

UNIVERSITY OF BIRMINGHAM

University of Birmingham

Department of Physics

Finding a Good Reflector Material for the System-integrated Modular Advanced ReacTor (SMART)

Author:
Fanta Diaby

Supervisor:
Dr. Paul Norman

A thesis submitted for the degree of

MSc Physics and Technology of Nuclear Reactors

March 13, 2020

UNIVERSITY OF
BIRMINGHAM

University of Birmingham Research Archive

e-theses repository

This unpublished thesis/dissertation is copyright of the author and/or third parties. The intellectual property rights of the author or third parties in respect of this work are as defined by The Copyright Designs and Patents Act 1988 or as modified by any successor legislation.

Any use made of information contained in this thesis/dissertation must be in accordance with that legislation and must be properly acknowledged. Further distribution or reproduction in any format is prohibited without the permission of the copyright holder.

Abstract

The behavior of five different reflector materials using the System-Integrated Modular Advanced ReacTor (SMART) was investigated to see which material works best as a reflector. The reflector materials used were Copper, Stainless Steel (301), Tungsten, Beryllium Copper, and Tungsten Carbide. A four-assembly model of the SMART was modeled using OpenMC. Different configurations were analyzed.

Acknowledgements

I would like to thank Dr. Paul Norman (Professor) and Yiming Zhong (PhD candidate) for guiding me throughout the project.

Contents

1 Introduction **6**
- 1.1 Benefits of Small Modular Reactors . . . 6
  - 1.1.1 Safety . . . 7
  - 1.1.2 Economic Aspect . . . 7
  - 1.1.3 Non-Proliferation . . . 7
- 1.2 System-Integrated Advanced Modular Reactor (SMART) . . . 8
- 1.3 Objectives . . . 8
- 1.4 Contributions . . . 8

2 Theory **9**
- 2.1 Neutron Transport . . . 9
  - 2.1.1 Neutron Balance in a PWR . . . 9
  - 2.1.2 Neutron Transport Equation . . . 9
  - 2.1.3 Solving the Neutron Transport Equation . . . 10
- 2.2 Neutrons Energy in a Reactor . . . 12
  - 2.2.1 Flux Distribution at the Reactor Core . . . 13
  - 2.2.2 Flux Distribution in the Reflector . . . 14
- 2.3 Reflectors . . . 14
- 2.4 SMART Reactor Vessel Assembly . . . 15
- 2.5 SMART Core . . . 16

3 Methodology **19**
- 3.1 Core Modeling . . . 19
- 3.2 Material Geometry . . . 22
  - 3.2.1 Reflector Materials . . . 22

4 Analysis and Results **23**
- 4.1 Neutron Flux and Fission Rate . . . 23
- 4.2 Distribution Uniformity . . . 26
- 4.3 Absorption and Scattering . . . 29
  - 4.3.1 A Different Configuration . . . 32
- 4.4 Result Summary . . . 34

5 Conclusion and Discussion **35**

A OpenMC Code **36**

Bibliography **48**

List of Figures

2.1 Neutron Energy Distribution at $T = 300°K$ 13
2.2 SMART Reactor Vessel Assembly 16
2.3 OpenMC Simulation of the SMART core 17
2.4 SMART Fuel Assembly Layout 17
2.5 Fuel assembly designs (a)Fuel Assembly type A (b) Fuel Assembly type B (c) Fuel Assembly type C 18

3.1 OpenMC Model of the SMART Core. 20
3.2 Four Assembly Model of the SMART Assembly 21
3.3 Representation of One Fuel Assembly 21

4.1 Thermal Neutron Flux Distribution 23
4.2 Fast Neutron Flux Distribution 24
4.3 Fission Rate Distribution 25
4.4 Assembly Slices 25
4.5 The Four Model SMART Assembly 26
4.6 Fission Rate Distribution using Different Reflectors. 27
4.7 Thermal Neutron Flux using Different Reflectors. 28
4.8 New Reflector Arrangement 30
4.9 A Different Reflector Configuration 32
4.10 Modified Assembly 33

List of Tables

3.1 Reflector Types 22

4.1 The Effective Neutron Multiplication Factor K_{eff} 23
4.2 Neutron Flux and Fission Rate 25
4.3 Fission Rate Uniformity Ratio 27
4.4 Thermal Flux Uniformity Ratio 28
4.5 Mean Thermal Absorption 29
4.6 Mean Fast Neutron Absorption 29
4.7 Mean Thermal Neutron Scatter 29
4.8 Mean Fast Neutron Scatter 29
4.9 Thermal Ratio (Scattering/Absorption) 30
4.10 Fast Ratio (Scattering/Absorption) 30
4.11 Uniformity Distribution for New Reflector Arrangement 31
4.12 Neutron Flux and Fission Rate for New Reflector Arrangement 31
4.13 K_{eff} for New Reflector Arrangement 31
4.14 Modified Assembly Results 33
4.15 Uniformity Distribution for the Modified Fuel Assembly 33

Chapter 1

Introduction

Although nuclear energy is often compared to renewable energies in the electricity generation market, it is also their best ally in the race to decarbonize our energy systems. The progressive abandonment of fossil fuels (coal and oil) in favor of "clean" energies requires us to turn towards future energies capable of ensuring efficient and durable electricity generation. The construction of new power stations, both expensive and technologically complex, is rugged and even inaccessible for some developing countries with growing needs. To address this gap between supply and demand, a new category of reactors that is smaller, more flexible and better suited to targeted needs could emerge in the next decade. The idea is to go from huge reactors (current conventional nuclear power plants) to smaller reactors, which would be manufactured in factories.

Small Modular Reactors (SMRs) are a category of nuclear reactors, smaller than conventional reactors, manufactured in a factory and transported to a chosen location for installation. Modular reactors reduce on-site work and increase nuclear materials' containment efficiency and safety. SMRs are a lower-cost alternative to conventional nuclear reactors. They are mainly intended for the power supply of isolated sites. They can be combined with heat and power generation for the supply of district heating networks, seawater desalination or heat for industrial processes.

Several SMR concepts are currently being developed, ranging from smaller versions of existing nuclear reactor designs to innovative full-generation IV concepts, both thermal neutron reactor and fast neutron reactor types. [1]

1.1 Benefits of Small Modular Reactors

The problems encountered by the current reactor led to the start of the SMR project. SMRs are much smaller than existing reactors: 50 to 100 MW in most cases and a maximum of 300 MW, compared to 1,000 to 1,700 MW (conventional nuclear reactors). They aim to reduce financing difficulties, the main obstacles many countries encounter. They are a nuclear solution to isolated sites, usually needing more skilled labor and high delivery costs. A gradual adaptation of the reactor's power to the changing needs is possible by adding additional modules. Conversely, if electricity demand is low, it can produce less electricity. Modular construction offers potential benefits in transportation, installation and decommissioning. SMRs have easier

containment due to integrating their components in a small and hermetically sealed volume to reduce the proliferation risks and allow reduced management to the strict minimum. They are fabricated and can be assembled in a dedicated factory before being sent to the site, where they can be installed more efficiently than conventional reactors. Some SMRs can use innovative fuels that allow higher burnup rates and longer life cycles (noticeable in isolated, often inaccessible locations). Extending refuelling intervals reduces the risk of proliferation and the likelihood of radiation escaping containment. Isolated sites often need help finding reliable and economically efficient sources of energy. SMRs can provide solutions for these hard-to-reach places. [2]

1.1.1 Safety

Due to the lack of qualified personnel available in remote areas, SMRs must be intrinsically safe. While large power plants use "active" safety systems [3] that require the intervention of intelligence or human controls, SMRs are designed to use "passive" or inherent safety systems, which are to operate without any outside intervention: a pressure relief valve can automatically release the pressure when it becomes too strong. Passive safety systems do not require motorized moving parts to function; they only depend on the laws of physics. Since there are several different concepts of SMRs, therefore several different safety systems can also be implemented. A negative vacuum coefficient in moderators and fuels controls fission reactions by slowing them down as the temperature increases. Some SMR concepts use underground placement of reactors and spent fuel storage pools for increased safety. Smaller reactors would be easier to modernize quickly, require less permanent workforce and have better quality control. [4]

1.1.2 Economic Aspect

Their modularity allows for short construction time and cost savings. A vital factor of the SMR is the economy of scale, compared to the large reactors, which arises from the possibility of prefabricating them in a manufacturing facility. Another economic advantage of SMRs is that the initial cost of constructing SMRs is much lower than that of building a large, much more complex and non-modular reactor. This makes SMRs a lower-risk investment compared to conventional reactors. [5]

1.1.3 Non-Proliferation

SMR sites are expected to have a much smaller staff than existing nuclear power plants. Many SMRs are designed to reduce the danger of theft or loss of materials. The nuclear fuel is low-enriched uranium, with a concentration of less than 20% of 235U fissile isotope. This small amount of non-military uranium makes the fuel less desirable for weapons production. After the fuel has been irradiated, the fission products mixed with the fissile materials are highly radioactive and require special treatment to extract them safely; this is another non-proliferating feature [6]. The modular construction of SMRs is another helpful feature. Since the reactor core is

often built inside a central manufacturing plant, only some people have access to fuel before and after irradiation.

1.2 System-Integrated Advanced Modular Reactor (SMART)

The System-Integrated Advanced Modular Reactor (SMART) is a pressurized water SMR with a built-in steam generator and advanced safety features [7]. The SMART project was developed by the Korea Atomic Energy Research Institute (KAERI) since 2012. It was approved by the South Korean surveillance authority. If the prospects for domestic use appear reduced, mainly due to the anti-nuclear policy of the government in place, the South Korean company is targeting a possible export market. It can also produce industrial heat, for example, to desalt seawater. In 2011, South Korea and Saudi Arabia signed an agreement on cooperation in the peaceful use of nuclear power. This agreement covers research and development, the construction and operation of a nuclear power plant, and nuclear safety and security policy issues. It also provides a legal framework for scientific, technical and economic collaboration. Saudi Arabia plans to build 16 nuclear units in the next 20 years.

1.3 Objectives

This research aims to study the behavior of different reflector materials and how they could affect the functionality of the SMART reactor. The SMART core will be modeled, and other reflector materials will be inserted around the core. The performance of the different reflector materials will be analyzed. The reflectors' arrangement will be modified and analyzed.

1.4 Contributions

Open MC was used to model the core. OpenMC, created by the Massachusetts Institute of Technology, is an open-source Monte Carlo particle transport code used for reactor physics. It enables to model the reactor core by simplifying its simulation process and output data analysis. Python API generates the inputs for OpenMC by quickly automating the generation process. It provides many software libraries for data analysis and post-processing tools. This research was done with the help of Dr Paul Norman (Professor) and Yiming Zhong (Ph.D. student), who provided the Open MC code for the SMART system.

Chapter 2

Theory

This chapter will explore the mechanism involved in neutron transport, a physics background, and tools used to find results.

2.1 Neutron Transport

Neutron transport is the study of the path of neutrons in matter and the reactions they induce, particularly power generation by the fission of nuclei of heavy atoms. Among other things, neutrons can be used to study neutron fluxes (number of neutrons per unit area and per unit of time: n / cm2 / s), the reactivity of the medium (this parameter allows us to report on the self-maintenance of nuclear reactions) and reaction rates (fission, absorption, diffusion) in the media. Neutron studies are the basis for operating nuclear reactors with control fission, such as pressurized water reactors (PWR) used by EDF or fast neutron reactors to produce energy and deliver it as electricity.

2.1.1 Neutron Balance in a PWR

It is assumed that the only fissile material is 235U. On average, 235 uranium 235 fission releases 242 neutrons, which give rise to the following reactions: 100 neutrons cause 100 new fissions, thus sustaining the chain reaction and consuming 100 nuclei of the fissile material. Seventy neutrons undergo fertile captures by 70 nuclei of the abundant U-235 material, transforming them into as many fissile nuclei of Pu-239. Seventy-five neutrons undergo sterile captures by fissile nuclei (30 neutrons) or by refrigerant nuclei, core structures, control elements or fission products. Five neutrons flee from the core (to be captured by neutron shields).

2.1.2 Neutron Transport Equation

Knowledge of the neutron distribution of a specific energy, or velocity, in space and time, makes it possible to operate and control a nuclear reactor or know the radiation dose around a nuclear site storage of used nuclear fuel. The transport equation describes neutron density as a function of the position and neutron velocity

at a fixed time. This equation is written generically,

$$\frac{1}{v}\frac{\partial\Phi}{\partial t}+\underline{\Omega}\cdot\nabla\Phi+\Sigma_t\Phi=\int_{4\pi}\int_0^{\infty}\Sigma_S\left(E'\rightarrow E,\Omega'\rightarrow\underline{\Omega}\right)\Phi\left(\underline{r},E',\Omega',t\right)dE'd\Omega'+s(\underline{r},E,\underline{\Omega},t) \tag{2.1}$$

where v is the neutron velocity, $\Phi(r,E',ro',t)$ is the desired flux of neutrons of velocity v and direction Ω is at the position r. The transport vector, Σ_t and Σ_s are the total interaction and scattering macroscopic cross sections for neutrons in the medium, respectively, r is the displacement vector, t is time, and s is the source term. The terms on the left represent group neutron losses, either by leaking out of range or by absorption or change of direction of propagation. The terms on the right represent the production of neutrons by radioactive decay or diffusion of neutrons in a particular domain.

