number representing the core layer identification number. By default, the fuel power is expressed in the unit of Watt per fuel element (W/fe).

3.5.5 Viewing the Reaction Rate

The reaction rate of each core channel is displayed on the *Reaction Rate* tab on the plot view tab control (**5**). Here, a column bar chart is utilized where each column represents a single core channel. In the reaction rate plot, the reaction rate of all core channels are included, therefore, if the user wishes to view the reaction rate value of a specific core channel, say, A-01, the user may have to mouse left-click through all column bars until the tooltip display "A-01 …". The reaction rate is normalised to one reaction. By default, TRIMON will plot the total reaction rate. To select a different type of reaction, click the *Reaction Types* drop down list an select the desired reaction type. The user can also scale the *y*-axis of the plot by typing the maximum value of the axis in *Set Max* textbox. To refresh the plot, click the [***Refresh***] button.

3.6 Managing Simulation Inputs and Outputs



Save your simulation work including all INP and OUT files, this program compresses and compile these files into a zip file.

Import a compressed zip file containing various simulation INP and OUT files.

Use this option to export FUEL_INVENTORY.OUT into ELM archive. Archived fuel inventories can be used for future calculations and reference.

Use this option to export FUEL_INVENTORY.INP into ELM archive. Archived fuel inventories can be used for future calculations and reference.

Use this option to load a fuel inventory from ELM archive.

Fig. 3.6-1: The *Simulation Data* tab.

3.6.1 Saving and Loading Simulations

For some reasons, the user is encouraged to save all simulation inputs (i.e. *main.inp* and *fuel_inventory.inp*) and outputs (all files with the *.out* extension) for future reference or future re-calculation. Therefore, TRIMON allows the user to do so by compiling all files into a compressed ZIP file. To save all simulation files, select *Simulation Data* tab on *TRIMON Dashboard* window (see Fig. 3.6-1) → click the [***Export Simulation Files***] button → a save file dialog will appear → provide a name for the *.zip* file → click the [***Save***] button.

To import a saved simulation ZIP file and load all saved information into TRIMON, click the [***Deploy Simulation Files***] button → an open file dialog will appear → locate the desired *.zip* file → click the [***Open***] button.

3.6.2 Fuel Inventory Transfers

It is also customary to save the *fuel_inventory.out* file for future reference. For example, the user may want to simulate a new reactor core configuration using the previously utilized fuels listed in *fuel_inventory.out.* **Note**: *fuel_inventory.out* contains a list of fuels utilized during the previous core calculation, where each fuel burnup level is updated according to the operational time interval of the previous core configuration. Therefore, the user will have to save the *fuel_inventory.out* as an *.elm* file → define the new core configuration in TRIMON → import the saved *.elm* file as *fuel_inventory.inp*. Now, the content of the previously generated *fuel_inventory.out* has become the content of the current *fuel_inventory.inp*.

To save the current *fuel_inventory.out* as an *.elm* file, select *Simulation Data* tab on *TRIMON Dashboard* window (see Fig. 3.6-1) → click the [***Export fuel inventory output as***

.elm file] button → a save file dialog will appear → provide a name for the *.elm* file, say, *burned_previous_core_fuels.elm* → click the [***Save***] button.

To load an *.elm* file as the current *fuel_inventory.inp*, simply click the [***Load .elm file as input***] button → an open file dialog will appear → locate the desired *.elm* file → click the [***Open***] button.

For users' convenient, TRIMON also allows the user to save the current *fuel_inventory.inp* file for future reference. To do so, click the [***Export fuel inventory input as .elm file***] button → a save file dialog will appear → provide a name for the *.elm* file → click the [***Save***] button.

3.7 Monitoring Neutron Tracks

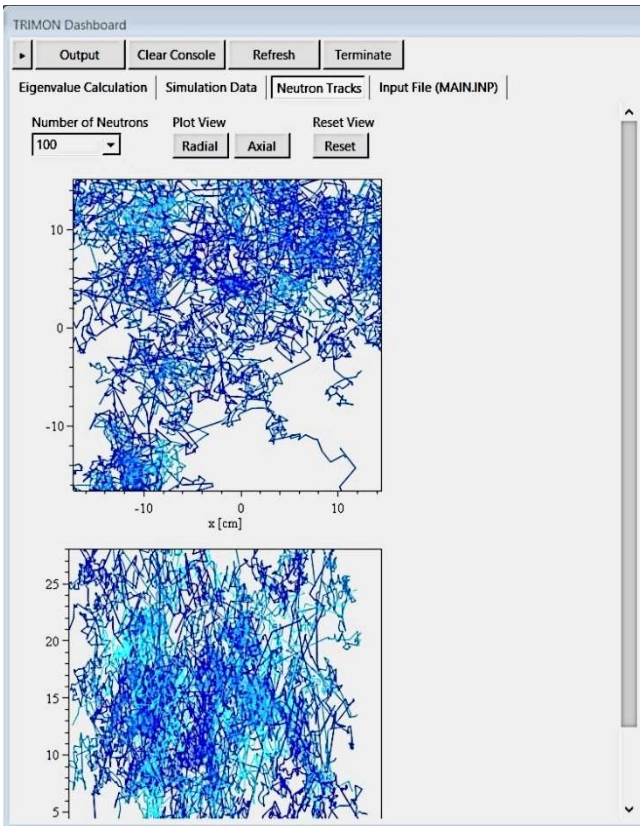


Fig. 3.7-1: The *Neutron Tracks* tab.

It is also important to avoid anomalies of the neutron tracks occur during the simulation to ensure results reliability and accuracy. Anomalies such as pre-mature track termination and tracks that extends past the outermost reactor boundary must be avoided. If any of these anomalies happens, the user is advised to double-check the simulation inputs. Therefore, TRIMON allows the user to view the first 1000 neutron tracks recorded during the simulation. Plus, the plot of these tracks is fascinating and enables users to understand the way neutrons get transported within a nuclear reactor.

To monitor the plot of neutron tracks, select *Neutron Tracks* tab on *TRIMON Dashboard* window (see Fig. 3.7-1). By default the radial and axial plot of the first 50 neutron tracks will appear on the plot view. The user can also limit the number of neutron tracks in both radial and axial plots. To do so, select the number of neutrons from the *Number of Neutrons* drop down list → click the [***Radial***]/[***Axial***] button to plot the radial/axial neutron tracks plot, respectively. To reset both radial and axial plots of the neutron tracks, click the [***Reset***] button.