2.1.3 Solving the Neutron Transport Equation

Different resolution methods have been developed over the past fifty years depending on the type of applications considered. There are two main types of methods: probabilistic methods and deterministic methods. The MCNP software, developed in Los Alamos, is the best-known and most used in the world for probabilistic methods. This is the method used in this paper. It is based on a statistical method of counting possible events at each neutron. This way of solving allows us to take into account very complex geometries. Deterministic methods, on the other hand, are based on the numerical resolution of Equation 2.1 described above. Different approximations are used to simplify the equation: the stationary hypothesis, the uni-dimensional hypothesis or the integral form. The deterministic approach gives a priori in all points of the phase space the (mean) value of the physical quantity of interest. In practice, this cannot be the case for a simple statistical reason: the probability that a simulated particle gives rise to an event at a given point in space is almost null.

Monte Carlo Probabilistic Method

The principle of simulation, in the ordinary sense of the term, is to use a model, that is, an abstract representation of a system or a problem, and to study the evolution of this model without operating the actual system. This is, for example, the case when you go to the bank for a loan: you are told how much you would pay each month if you accept the offer without necessarily having to do so. To model is to pose a problem on paper or computer to study it. This is also done for weather forecasts, studying a physical phenomenon, constructing and developing new aircraft, etc. In the case of a bank loan, these are relatively basic calculations, but other examples require using particular methods.

Monte Carlo simulation is a method of estimating a numerical quantity that uses random numbers. Stanisław Ulam and John von Neumann named it, referring to gambling in casinos, during the Manhattan Project that produced the first atomic

bomb during the Second World War [8]. For example, the simulation of a bank loan is not a Monte Carlo simulation because it is an exact calculation based on the number of monthly payments and the interest rate; no random phenomenon intervenes in the calculations. Applied to the solution of the transport equation, Monte Calo for Neutron Transport (MCNP) explicitly simulates the collection of the equivalent physical systems considered above within the framework of the ergodic hypothesis and then operates at the end of the simulation the statistical average of all the results produced by each simulation, as a set of measures that would be made and "averaged" in a purely experimental setting. This average is a statistical estimate of the desired result. The microscopic level is preserved in that the characteristics (positions, energies, events) of the simulated history of each of the particles of the systems can be stored, analyzed, and reprocessed.

The Monte Carlo method involves the drawing of random variables within probability densities f, defined as:

$$dP = f(u)du \tag{2.2}$$

The probability P(a $<$ X $\leq$ b) is therefore,

$$P(a \leq X \leq b) = \int_a^v f(u)du \tag{2.3}$$

The probability that the event will occur before the variable reaches a limit value x is given by the distribution function F:

$$F(x) = P(X \leq x) = \int_{-\infty}^{x} f(u)du \tag{2.4}$$

The normalized probability is:

$$\lim_{x \to +\infty} F(x) = \int_{-\infty}^{+\infty} f(u)du = 1 \tag{2.5}$$

If we take a random variable, it will be uniformly distributed in [a, b]. The probability density is

$$f(x) = \begin{cases} \frac{1}{b-a} & \text{if} \quad a \leq x \leq b \\ 0 & \text{else} \end{cases} \tag{2.6}$$

And the associated distribution function is:

$$F(x) = \begin{cases} 0 & \text{if} \quad x < a \\ \frac{x-a}{b-a} & \text{if} \quad a \leq x \leq b \\ 1 & \text{if} \quad x > b \end{cases} \tag{2.7}$$

In particular, if a = 0 and b = 1, the variable $x = \xi$ is uniformly distributed on [0, 1]. Let ξ a random variable defined over a domain D. A function $f(\xi)$ is also a random variable. The expectation value of $f(\xi)$ is defined in terms of $P(\xi)$:

$$E[f(\xi)] = \int_{\mathcal{D}} d\xi P(\xi) f(\xi) \tag{2.8}$$

For $P(\xi)$ of the probability of the random variable to have its value between ξ and $f(u)$ of (2) The probability that the event will occur before the variable reaches a $\xi + d\xi$.

The variance of $f(\xi$ is given by:

$$\mathrm{Var}[f(\xi)] = \int_{\mathcal{D}} d\xi P(\xi)\{f(\xi) - E[f(\xi)]\}^2 \tag{2.9}$$

The standard deviation of $f(\xi)$ is the root of the variance:

$$\sigma[f(\xi)] = \sqrt{\mathrm{Var}[f(\xi)]} \tag{2.10}$$

2.2 Neutrons Energy in a Reactor

Neutrons, which arise during fission, are highly energetic and have an average energy of around 1MeV. As a consequence of the diffusion, the original high-energy neutron reacts with the nucleus of the material, and the neutron's energy decreases, resulting in a slow neutron. This process is called moderation. The energy of the neutron becomes so low that the energy of the neutron is in the same order as the kinetic energy of the atoms.

When the neutrons in a material collect, during collisions, as much energy as they give up, we then say that they are in thermal equilibrium with the material in which they are found. The kinetic energy depends on the temperature; we also speak of thermal neutrons. The energy distribution of thermal neutrons will be described by the Maxwell-Boltzmann equation:

$$\frac{n(E)}{n} = \frac{2}{\sqrt{\pi}kT} \cdot \sqrt{\frac{E}{kT}} \cdot e^{-\frac{E}{kT}} \tag{2.11}$$

With:

- n the neutron density in $neutron/cm^3$
- T the temperature in $^\circ K$
- k the Boltzmann constant = $13.810^{24} J/^\circ K$

This Maxwell-Boltzmann distribution is proposed graphically in Figure 2.1.

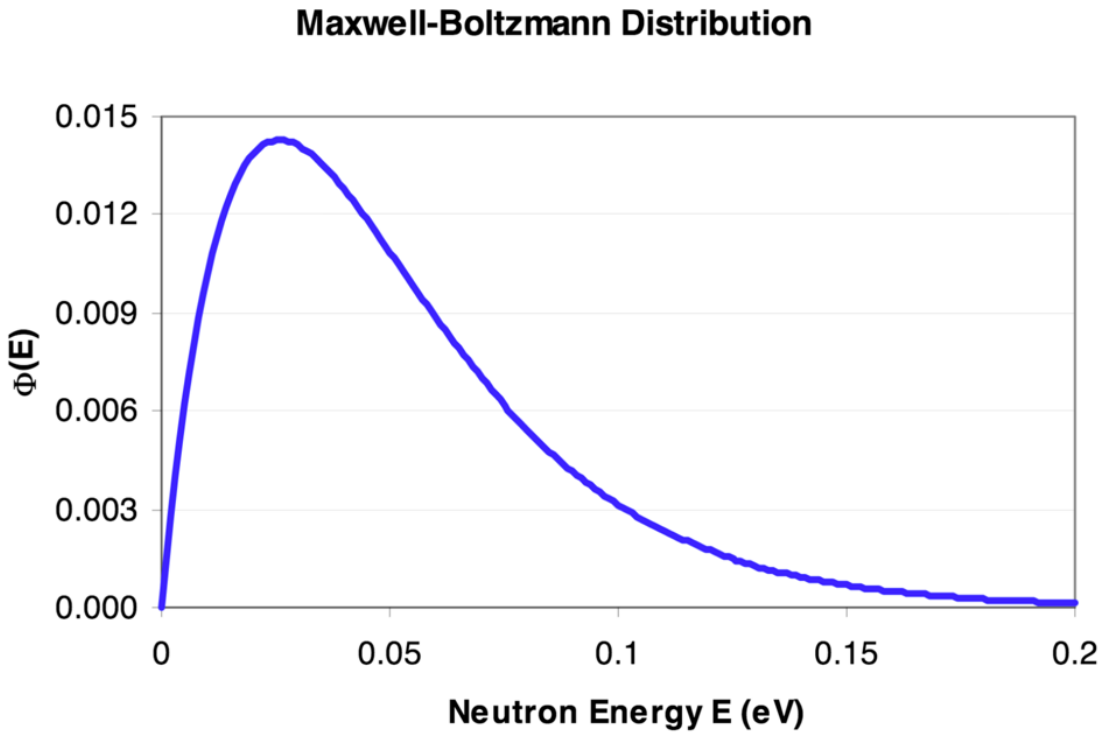


Figure 2.1: Neutron Energy Distribution at $T = 300°K$

The most probable energy is given by $E_{th} = kT$. The average energy is $1.5kT$. The most probable neutron energy can be determined by:

$$\mathrm{v_{th}} = \sqrt{\frac{2 \cdot \mathrm{E_{th}}}{\mathrm{m}}} = 2200 \frac{\mathrm{m}}{\mathrm{s}} \tag{2.12}$$

with $E_{th} = 0.025eV$.

In reactor physics, a distinction between the different neutrons is made according to their energy:

- Fast neutrons with an energy $E > 1keV$
- Epithermal neutrons with an energy between $0.4eV < E < 1keV$
- Thermal neutrons with an energy $E < 0.4eV$

2.2.1 Flux Distribution at the Reactor Core

Starting from n_i fast neutrons the number of new fast neutrons after a reaction cycle is given by:

$$n_{i+1} = \varepsilon \cdot \mathrm{p} \cdot \mathrm{f} \cdot \eta \cdot \mathrm{n_i} \tag{2.13}$$

The product of these factors is equivalent to the multiplication factor:

$$\mathrm{k_\infty} = \varepsilon \cdot \mathrm{p} \cdot \mathrm{f} \cdot \eta \tag{2.14}$$

In a reactor the production of neutrons is not zero. According to equation 2.13 this can also be written as:

$$\begin{array}{l} \frac{\mathrm{d}^2\Phi}{\mathrm{dx}^2} + \mathrm{B}^2 \cdot \Phi = 0 \\ \text{where} \\ \frac{\mathrm{k}_\infty - 1}{\mathrm{L}^2} = \mathrm{B}^2 \end{array} \tag{2.15}$$

The coefficient B is determined by the shape of the reactor and is called the "geometric Laplacian" (curvature factor). Equation 2.15 gives as a general solution:

$$\Phi = \mathrm{A}_1 \cdot \cos(\mathrm{B} \cdot \mathrm{x}) + \mathrm{A}_2 \cdot \sin(\mathrm{B} \cdot \mathrm{x}) \tag{2.16}$$

The flux distribution in a reactor is symmetrical with respect to the center. If we take the center as x = 0, the constant A_2 must be equal to 0. So that equation 2.16 is reduced to

$$\Phi = \mathrm{A} \cdot \cos(\mathrm{B} \cdot \mathrm{x}) \tag{2.17}$$

Where A is again a constant which is determined by the power of the reactor.

2.2.2 Flux Distribution in the Reflector

In the reflector, where there is no fuel present and therefore no production of neutrons, the production factor is equal to 0 ($K_\infty = 0$). The diffusion equation therefore becomes:

$$\frac{\mathrm{d}^2\Phi}{\mathrm{dx}^2} - \frac{1}{\mathrm{L}^2} \cdot \Phi = 0 \tag{2.18}$$

A general solution is given by equation 2.18:

$$\Phi = \mathrm{A}_1 \cdot \mathrm{e}^{\frac{\mathrm{x}}{\mathrm{L}}} + \mathrm{A}_2 \cdot \mathrm{e}^{-\frac{\mathrm{x}}{\mathrm{L}}} \tag{2.19}$$

Since at a large distance x the density of the flux must be equal to 0, A_1 must also be equal to 0 and equation 2.19 is reduced to:

$$\Phi = \mathrm{A} \cdot \mathrm{e}^{-\frac{\mathrm{x}}{\mathrm{L}}} \tag{2.20}$$

Where A is a constant which will be determined by the power of the reactor.

2.3 Reflectors

A neutron reflector core is located around the reactor core. The reflector increases the uniformity of the power distribution in the fuel assemblies. Unlike the reactor core, the neutron reflector is a non-multiplying medium. The neutron reflector reduces neutron leakage by reflecting left-over neutrons into the core. The reflector is then able to increase the reactor's criticality (K_{eff}). In SMRs, it could help increase the power output without altering the size of the reactor. It is installed in PWRs for the following purposes:

- To increase the uniformity of the power distribution so that the neutron flux distribution is "flattened, "i.e., an increase in the ratio of the average flux to the maximum flux.

- The fuel at the edges of the core can increase the total power production due to the reflectors.

- To reflect neutrons that have escaped back into the core. These neutrons can then undergo further chain reactions. This implies that the reactor size can be reduced, so reflectors are important in SMRs because of the reactor size.

A good reflector must possess the following properties:

- Stability in radiation

- Oxidation resistance

- Elastic scattering cross-sections for light and heavy elements.

- Low absorption

- High reflection

[9]

2.4 SMART Reactor Vessel Assembly

Figure 2.2 shows a diagram of the SMART reactor vessel assembly. One can observe the integrated arrangement of its components. Compared to traditional PWRs, SMART does not use large-size pipes as connections between the reactor's coolant systems, which prevents the risk of significant break loss of coolant accident (LBLOCA), which has been the cause of considerable accidents in the history of the nuclear industry. Its major primary components are located in the reactor's assembly, which include fuel and core, four reactor coolant pumps (RCP), eight steam generators (SG), 25 control rod drive mechanisms (CRDM), a pressurizer (PZR), all placed inside the pressurized reactor vessel (PRV) [10] The RCPs, which are placed horizontally in the upper part of the RPV, exert a force on the coolant, which then flows in an upward direction through the core. The coolant penetrates the SG from the top. The secondary feedwater enters the SG from the bottom and moves upward by removing heat. As a result, it leaves the SG as hot steam. The PZR region is located above the reactor water level in the upper part of the RPV. Due to the control of heat and a large volume of PZR steam, the system's primary pressure stays constant. The 8 SGs are responsible for coolant natural circulation. [10]

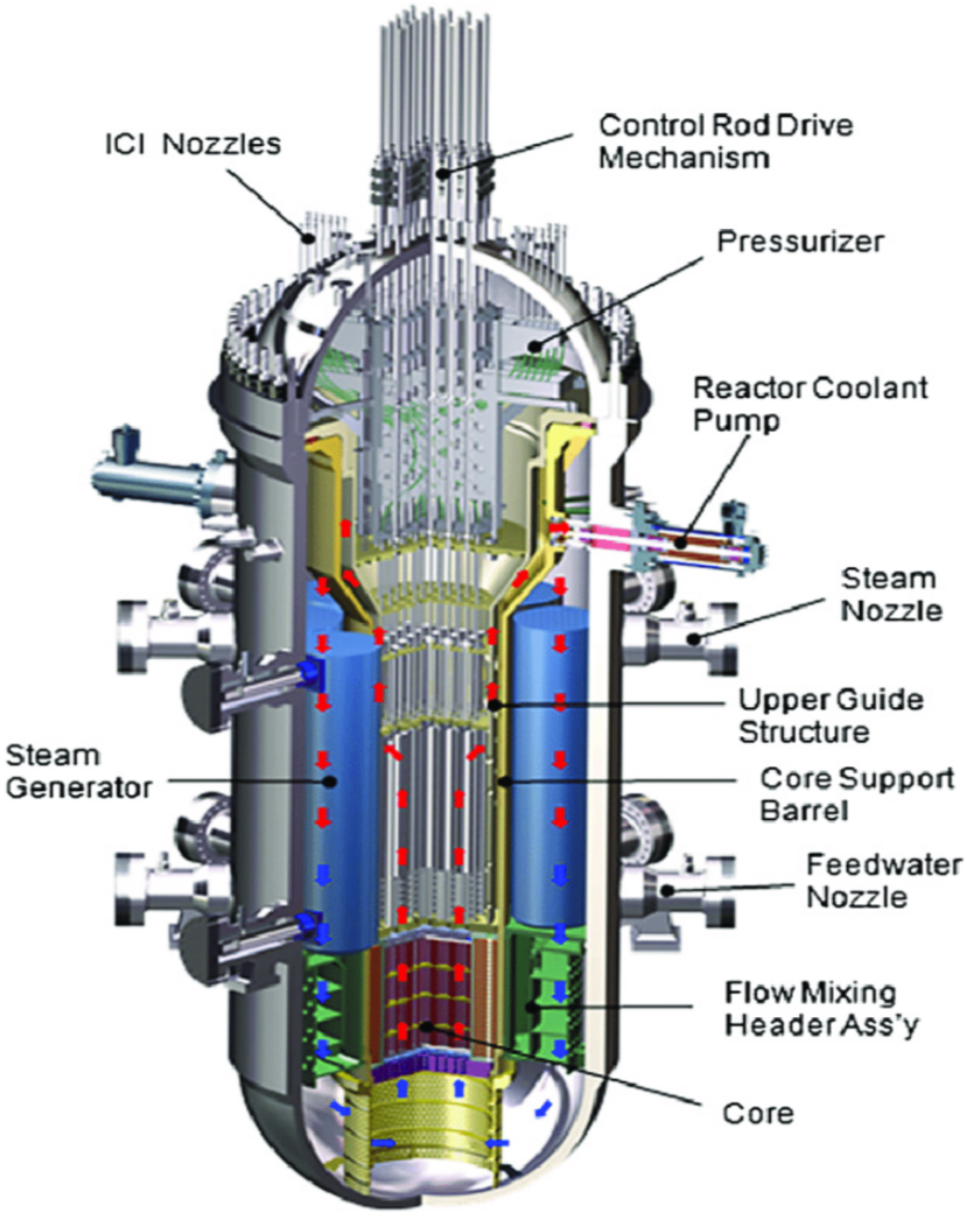


Figure 2.2: SMART Reactor Vessel Assembly

2.5 SMART Core

Figure 2.3 represents the full core SMART assembly layout, modelled by Yiming Zhong, using OpenMC. The fuel assembly lattices have a 17 by 17 dimension. There are 16,473 fuel pins and 57 fuel assemblies. The fuel rods are arranged in a rectangular lattice. The nuclear fuel is ceramic uranium dioxide (UO_2) with a 4.95% enrichment for all the fuel assemblies. The fuel assemblies contain a mixture of fuel rods and burnable poison rods or neutron absorbers. The control rod absorbers are Ag-In-Cd, and the burnable poison materials are Gd_2O3 and B_4C in AL_2O_3.

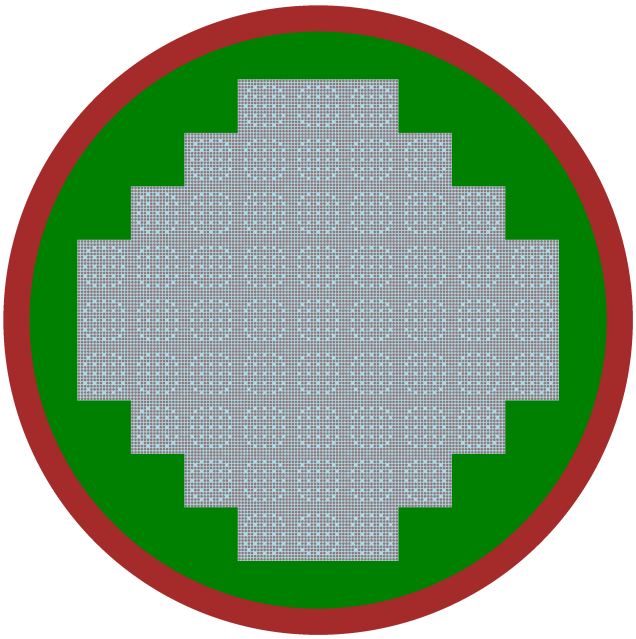


Figure 2.3: OpenMC Simulation of the SMART core

Figure 2.4 shows the SMART core fuel assembly pattern with the three fuel types mentioned above. More burnable poisons with a high concentration in ^{10}B are used in the core where the neutron fluxes are expected to be higher. These different concentrations, A, B, and C, are used in other core areas so that there is an even distribution throughout the core to investigate the arrangements in the fuel assemblies [11].

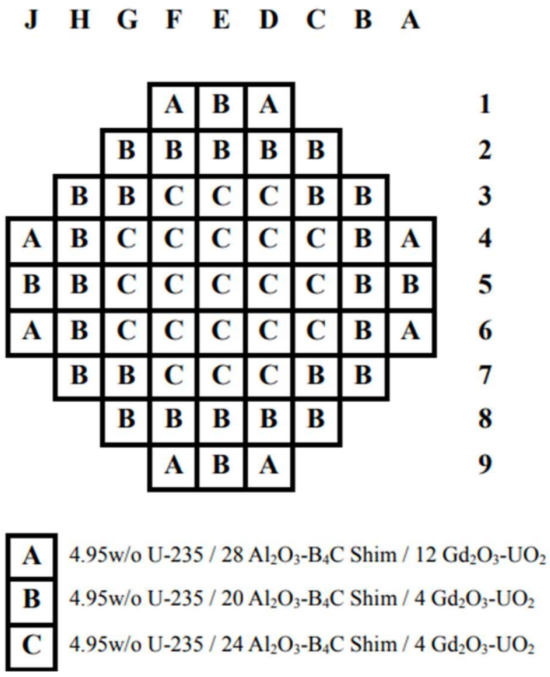


Figure 2.4: SMART Fuel Assembly Layout

For example, the three configurations below in Figure 2.5 were studied by Rahmi N. Ramdhani et al. in their "Neutronics Analysis of SMART Small Modular Reactor using SRAC 2006 Code" by comparing them in terms of reactivity control and uniform flux distribution. [12] [13]

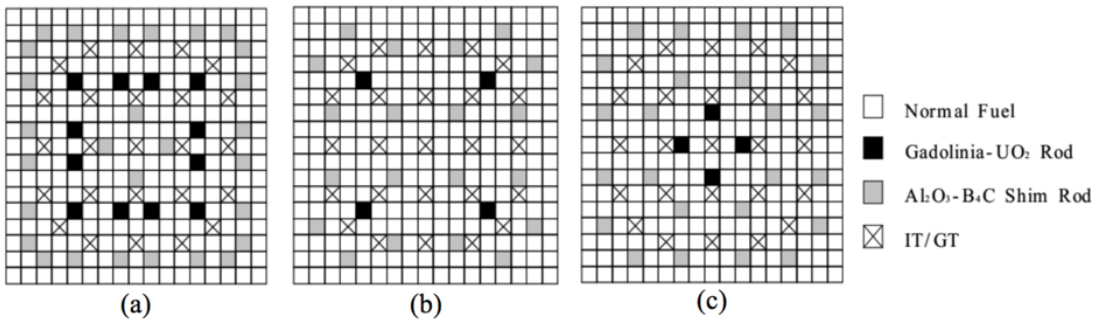


Figure 2.5: Fuel assembly designs (a)Fuel Assembly type A (b) Fuel Assembly type B (c) Fuel Assembly type C

.

Figure 2.5 shows three different arrangements of the fuel assembly by changing the placements of the burnable fuels GdO_2, B_4C, Al_2O_3, the control rods $Ag-In-Cd$ Alloy, and the fissionable nuclear fuel UO_2.

Chapter 3

Methodology

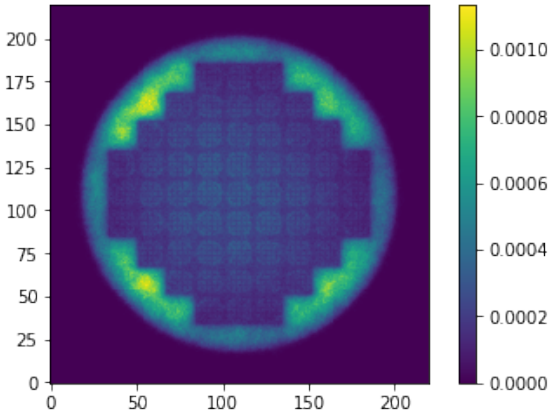
This chapter will discuss the process of modeling the SMART reactor and the methods to analyze the behavior of the different reflectors.

3.1 Core Modeling

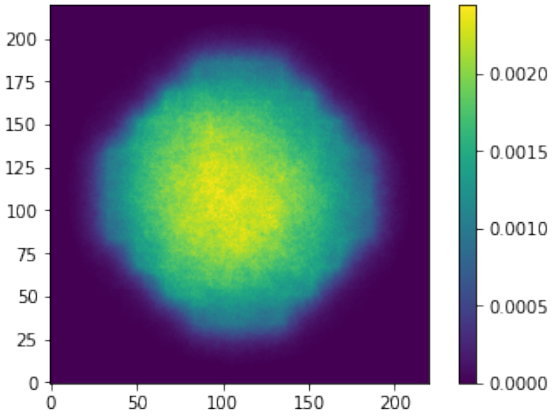
Figure 3.1 is the simulation of some results of the neutron behavior in the entire SMART core modelled by Yiming using OpenMC. A two-dimensional mesh plot represents these simulations. The first image shows the thermal flux distribution (Figure 3.1(a)), with an even flux distribution throughout the core.

The second (Figure 3.1(b)) and third (Figure 3.1(c)) images show the fast neutron flux and the fission rate distribution in the core, respectively. The center of the core has more activity than the peripherals.

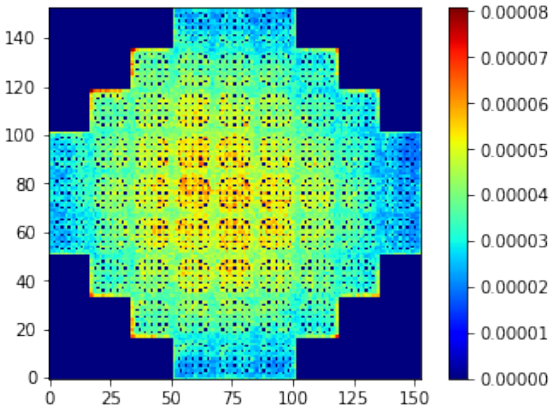
For simplification in this project, a reduced SMART core of four assemblies is modeled as shown in Figure 3.2. The "transmission" boundary type is used. Light water is used in reflector region 1 for all the simulations in this configuration. Reflector region 2, however, changes depending on the reflector used. The areas in the assemblies labelled H_2O represent the control rod and moderator regions. In this configuration, the control rods are not inserted, so the control rod region and the moderator are light water. The UO_2 fuel rods are surrounded by light water.



(a) Thermal Neutron Flux



(b) Fast Neutron Flux



(c) Fission Rate

Figure 3.1: OpenMC Model of the SMART Core.

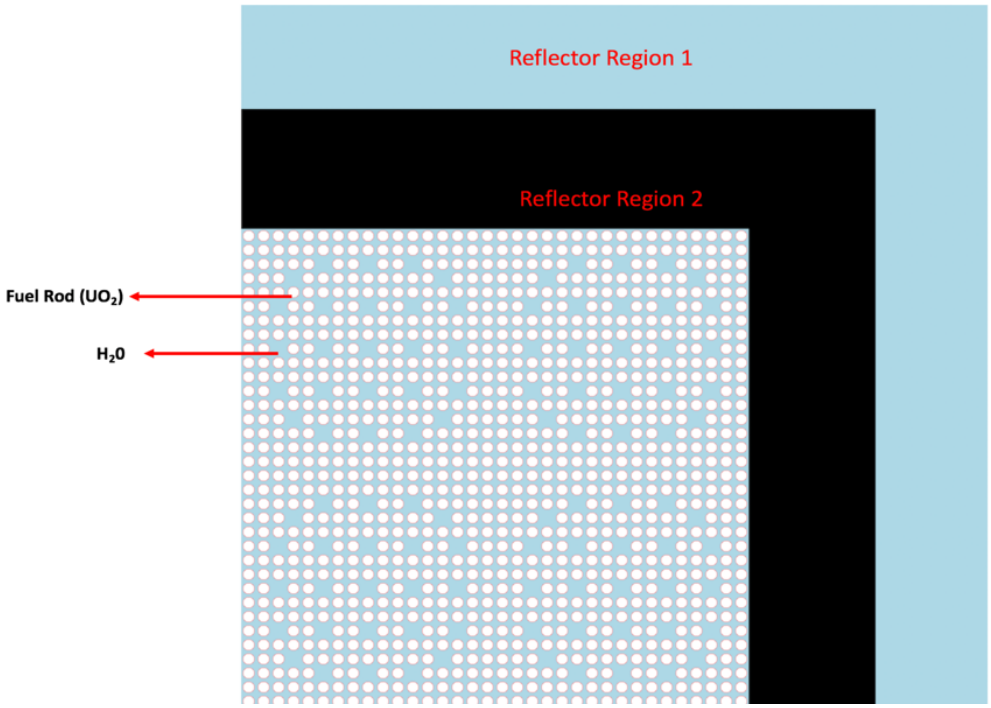


Figure 3.2: Four Assembly Model of the SMART Assembly

Figure 3.3 is a closeup image of a single fuel assembly of the model used.

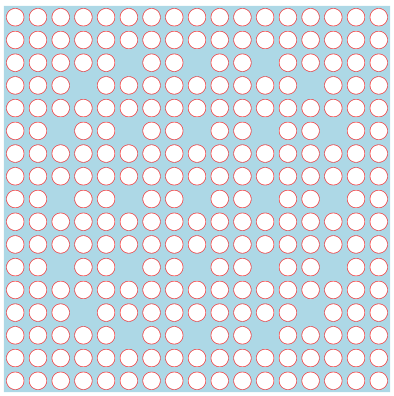


Figure 3.3: Representation of One Fuel Assembly

RecLattice is the geometry modelling method of OpenMC used to simulate the geometry of the fuel assembly.

As shown in Figure 3.1, a two-dimensional Cartesian mesh is used to decipher the fuel assembly. This aims to score different attributes of the nuclear reaction, such as the fission rate, the thermal flux, the absorption reaction rate, etc. In this model, the core lattice is a 2 x 2. The width of the mesh is [1.2, 1.2]. The diameter of the fuel assembly is 20.4cm. The radius of the fuel is 0.4465cm. The outer radius of the cladding is 0.475cm. The thickness of the zircaloy cladding is 0.0285cm. The

mesh plot has a dimension of 17 by 17. One lattice cell corresponds to a single mesh containing the nuclear fuel, cladding and moderator. For each reflector, 10000 histories were run.

3.2 Material Geometry

In the openMC input file, the material geometry was written to model the SMART core. The material geometry defines the geometric structure and the density of the materials. Each material was created using the Material *class* option in the OpenMC.

The UO_2 nuclear fuel used has a $10.97 g/cm^3$ density. A reduced nuclear fuel enrichment of 4.8% was set compared to 4.95% used in the entire SMART core. The enrichment is uniform for all the fuel rods in this case. The zirconium alloy or zircoloy clad tube, which contains the cylindrical shaped nuclear fuel, has a density of $6.56 g/cm^3$ and is composed of 97.8% zirconium (Zr), 1.2% tin (Sn) and 1% niobium (Nb). To even out the reactivity, light water (H_2O) with a density of $1 g/cm^3$ is used as the burnable poison mixed with the nuclear fuel. The control rods are not inserted, so there is light water in the regions where the control rods are supposed to be inserted.

3.2.1 Reflector Materials

The information about the reflector materials used is given in the table below Table 3.1.

Table 3.1: Reflector Types

Elements	**Density**	**Composition**
Copper	8.96	Cu
Stainless Steel 301	8.03	18%Cr, 2%Mn, 75%Fe, 1%Si, 8%Ni, 2%Mn, 0.045%P, 0.03%S, 0.15%C, 0.1%N
Tungsten	19.25	W
Beryllium Copper	8.25	3%Be, 97%Cu
Tungsten Carbide	15.63	0.939%W, 0.061%C

Five reflector materials with different densities are used. Two of the materials are simple elements such as Copper and Tungsten, and the rest are alloys.

Chapter 4

Analysis and Results

The reactivity coefficient K_{eff} using different reflector materials and its associated uncertainly are shown in Table 4.1.

Table 4.1: The Effective Neutron Multiplication Factor K_{eff}

Reflector Materials	K_{eff}	K_{eff} **Uncertainty**
Copper	1.14	+/- 0.0007
Stainless Steel	1.14	+/- 0.00113
Tungsten	1.17	+/- 0.0008
Beryllium Copper	1.15	+/- 0.00084
Tungsten Carbide	1.13	+/- 0.00077

In general, K_{eff} indicates the criticality level of the reactor. Since the SMART core is modified in this project, many things can affect the criticality. Therefore, getting the reactor to work at a critical level is not necessary in this case.

4.1 Neutron Flux and Fission Rate

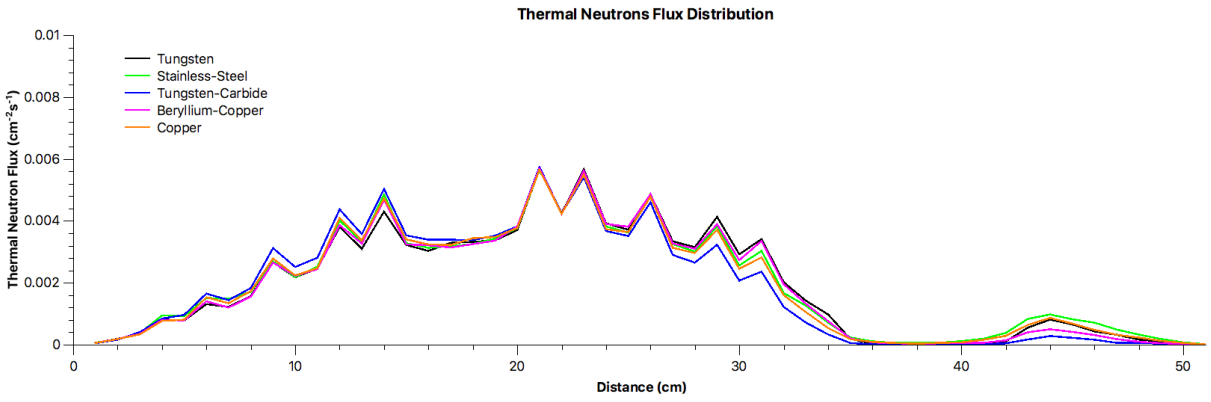


Figure 4.1: Thermal Neutron Flux Distribution

The graph in Figure 4.1 represents the thermal flux distribution for the different reflectors over the radial distance. There is a color assigned to each reflector. The thermal neutron flux distribution graph shows how the thermal neutrons are distributed in all directions per unit time when crossing through the cross-sectional area. This distribution is not smooth. The zig-zag-like shape along the curve shows

a fluctuation in flux. The thermal flux increases in moderator and control rod regions. In these regions, light water slows the neutrons to a thermal level. Light water has a high cross-section for neutron scattering with a relatively low absorption cross-section, and its atomic mass is close to the atomic mass of the neutron. Hence, there is a high thermal flux in this region.

There is a decrease in flux around the 35 cm radial distance. This is the position around which the metal reflector region starts (Reflector Region 2). In this region, the thermal neutron production is low because the neutrons are not slowed down to a thermal level. In the reflector 1 region containing light water, the neutrons are slowed down to a thermal range by interaction with water hydrogen atoms. Because the reflector region is a non-multiplying medium, the thermal flux would not be as high as in the reactor's core because there is no fission reaction, hence the absence of the zig-zag-like shape in this region. The results are recorded in Table 4.2.

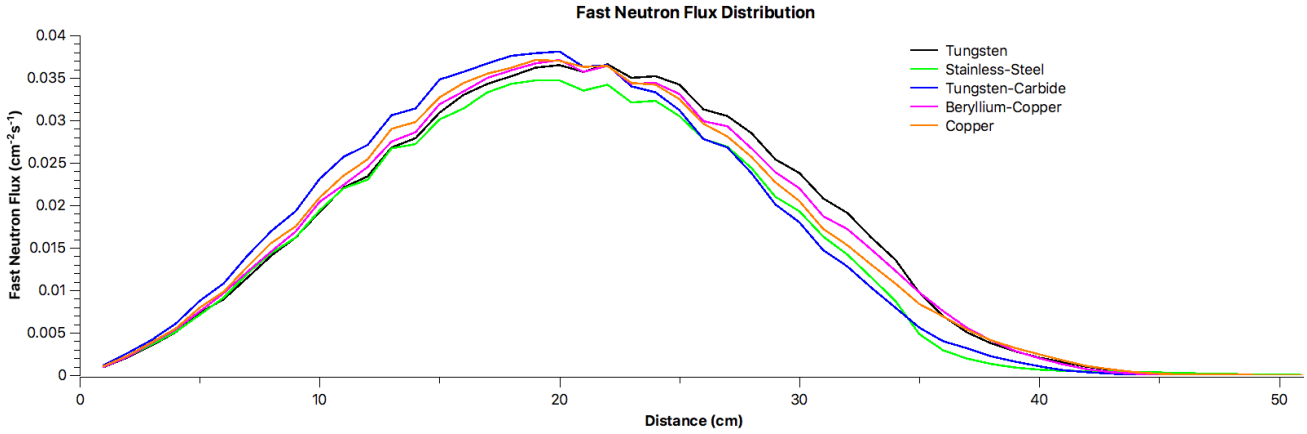


Figure 4.2: Fast Neutron Flux Distribution

Figure 4.2 shows the fast neutron flux distribution. The fast neutrons are the direct result of the nuclear fission reaction in the reactor. The energy of the neutrons is more significant than 0.4 eV, as mentioned in section 2.2. The fast neutron flux distribution is narrower than the thermal neutron distribution because it is less uniform. The flux is very high at the core's center compared to the core's peripheral regions. In the reflector region 1, the neutrons lose all their energy because of the moderator property of light water. However, there is a presence of fast neutron flux in reflector region 2 because of the nature of the reflector material used. Since the reflector region is not a multiplying medium, the fast neutron flux dies down because there is no fission reaction. The results are recorded in Table 4.2.

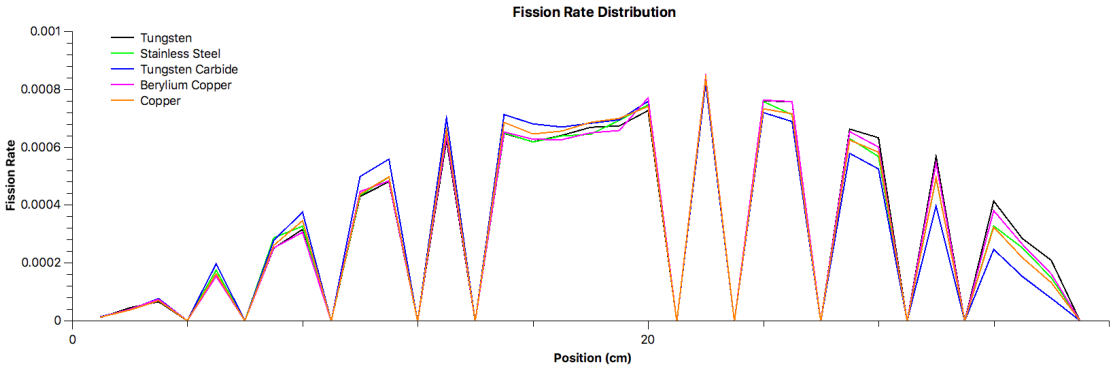


Figure 4.3: Fission Rate Distribution

Figure 4.3 displays the fission rate distribution. The fission rate is zero when there is no interaction with a nuclear fuel pin, meaning there is no fission reaction at these points, where there is H_2O. To show the reaction rate in the fuel pins only, these zero points are removed, and the shape of the graph is shown in Figure 4.4 below. The results are recorded in Table 4.2.

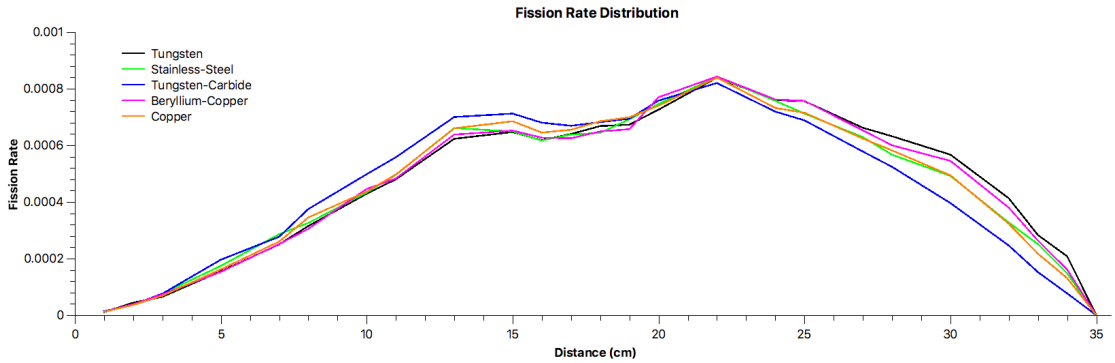


Figure 4.4: Assembly Slices

reflector regions. The effect of the reflectors in the reactor is shown. The increased reaction rates at a radial distance range of 20 to 22 cm are due to the impact of the reflectors scattering back the neutrons to the core. The center of the core stays the same, but the average reaction rate is increased due to the reflectors.

A summary of the thermal and fast neutron flux and the reaction rates using the five different reflectors is summarized in Table 4.2.

Table 4.2: Neutron Flux and Fission Rate

Reflector Materials	**Mean Thermal Flux**	**Mean Fast Flux**	**Mean Fission Rate**
Copper	0.001496	0.01137	0.000179
Stainless Steel	0.00157	0.0114	0.00017898
Tungsten	0.00144	0.0113	0.000182
Beryllium Copper	0.001401	0.0114	0.00018
Tungsten Carbide	0.00121	0.0107	0.000176

Tungsten has the highest mean fission rate.

4.2 Distribution Uniformity

To maintain a stable reaction cycle, having an even flux distribution and reaction rate throughout the core is essential. A non-uniform distribution leads to some fuel rods getting used up faster than others, leading to uneven reactivity in the reactor. To calculate uniformity, one would find the ratio of the average to the maximum value. Because of the configuration of the simulated core model in this project, the strategy is to find the ratio of the two opposing assemblies at the most extreme sides with the highest and lowest distribution. The labelled assemblies are shown in Figure 4.5.

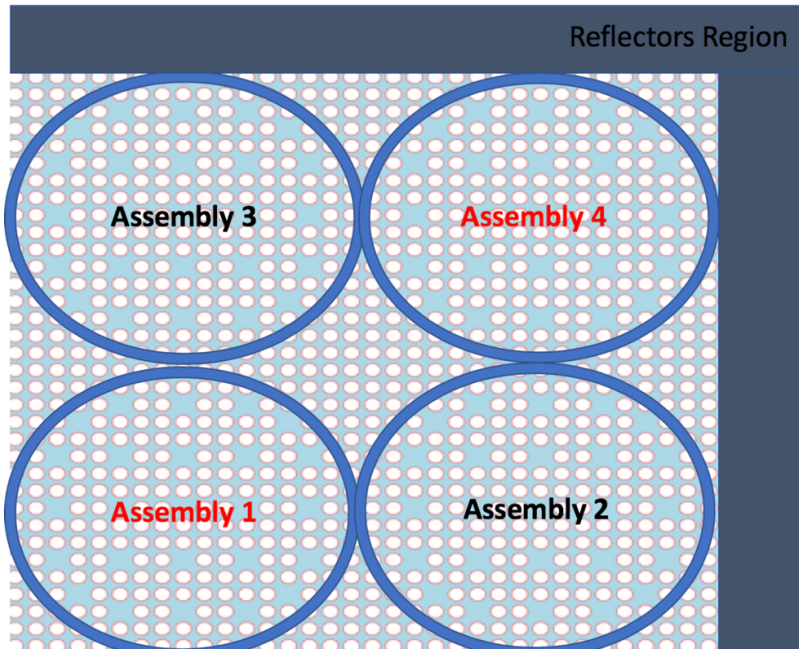
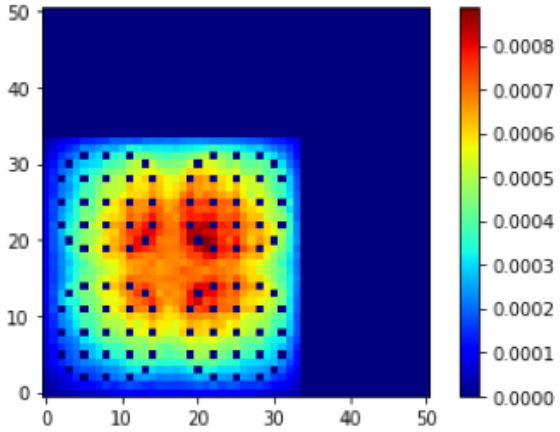


Figure 4.5: The Four Model SMART Assembly

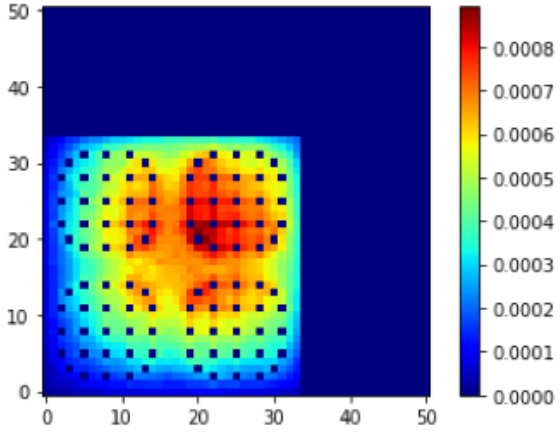
The assemblies were broken down into single assembly arrays in the OpenMC code, investigating them individually (See Code in Appendix). Figure 4.6(a) and 4.6(b) show the mesh plots for the fission rate using tungsten carbide and tungsten reflectors, respectively. There is a noticeable difference between assemblies 1 and 4 in both figures. Assembly 2 and 3, on the other hand, have similar distribution. This is consistent for all the reflectors used. The assemblies used are, therefore, 1 and 4.

The ratio of the mean thermal flux and fission rate of assembly 1 to assembly 4 was determined for each reflector material to determine the uniformity distribution. Let u be the ratio:

$$u_{\%} = [(Assembly1)_{average}/(Assembly4)_{average}] \times 100\% \tag{4.1}$$



(a) Tungsten Carbide Fission Rate



(b) Tungsten Fission Rate

Figure 4.6: Fission Rate Distribution using Different Reflectors.

Table 4.3 displays the uniformity of the fission rate distribution from highest to lowest.

Table 4.3: Fission Rate Uniformity Ratio

Tungsten Carbide	67.6%
Copper	58.4%
Stainless Steel	56.4%
Beryllium Copper	54.2%
Tungsten	50%

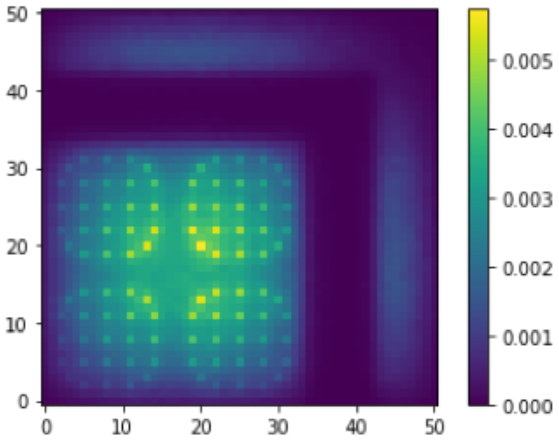
In Figure 4.6(a) and 4.6(b), The hot region is more centered in the tungsten carbide mesh plot, hence a higher uniformity distribution. Tungsten carbide has a uniformity distribution ratio of 67.6%, and tungsten has 50%. However, tungsten has a higher mean fission rate than tungsten carbide (See Table 4.2.) More neutrons are scattered back by the tungsten reflector, increasing the reaction rate but not the uniformity.

Having a uniform thermal neutron flux distribution ensures some equilibrium in the usage of the fuel assemblies. The results are recorded in Table 4.4 from the highest to the lowest uniformity distribution.

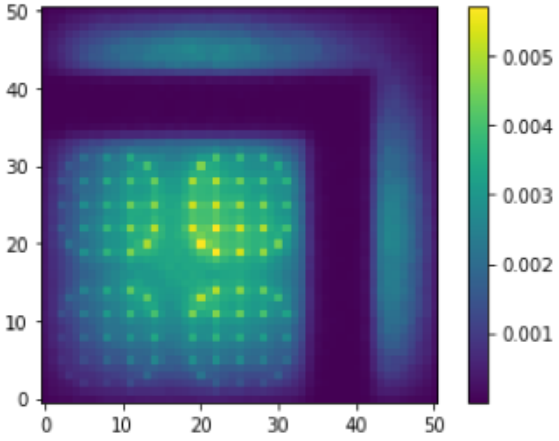
Table 4.4: Thermal Flux Uniformity Ratio

Tungsten Carbide	67.3%
Copper	58.0%
Stainless Steel	56.4 %
Beryllium Copper	53.5%
Tungsten	51.4%

Tungsten carbide has the highest distribution uniformity, followed by copper. Tungsten, on the other hand, has the lowest uniformity. The mesh plots in Figure 4.7(a) and 4.7(b) show the mean thermal flux distribution in Tungsten Carbide and Tungsten, respectively.



(a) Tungsten Mean Thermal Flux



(b) Tungsten Mean Thermal Flux

Figure 4.7: Thermal Neutron Flux using Different Reflectors.

When using the tungsten reflector, one can notice in its mesh plot that the flux activity seems higher, especially in assembly 4, compared to the tungsten carbide reflector.

4.3 Absorption and Scattering

A tally for the absorption reaction rate in the reflector was created for both thermal and fast neutrons. The mean thermal and fast neutron absorption rate for each reflector material is written in Table 4.5 and 4.6, respectively, from highest to smallest.

Table 4.5: Mean Thermal Absorption

Stainless Steel	0.00001155
Copper	0.0000111
Beryllium Copper	0.00001084
Tungsten	0.00001066
Tungsten Carbide	0.00000758

Table 4.6: Mean Fast Neutron Absorption

Tungsten Carbide	0.00002304
Beryllium Copper	0.00001245
Copper	0.00000888
Tungsten	0.00000877
Stainless Steel	0.000006103

Stainless Steel absorbs the most thermal neutrons and the least fast neutrons, while tungsten carbide absorbs the most fast neutrons and the least thermal neutrons. A low absorption rate indicates a more efficient reflector. Tungsten shows a consistent behavior for both thermal and fast neutron absorption, with the second lowest absorption rate for both absorption types.

A tally for the scattering reaction rate in the different reflector materials was created for both thermal and fast neutrons, and the results are recorded in Table 4.7 and 4.8, respectively, from highest to smallest.

Table 4.7: Mean Thermal Neutron Scatter

Tungsten	8.81E-05
Stainless Steel	7.03E-05
Beryllium Copper	5.02E-05
Copper	4.17E-05
Tungsten Carbide	7.92E-06

Table 4.8: Mean Fast Neutron Scatter

Tungsten	0.00169
Beryllium Copper	0.001296
Stainless Steel	0.0009378
Copper	0.0008186
Tungsten Carbide	0.0007903

High neutron scattering is one of the properties of a good reflector. Tungsten scatters more thermal and fast neutrons than the other reflectors, while tungsten carbide has a poor absorption rate.

The ratio of neutron scattering to neutron absorption is shown in Table 4.9 for thermal neutrons and Table 4.10 for fast neutrons.

Table 4.9: Thermal Ratio (Scattering/Absorption)

Tungsten	8.262664165
Stainless Steel	6.088311688
Beryllium Copper	4.63099631
Copper	3.756756757
Tungsten Carbide	1.045118734

Table 4.10: Fast Ratio (Scattering/Absorption)

Tungsten	192.7023945
Stainless Steel	153.6621334
Beryllium Copper	104.0963855
Copper	92.18468468
Tungsten Carbide	34.30121528

The tables above show that tungsten has the highest scattering-to-absorption ratio. This implies that tungsten has the highest efficiency. Tration

A different reflector configuration is modeled. The goal is to see how a reflector configuration different from the reflector configuration shown in Figure 3.2 could affect the neutron behavior in the reactor. Figure 4.8 below shows the new reflector arrangement that will be investigated.

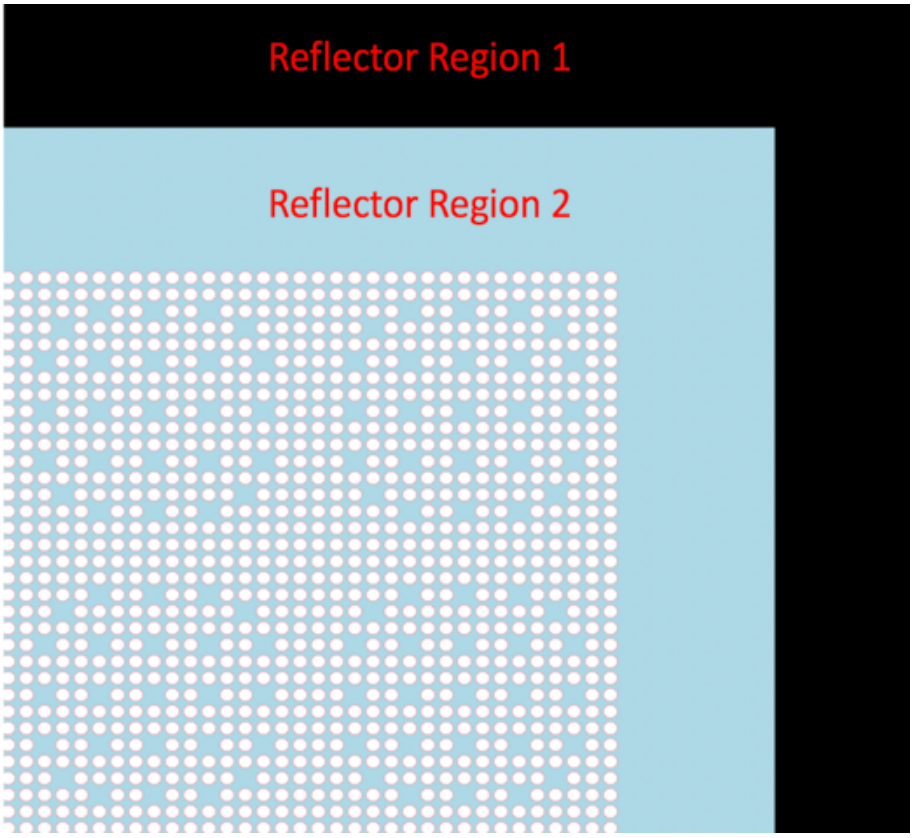


Figure 4.8: New Reflector Arrangement

Table 4.11 records the thermal flux and fission rate uniformity ratios. These results show that the overall uniformity distribution is decreased in this configuration.

Table 4.11: Uniformity Distribution for New Reflector Arrangement

Reflector Materials	**Thermal Flux Uniformity Ratio**	**Fission Rate Uniformity Ratio**
Copper	50.4%	50.1%
Stainless Steel	51.6%	50.5%
Tungsten	51.6%	50%
Beryllium Copper	53.5%	50.9%
Tungsten Carbide	51.4%	50.9%

Table 4.12 below summarizes the results of the different attributes obtained using this configuration.

Table 4.12: Neutron Flux and Fission Rate for New Reflector Arrangement

Reflector Materials	**Mean Thermal Flux**	**Mean Fast Flux**	**Mean Fission Rate**
Copper	0.0025	0.00978	0.000179
Stainless Steel	0.002521	0.00977	0.000179
Tungsten	0.002525	0.00974	0.0001799
Beryllium Copper	0.0025	0.009796	0.0001796
Tungsten Carbide	0.00242	0.00974	0.000179

One significant difference between this reflector arrangement and the one previously used is that in this reflector arrangement, all the reflectors give almost the same results. This could be due to the moderation property of light water. The neutrons must go through light water before getting to the reflector material being investigated. Thus, making it challenging to analyze the behavior of the reflector being studied. Despite the similarity in results, tungsten still shows a slightly higher mean fission rate than the other reflector materials.

Table 4.13: K_{eff} for New Reflector Arrangement

Reflector Materials	K_{eff}	K_{eff} **Uncertainty**
Copper	1.15	+/- 0.00084
Stainless Steel	1.15	+/- 0.0009
Tungsten	1.15	+/- 0.00074
Beryllium Copper	1.15	+/- 0.00077
Tungsten Carbide	1.15	+/- 0.00083

K_{eff} increased in this arrangement as shown in Table 4.13. It can be concluded that the higher the reactivity, the lower the distribution uniformity ratio. A relationship between the reactivity coefficient K_{eff} and the thermal flux uniformity ratio for the first configuration (Figure 3.2) is shown below, with the error bars.

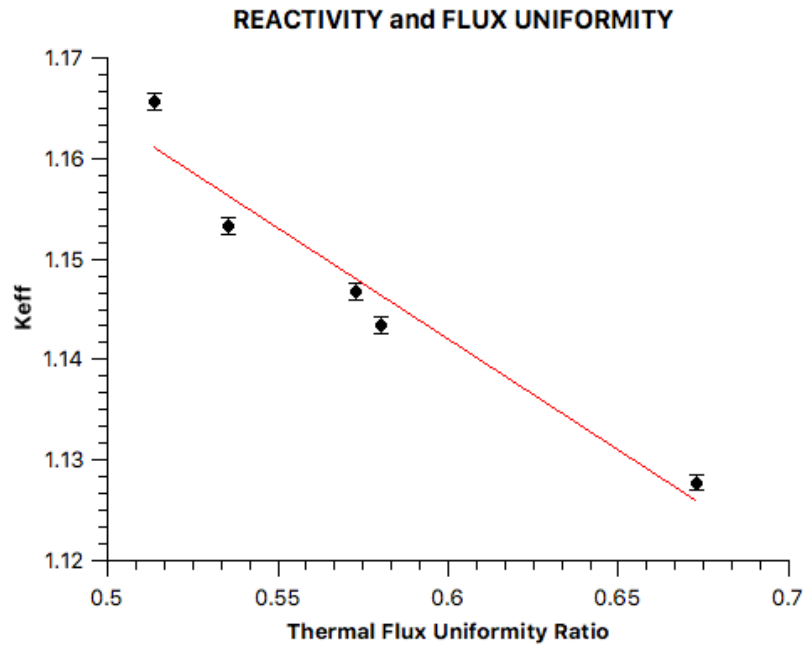


Figure 4.9: A Different Reflector Configuration

The coefficient of linearity r^2 is 0.947. Therefore, the two quantities are fairly related to each other. The equation of the line of best fit is:

$$y = -0.22x + 1.27 \tag{4.2}$$

this equation's slope is negative, proving the inverse proportionality of the two quantities. The smaller K_{eff}, the better the uniformity ratio.

4.3.1 A Different Configuration

For further investigation, aluminium oxide (Al_2O_3) was added to the fuel assembly in some of the control rod regions in this new configuration, as shown in Figure 4.10 below.

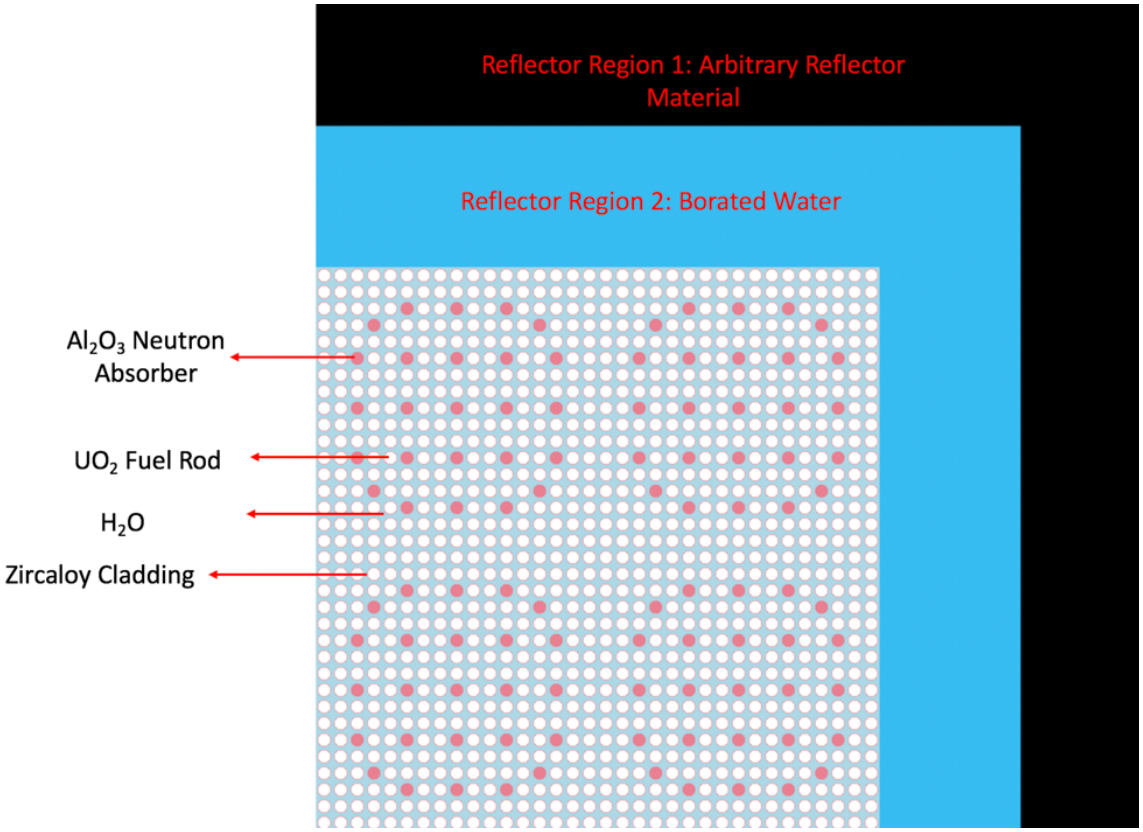


Figure 4.10: Modified Assembly

The results are shown in Table 4.14 below.

Table 4.14: Modified Assembly Results

Reflector Materials	K_{eff}	K_{eff} **Uncertainty**
Copper	1.09625	+/- 0.00082
Stainless Steel	1.09921	+/- 0.00093
Tungsten	1.09899	+/- 0.00082
Beryllium Copper	1.0973	+/- 0.00077
Tungsten Carbide	1.09558	+/- 0.00077

Since K_{eff} decreases, the uniformity distribution ratio is expected to increase. The Uniformity distribution for the thermal flux and fission rate is recorded in Table 4.15

Table 4.15: Uniformity Distribution for the Modified Fuel Assembly

Reflector Materials	**Thermal Flux Uniformity Ratio**	**Fission Rate Uniformity Ratio**
Copper	62.2%	61.9%
Stainless Steel	62.5%	50.5%
Tungsten	61.7%	61.3%
Beryllium Copper	62.2%	60%
Tungsten Carbide	62.3%	62.3%

Overall, the uniformity distribution for all the reflector materials increases.

4.4 Result Summary

A good reflector has a low neutron absorption and a high neutron scattering. Table 4.9 and Table 4.10 prove that tungsten can be a good reflector option in this reactor type compared to all the reactor materials studied because it scatters more than it absorbs. Tungsten also increases the fission rate more than the other reflector materials.

In Table 4.5 and 4.6, tungsten has the second lowest absorption rate for both fast and thermal neutrons and shows a constant behavior. The rest of the reflector materials do best in one absorption category and worst in the other. Tungsten, however, behaves well in both thermal and fast neutron absorption.

One of the properties of a good reflector is to flatten the flux distribution, as mentioned in Section 2.3. Although tungsten is a good reflector choice, its flux and fission rate distribution is the lowest, about 50% on average. Tungsten carbide has the highest uniformity distribution ratio on average, about 67%. However, it gives the lowest fission rate and scatters neutrons the least. This is shown in table 4.7 and 4.8, where tungsten has the highest scattering rate for both thermal and fast neutrons. Tungsten carbide has the lowest.

Based on the analysis, it can be implied that a high uniformity distribution ratio results in a high neutron absorption, which results in a low K_{eff}. The K_{eff} then increases when more neutrons are scattered back, which also increases the fission rate. In the configuration analyzed in Section 4.4.1, the K_{eff} was decreased because the neutron absorber Al_2O_3 was inserted, which led to an increase in uniformity. This aims to show that the flux can be flattened by adding neutron absorbers in the core.

Chapter 5

Conclusion and Discussion

The development of the SMART design has led to many design concepts. Although the realization is underway, there is room for many fascinating design improvements. This project aims to study the behavior of different types of reflectors in the SMART reactor. The importance of this work is to show that some changes in the SMART design could increase its power output without altering the size of the reactor. The theory of neutron physics was used to understand the fission reaction in a nuclear reactor.

A simple and small-size model of the SMART reactor was simulated using the Monte Carlo source code OpenMC. Various quantities were calculated to help describe the use of the different reflector materials in the reactor. One of the most essential takeaways from this project is that the more uniformity, the less the reaction rate. Although it is necessary to use a reflector that increases the fission rate, using a reflector that results in a uniform distribution is also ideal. If one chooses the latter, components like nuclear fuel enrichment and fuel lengths... could be easily altered to increase the power output and keep a uniform distribution. This is a simple design; therefore, many aspects have been ignored. A uniform fuel enrichment was used for all the fuel rods.

The analysis helped us understand various phenomena in the reactor, such as how neutron scattering and absorption in the reflector could affect the functioning of the reactor core. Different reactor arrangements can also lead to different results. It is important to investigate other reflector arrangements, as shown in Section 4.6. The best reflector arrangement here is the first arrangement used (Figure 3.3) compared to the one used in Section 4.4 to study the behaviour of different reflector materials. The results vary more in this configuration compared to the second configuration in Section 4.6 (Figure 4.8), where the results are more uniform (Table 4.12). Further research could include changing the thickness of the reflectors and using more alloy materials as reflectors. A good understanding of material physics is also necessary. Modeling the full assembly can add more benefits to the study.

Appendix A

OpenMC Code

```
# In[1]:
    from IPython.display import Image
    import matplotlib.pyplot as plt
    import pandas as pd
    import numpy as np
    import os
    import openmc
    # In[2]:
    print(os.environ['OPENMC_CROSS_SECTIONS'])
    # Cell 3 is material compositions definition cell. In this model, the material of
reflector is Tungsten.
    # Creating a new material by "openmc.Material" class and named 'Tungsten'.
Once a material has been instantiated, nuclides can be added with Material.add_nuclide()
and elements can be added with Material.add_element().
    # In[3]:
    # 4.8 enriched fuel
    fuel=openmc.Material(name='high enrichment fuel')
    fuel.set_density('g/cm3',10.97)
    fuel.add_element('U', 1. , enrichment=4.8)
    fuel.add_element('O', 2.)
    #Zircaloy Cladding
    zircaloy=openmc.Material(name='Zircaloy')
    zircaloy.set_density('g/cm3',6.56)
    zircaloy.add_element('Zr',0.978,'wo')
    zircaloy.add_element('Sn',0.012,'wo')
    zircaloy.add_element('Nb',0.01 ,'wo')
    #Water
    water=openmc.Material(name='water')
    water.set_density('g/cm3',1.00)
    water.add_nuclide('H1', 2.00)
    water.add_nuclide('O16',1.00)
    # borated water
```

```
boratedwater = openmc.Material(name='boratedwater')
boratedwater.set_density('g/cm3', 0.740582)
boratedwater.add_nuclide('H1', 4.9457e-2)
boratedwater.add_nuclide('O16', 2.4732e-2)
boratedwater.add_nuclide('B10', 8.0042e-6)
#Aluminium Oxide
aluminiumoxide=openmc.Material(name='aluminiumoxide')
aluminiumoxide.set_density('g/cm3', 3.95)
aluminiumoxide.add_element('Al', 2.00)
aluminiumoxide.add_element('O', 3.00)
# Tungsten reflector material
tungsten=openmc.Material(name='tungsten')
tungsten.set_density('g/cm3',19.3)
tungsten.add_element('Fe', 1.00)
materials_file=openmc.Materials([fuel,zircaloy, boricacid, water, boratedwater,
tungsten])
materials_file.export_to_xml()
# In[4]:
# Fuel Geometry
# Create cylinder for the fuel and clad
fuel_outer=openmc.ZCylinder(R=0.4465)
clad_outer=openmc.ZCylinder(R=0.475)
#create the surfaces for cells
left=openmc.XPlane(x0=-0.6)
right=openmc.XPlane(x0=0.6)
up=openmc.YPlane(y0=0.6)
down=openmc.YPlane(y0=-0.6)
top=openmc.ZPlane(z0=+100.0,boundary_type =' vacuum')
bot=openmc.ZPlane(z0=-100.0,boundary_type =' vacuum')
# In[5]:
fuel_region= -fuel_outer  -top  + bot
absorber_region=-clad_outer  -top  + bot
clad_region= +fuel_outer  -clad_outer  -top  + bot
water_region= +left  -right  +down  -up  +clad_outer  -top  + bot
fuelpin_region=+left  -right  +down  -up  -top  + bot
In[6]:
#Fuel cell
fuel_cell=openmc.Cell(name='enrichment fuel')
fuel_cell.fill=fuel
fuel_cell.region=fuel_region
#Cladding cell
clad_cell=openmc.Cell(name='cladding')
clad_cell.fill=zircaloy
clad_cell.region=clad_region
```

#Moderator cell
moderator_cell=openmc.Cell(name='moderator')
moderator_cell.fill=water
moderator_cell.region= water$_region$
fuelpin_universe=openmc.Universe(name='fuel pin universe')
fuelpin_universe.add_cells([fuel_cell,clad_cell,moderator_cell])
fuelpin_cell=openmc.Cell(name='fuel pin')
fuelpin_cell.region=fuelpin_region
fuelpin_cell.fill=fuelpin_universe
# In[7]:
#Neutron absorber cell(control rods)
rod_cell=openmc.Cell(name='burnable poison')
rod_cell.fill=water
rod_cell.region=absorber_region
# Moderator cell
moderator_cell=openmc.Cell(name='moderator')
moderator_cell.fill=water
moderator_cell.region=water_region
controlrod_universe=openmc.Universe(name='control rod')
controlrod_universe.add_cells([rod_cell,moderator_cell])
controlrod_cell=openmc.Cell(name='control rod')
controlrod_cell.region=fuelpin_region
controlrod_cell.fill=controlrod_universe
# In[8]:
f=fuelpin_universe
c=controlrod_universe
# Instantiate the A type fuel assembly lattice
PWR_lattices=openmc.RectLattice(name='Fuelpin_lattice')
PWR_lattices.dimension=[17,17]
PWR_lattices.lower$_left = [-10.2, -10.2]$
PWR_lattices.pitch=[1.20,1.20]
PWR_lattices.universes=
[[f,f,f,f,f,f,f,f,f,f,f,f,f,f,f,f,f],
[f,f,f,f,f,f,f,f,f,f,f,f,f,f,f,f,f],
[f,f,f,f,f,c,f,f,c,f,f,c,f,f,f,f,f],
[f,f,f,c,f,f,f,f,f,f,f,f,c,f,f,f],
[f,f,f,f,f,f,f,f,f,f,f,f,f,f,f,f,f],
[f,f,c,f,f,c,f,f,c,f,f,c,f,f,c,f,f],
[f,f,f,f,f,f,f,f,f,f,f,f,f,f,f,f,f],
[f,f,f,f,f,f,f,f,f,f,f,f,f,f,f,f,f],
[f,f,c,f,f,c,f,f,c,f,f,c,f,f,c,f,f],
[f,f,f,f,f,f,f,f,f,f,f,f,f,f,f,f,f],
[f,f,f,f,f,f,f,f,f,f,f,f,f,f,f,f,f],
[f,f,c,f,f,c,f,f,c,f,f,c,f,f,c,f,f],

[f,f,f,f,f,f,f,f,f,f,f,f,f,f,f,f,f],
[f,f,f,c,f,f,f,f,f,f,f,f,f,c,f,f,f],
[f,f,f,f,f,c,f,f,c,f,f,c,f,f,f,f,f],
[f,f,f,f,f,f,f,f,f,f,f,f,f,f,f,f,f],
[f,f,f,f,f,f,f,f,f,f,f,f,f,f,f,f,f]]
assembly_cell=openmc.Cell(name='assembly')
assembly_cell.fill=PWR_lattices
assembly_universe=openmc.Universe(name='nuclear assembly')
assembly_universe.add_cell(assembly_cell)
# In[9]:
# Defining surfaces to surround the assembly region array geometry
min_x = openmc.XPlane(x0=+5.0, boundary_type='vacuum') max_x = openmc.XPlane(x0=+4
boundary_type='transmission')
min_y = openmc.YPlane(y0=+5.0, boundary_type='vacuum')
max_y = openmc.YPlane(y0=+45.8, boundary_type='transmission')
min_z = openmc.ZPlane(z0=-100.0,boundary_type='vacuum') max_z = openmc.ZPlane(z0=+1
min_x.periodic_surface=min_y
# In[10]:
core_region= +min_x -max_x +min_y -max_y +min_z -max_z
# In[11]:
n=assembly_universe
# Creating 9 assemblies array
core_lattices=openmc.RectLattice(name='9 assemblies array')
core_lattices.dimension=[2,2]
core_lattices.lower$_left = [+5.0, +5.0]$
core_lattices.pitch=[20.4,20.4]
core_lattices.universes= [[n,n], [n,n]]
core_cell=openmc.Cell(name='array')
core_cell.fill=core_lattices
core_cell.region=core_region
core_universe=openmc.Universe(name='assembly array') core_universe.add_cell(core_cell)
# In[12]:
# Defining surfaces to surround the reflector region geometry
refX_max = openmc.XPlane(x0=+56.0, boundary_type='transmission')
refY_min = openmc.YPlane(y0=+56.0, boundary$_type =' transmission'$)
# In[13]:
reflector1_region= +max$_x$ $- refX_max + min_y - refY_min + min_z -$
max_z
reflector2_region= +min_x -max_x +max_y -refY_min +min_z -max_z
# The below cell aim to link the reflector materials to reflector geometry which
already defined in the above codes.
# In[14]:
# Defining the reflector cell surrounding the assemblies
reflector1_cell=openmc.Cell(name='reflector right')

reflector1_cell.fill=tungsten
reflector1_cell.region=reflector1_region
reflector2_cell=openmc.Cell(name='reflector down')
reflector2_cell.fill=tungsten
reflector2_cell.region=reflector2_region
# In[15]:
# Defining the moderator cell outside of the reflector
ColX_max = openmc.XPlane(x0=+66.2, boundary_type='vacuum')
ColY_min = openmc.YPlane(y0=+66.2, boundary_type='vacuum')
# In[16]:
coolant1_region= +refX_max -ColX_max +min_y -ColY_min +min_z -max_z
coolant2_region= +min_x -refX_max +refY_min -ColY_min +min_z -max_z
# In[17]:
coolant1_cell=openmc.Cell(name='coolant right')
coolant1_cell.fill=water
coolant1_cell.region=coolant1_region
coolant2_cell=openmc.Cell(name='coolant down')
coolant2_cell.fill=water
coolant2_cell.region=coolant2$_region$
# In[18]:
reflector_universe=openmc.Universe(name='reflector university')
reflector_universe.add_cell(reflector1_cell)
# In[19]:
root=openmc.Universe(name='root')
root.add_cells((core_cell,reflector1_cell,reflector2_cell,coolant1_cell,coolant2_cell))
openmc_geometry=openmc.Geometry(root)
openmc_geometry.export_to_xml()
# In[20]:
p=openmc.Plot()
p.filename='PWR_assembly'
p.width=(100,100)
p.pixels=(3000,3000)
p.origin=(15.0,15.0,0.0)
p.basis='xy'
p.color_by='material'
p.colors=fuel:'white',water:'lightblue',zircaloy:'red',tungsten:'black'
plots=openmc.Plots([p])
plots.export_to$_xml$()
openmc.plot_geometry()
# In[21]:
# In order to obtain estimates of physical quantities in the simulation, one or more tallies were created using the openmc.Tally class. In this case, a mesh filter

was created to cover the geometry on 2D, and link the filter to 'flux' and 'fission' scores. Fission rate, flux distribution, absorption and scattering rate were obtained from simulation process.

In[21]:

Instantiate an empty tallies object

tallies$_file = openmc.Tallies()$

Creating a rough mesh for saving calculating time

mesh=openmc.Mesh()

mesh.type='regular'

mesh.dimension=[17*3, 17*3]

mesh.lower_left =[+5.0,+5.0]

mesh.upper_right=[+66.2,+66.2]

Setting mesh to the openmc mesh tally filter

mesh_filter=openmc.MeshFilter(mesh)

Creating a energy filter to opnemc tally

energy_filter=openmc.EnergyFilter([0, 0.625, 20.0e6])

Crearing a material filter contains reflector material material_filter=openmc.MaterialFilter(tungst

Creating a mesh fission rate tally to openmc tally=openmc.Tally(name='mesh tally') tally.filters=[mesh$_filter]tally.scores = ['fission']tallies_file.append(tally)$

Create an absorption rate tally

tally=openmc.Tally(name='absorption tally')

tally.filters = [mesh$_filter]$

tally.filters.append(energy_filter)

tally.filters.append(material_filter)

tally.scores=['absorption']

tallies_file.append(tally)

#Create a scattering rate tally

tally=openmc.Tally(name='scattering tally')

tally.filters = [mesh$_filter]$

tally.filters.append(energy_filter)

tally.filters.append(material_filter)

tally.scores=['scatter']

tallies_file.append(tally)

Creating a neutron flux tally to openmc

tally=openmc.Tally(name='flux tally')

tally.filters=[mesh_filter]

tally.filters.append(energy$_filter)$

tally.scores=['flux']

tallies_file.append(tally)

Export all tallies to tallies.xml file

tallies_file.export_to_xml()

In[22]:

OpenMC simulation parameters

settings = openmc.Settings()

settings.batches =250
settings.inactive = 100
settings.particles = 10000
settings.energy_mode='continuous-energy'
sour1=openmc.Source()
sour1.space=openmc.stats.Box((+5.0,+5.0,-100),
(+45.8,+45.8,+100.0),only_fissionable=True)
settings.source=[sour1]
settings.run_mode = 'eigenvalue'
settings.export_to$_xml$()
In[23]:
Remove old HDF5 files
get$_ipython$().$system$('$rmstatepoint.*$')
openmc.run()
# The simulation ran successfully and a statepoint file is created with all the tally data in it. The analysis begins loading the statepoint file and 'reading' the results. By default, data from the statepoint file is only read into memory when it is requested.
# In[24]:
# Load the statepoint file
sp=openmc.StatePoint('statepoint.250.h5')
# Next we need to get the tally, which can be done with the StatePoint.get_tally(...) method.
# In[25]:
tally=sp.get_tally(name='flux tally')
# In[26]:
flux=tally.get_slice(scores=['flux'])
thermal_flux=flux.get_slice(filters=[openmc.EnergyFilter], filter_bins=[((0, 0.625),)])
fast_flux=flux.get_slice(filters=[openmc.EnergyFilter], filter_bins=[((0.625,20e6),)])
# In[27]:
thermal_flux.mean.shape=(17*3,17*3)
plt.figure()
plt.imshow(thermal$_flux.mean, interpolation =' none', origin =' lower')plt.colorbar$()
# In[28]:
a1=thermal_flux.mean
print(a1)
print(a1.shape)
In[29]:
Slicing the square 1 (left bottom) from 4 assemblies-reflector model s1=a1[:17,:17]
#print(s1)
print(s1.shape)
print(np.max(s1))
z1=a1[:17,:17]
print(np.mean(z1))

```
plt.imshow(s1, interpolation='none',origin='lower')
# In[30]:
# Slicing the square 2 (right bottom) from 4 assemblies-reflector model
#print(a1[0,17],a1[0,18],a1[1,17],a1[1,18])
s2=a1[:17,17:34]
#print(s2)
print(s2.shape)
print(np.max(s2))
z2=a1[:17,17:34]
print(np.mean(z2))
plt.imshow(s2, interpolation='none',origin='lower')
# In[31]:
# Slicing the square 3 (left up) from 4 assemblies-reflector model
#print(a1[17,0],a1[18,0],a1[17,1],a1[18,1])
s3=a1[17:34,:17]
#print(s3)
print(s3.shape)
print(np.max(s3))
z3=a1[17:34,:17]
print(np.mean(z3))
plt.imshow(s3, interpolation='none',origin='lower')
# In[32]:
# Slicing the square 4 (right up) from 4 assemblies-reflector model
s4=a1[17:34,17:34]
#print(s4)
print(s4.shape)
print(np.max(s4))
z4=a1[17:34,17:34]
print(np.mean(z4))
plt.imshow(s4, interpolation='none',origin='lower')
# In[33]:
t=thermal_flux.mean
print(np.max(t))
print (np.mean(t))
print(np.where(t==np.max(t)))
radial_thermal=t.diagonal()
print(radial_thermal.reshape(51,1))
# In[34]:
fast_flux.mean.shape=(17*3,17*3)
plt.figure()
plt.imshow(fast_flux.mean, interpolation = 'none', origin = 'lower')
plt.colorbar()
# In[35]:
a2=fast_flux.mean
```

```
#print(a2)
print(a2.shape)
# In[36]:
# Slicing the square 1 (left bottom) from 4 assemblies-reflector model
f1=a2[:17,:17]
#print(f1)
print(f1.shape)
print(np.max(f1))
p1=a1[:17,:17]
print(np.mean(p1))
plt.imshow(f1, interpolation='none',origin='lower')
# In[37]:
# Slicing the square 2 (right bottom) from 4 assemblies-reflector model
f2=a2[:17,17:34]
#print(f2)
print(f2.shape)
print(np.max(f2))
p2=a2[:17,17:34]
print(np.mean(p2))
plt.imshow(f2, interpolation='none',origin='lower')
# In[38]:
# Slicing the square 3 (left up) from 4 assemblies-reflector model
f3=a2[17:34,:17]
#print(s3)
print(f3.shape)
print(np.max(f3))
p3=a2[17:34,:17]
print(np.mean(p3))
plt.imshow(f3, interpolation='none',origin='lower')
# In[39]:
# Slicing the square 4 (right up) from 4 assemblies-reflector model
f4=a2[17:34,17:34]
#print(s4)
print(f4.shape)
print(np.max(f4))
p4=a2[17:34,17:34]
print(np.mean(p4))
plt.imshow(f4, interpolation='none',origin='lower')
# In[40]:
f=fast_flux.mean.reshape(51, 51)
print(np.max(f))
print(np.mean(f))
print (np.where(f==np.max(f)))
f0=f[:,0]
```

print(f0.reshape(51,1))
radial$_f = f.diagonal()$
print(radial$_f.reshape(51,1)$)
# In[41]:
tally=sp.get_tally(name='absorption tally')
# In[42]:
absorption=tally.get$_slice(scores = ['absorption'])$
thermal_absorption=absorption.get_slice(filters=[openmc.EnergyFilter], filter_bins=[((0, 0.625),)])
fast_absorption=absorption.get_slice(filters=[openmc.EnergyFilter], filter_bins=[((0.625,20e6),)
# In[43]:
thermal_absorption.mean.shape=(17*3,17*3)
plt.figure()
plt.imshow(thermal$_absorption.mean, interpolation =' none', origin =' lower', cmap =' plasma')$
plt.colorbar()
# In[44]:
a=thermal_absorption.mean.reshape(51,51)
print(np.max(a))
print(np.mean(a))
print (np.where(a==np.max(a)))
# In[45]:
fast_absorption.mean.shape=(17*3,17*3)
plt.figure()
plt.imshow(fast$_absorption.mean, interpolation =' none', origin =' lower', cmap =' plasma')$
plt.colorbar()
# In[46]:
p=fast_absorption.mean.reshape(51,51)
print(np.max(p))
print(np.mean(p))
print (np.where(p==np.max(p)))
# In[47]:
tally=sp.get_tally(name='scattering tally')
# In[48]:
scatter=tally.get_slice(scores=['scatter'])
thermal_scatter=scatter.get_slice(filters=[openmc.EnergyFilter], filter_bins=[((0, 0.625),)])
fast_scatter=scatter.get_slice(filters=[openmc.EnergyFilter], filter_bins=[((0.625,20e6),)])
# In[49]:
thermal$_scatter.mean.shape = (17 * 3, 17 * 3)$
plt.figure()
plt.imshow(thermal_scatter.mean, interpolation='none',origin='lower',cmap='plasma')
plt.colorbar()

In[50]:

a=thermal$_s catter.mean.reshape(51, 51)$

print(np.max(a))

print(np.mean(a))

print (np.where(a==np.max(a)))

In[51]:

fast$_s catter.mean.shape = (17 * 3, 17 * 3)$

plt.figure()

plt.imshow(fast$_s catter.mean, interpolation =' none', origin =' lower', cmap =' plasma')$

plt.colorbar()

In[52]:

p=fast_scatter.mean.reshape(51,51)

print(np.max(p))

print(np.mean(p))

print (np.where(p==np.max(p)))

In[53]:

Get a pandas dataframe for the mesh tally data

df = tally.get$_p andas_d ataframe(nuclides = False)$

In[54]:

Extract the scatter tally data from pandas

scatter = df[df['score'] == 'scatter']

scatter['rel. err.'] = scatter['std. dev.'] / scatter['mean']

Show a scatter plot of the mean vs. the std. dev. scatter.plot(kind='scatter', x='mean', y='rel. err.', title='Scattering Rates')

In[55]:

scatter['mean'].plot(kind='hist', bins=25)

scatter['mean'].plot(kind='kde')

plt.title('Scattering Rates')

plt.xlabel('Mean')

plt.legend(['KDE', 'Histogram'])

In[56]:

tally=sp.get$_t ally(name =' meshtally')$

In[57]:

fission=tally.get_slice(scores=['fission'])

fission.mean.shape=(17*3, 17*3)

plt.figure()

plt.imshow(fission.mean, interpolation='none',origin='lower',cmap='jet')

plt.colorbar()

In[58]:

a3=fission.mean

#print(a3)

print(a3.shape)

In[59]:

```
# Slicing the square 1 (left bottom) from 4 assemblies-reflector model
r1=a3[:17,:17]
#print(r1)
print(r1.shape)
print(np.max(r1))
v1=a3[:17,:17]
print(np.mean(v1))
plt.imshow(r1, interpolation='none',origin='lower')
# In[60]:
# Slicing the square 2 (right bottom) from 4 assemblies-reflector model
#print(a1[0,17],a1[0,18],a1[1,17],a1[1,18])
r2=a3[:17,17:34]
#print(r2)
print(r2.shape)
print(np.max(r2))
v2=a3[17,17:34]
print(np.mean(v2))
plt.imshow(r2, interpolation='none',origin='lower')
# In[61]:
# Slicing the square 2 (right bottom) from 4 assemblies-reflector model
#print(a1[0,17],a1[0,18],a1[1,17],a1[1,18])
r3=a3[:17,17:34]  print(r3)
print(r3.shape)
print(np.max(r3))
v3=a3[17:34,:17]
print(np.mean(v3))
plt.imshow(r3, interpolation='none',origin='lower')
# In[62]:
# Slicing the square 4 (right up) from 4 assemblies-reflector model
r4=a3[17:34,17:34]
#print(s4)
print(r4.shape)
print(np.max(r4))
v4=a3[17:34,17:34]
print(np.mean(r4))
plt.imshow(r4, interpolation='none',origin='lower')
# In[63]:
fr=fission.mean
print(np.max(fr))
print(np.mean(fr))
print (np.where(fr==np.max(fr)))
radial=fr.diagonal()
print(radialfr.reshape(51, 1))
```

Bibliography

[1] Department of Nuclear Energy International Atomic Energy Agency. Small Modular Reactors. *IAEA Bulletin*, 2018.

[2] Department of Energy-Office of Nuclear Energy. Small Modular Reactors. *Archive*, 2010.

[3] World Nuclear Association. Safety of Nuclear Reactors. *Archive*, 2019.

[4] Ernest Moniz. Why we still need nuclear power: Making clean energy safe and affordable. *Foreign Affairs*, 90(6):83–94, 2011.

[5] Department of Nuclear Energy International Atomic Energy Agency. Small Modular Reactors: a Chanllenge for Spent Fuel Management. *IAEA Bulletin*, 2019.

[6] J.Kang et F.N. Von Hippe. U-232 and the proliferation-resistance of u-233 in spent fuel. *Science & Global Security*, 9:1, 2001.

[7] Advanced Reactors Information Sytems International Atomic Energy Agency. System-Integrated Modular Advanced Reactor (SMART). *Status Report*, 77, 2011.

[8] D Benov. The Manhattan Project, the first electronic computer and the Monte Carlo method. *Monte Carlo Methods and Applications*, 22(1):73–79, 2016.

[9] A.J. Baratta J. R. Lamarsh. Introduction to Nuclear Engineering. *Prentice-Hall*, 3d ed, 2001.

[10] Advanced Reactors Information Sytems International Atomic Energy Agency. System-Integrated Modular Advanced Reactor (SMART). *Status Report*, 77, 2011.

[11] Zee S Q Lee CC, Hwang D H and Chang M H. Nuclear and Thermal Hydraulic Design Characteristics of the SMART Core. *International Conference on Global Environment and Advanced Nuclear Power Plants, Kyoto, Japan*, 2003.

[12]

[13] Sim Hyung Jin Kim Chang Hyo Lee Jung Chan Ji Sung Kyun Jung, Jong Sung. Monte Carlo Depletion Analysis of SMART Core by MCNAP Code. *Proceedings of the Korean Nuclear Society spring meeting*, 2001.