# Benchmarking of G4STORK for the Coolant Void Reactivity of the Super Critical Water Reactor Design

## Acknowledgements

There are many people that deserve thanks for supporting me during my thesis. First, I would like to thank my supervisor, Dr. Adriaan Buijs, for guiding me through the last two years and keeping me on track. Next, I would like to thank my project members, Andrew Tan and Salma Mahzooni, for diligently working with me on the G4STORK project and supplying some much needed distraction. Liam Russell, I would like to thank for introducing me to the G4STORK project and mentoring me during my early months. I would also like to thank Michel Nowak for supplying the MCNP input files for the SCWR reactor. I would also like to thank Jason Sharpe for coordinating the SCWR code comparison and answering my questions. The graduate students of JHE 327, deserve my thanks as well for creating a great work environment and providing helpful advice. I would also like to thank my housemates for putting up with my disorganization and supplying some much needed humor. Finally, I would like to thank my family and friends for helping to motivate and entertain me during this process.

## Tips for Navigating This Document

Many hotlinks have been created in this document to allow the reader to efficiently navigate this document. If the user wants to navigate directly to a chapter or subchapter, there are two methods for doing so. The first way is to activate the appropriate hyperlink. In Adobe Reader this done by clicking (use ctrl + click for Microsoft word) the hyperlink in the table of contents. The second way is to bring up the bookmark menu in Adobe Reader (the headings tab under the navigation tool in word), and click on the appropriate link. Similarly, specific tables, figures, or equations can be found by activating the relevant hyperlink in the list of tables, list of figures, or the list of equations. Throughout this document, when another place in the document is referred to (a chapter, table, figure, or equation) for further explanation, a hyperlink will usually be present in the text. These hyperlinks can be activated by the user to quickly jump to the place in the document that is being referred to. Commonly used acronyms, and terms without obvious meaning have also been hyperlinked for the benefit of the reader. By clicking on the word itself (activating the hyperlink) the user will be redirected to the glossary, where the word is described in greater detail. Note that if the word does not exist in the glossary, then nothing will happen. Finally, the references have been hyperlinked as well, allowing the user to jump to the IEEE description of the source being referred to.

## Abstract

The objectives of this thesis were the validation of G4STORK to use it for the investigation of the SCWR lattice cell. MCNP6 was chosen as the program that the methodology of G4STORK would be validated against. Over multiple steps, the methodology of G4STORK was matched to that of MCNP6 (described here, 3.4). After each step, the output of the two programs were compared, allowing us to pinpoint why and where discrepancies came about. At the end of this process, we were able to show that when G4STORK used the same assumptions as MCNP6, it produced similar results (shown here, 4.1.4). The results of G4STORK simulating the SCWR lattice cell, using its more accurate default methodology, was then compared to those of MCNP6 (shown here, 4.2.1). Large differences in the results were seen to occur, because of the inaccurate assumptions used by MCNP6, during transient cases. We concluded that despite the existence of minor discrepancies between the results of MCNP and G4STORK for some cases, G4STORK is still the theoretically more accurate method for simulating lattice cell cases such as these, due to MCNP's use of the generational method.


## Table of Contents

## List of Tables

## List of Figures

## List of Equations

## Glossary

CNL: Stands for Canadian Nuclear Laboratories, is Canada's primary nuclear research institution. Was formerly known as Atomic Energy Canada Limited (AECL).

SCWR: Stands for Super Critical Water Reactor, is fourth generation pressurized heavy water nuclear reactor design created by CNL. For further information, refer to section 1.2.

GEANT4: Stands GEometry ANd Tracking, it is toolkit for creating programs that track particles through time and matter.

G4STORK: Stands for GEANT4 Stochastic Reactor Kinetics, is an open source program built off GEANT4 for simulating nuclear reactor physics. For more information, refer to section 2.6.8.

MCNP: Stands for Monte Carlo N-Particle code, it is a program that tracks particles through time and matter (often used in the field of nuclear reactor physics). For more information, refer to section 2.6.7

NeutronHP: Stands for the Neutron High Precision model, it is an empirical model used to describe then neutron interactions between the $10^{-5}$eV to 20MeV incoming neutron energy range. For more information, refer to section 2.6.5.

G4NDL: Stands for Geant4 Neutron Data Library, it is the empirical dataset used by the NeutronHP model in GEANT4.

$K_{eff}$: is the ratio of the rate of neutron production, to the rate of neutron loss, within a reactor. For more information, refer to section 2.4.

$K_{gen}$: is the ratio of the number of neutrons produced by the current fission generation, to the number of neutrons produced by the previous fission generation. For more information, refer to section 2.6.1.2.

$K_{dyn}$: is the ratio of the number of neutrons produced, to the number of neutrons lost, over a set time interval. For more information, refer to section 2.6.1.1.

Flooring: the process of converting a real number to an integer by removing the decimal component of the number

# 1 Introduction

## 1.1 Nuclear Technology

Nuclear power is one of the most economical sources of non-carbon emitting power (second only to hydroelectric dams). However, there is a lot of fear about the use of nuclear power from the public, due to the perceived severity of nuclear accidents when they occur (the danger of radiation has been somewhat inflated by modern culture). This fear has driven many countries to choose more expensive non-carbon emitting energy sources and has made the nuclear industry obsessed with safety. Even though nuclear reactor research has received lots of funding from the government and from industry in the past, it is still one of the slowest changing technologies. This is because nuclear technology is held to far higher standard of safety than just about any other technology. Therefore, any changes in the design of a nuclear reactor, no matter how small, must be rigorously validated before they are allowed to become a reality. In the past this meant constructing very expensive experiments and prototype plants, putting a heavy economic burden on the nuclear industry.

With the advancement of computer technology, more and more of these experiments can be simulated on a computer for a fraction of the cost [1]. Computer simulations often only take minutes or hours to set up and run, whereas physical experiments often take days to setup and run. Additionally, many different computer simulations can be run at the same time by different researchers, whereas with physical experiments, multiple researchers

often share the same experimental facility. Both of these advantages allow computer simulations to streamline much of the design process. Another key advantage of using computer simulations is that they can simulate scenarios that would be nearly impossible to replicate experimentally. This in turn allows researchers to improve the overall safety of the design. Finally, computer simulations can be run safely by anyone, whereas running experiments puts expensive equipment and researchers at risk.

The more complicated transients that occur in a nuclear reactor (such as accident scenarios) can be very difficult and time consuming to model accurately using a computer simulation. Historically this is why computational models are usually limited to simulating static cases (non-transient), or semi static cases (such as full power criticality or fuel depletion). With recent improvements in computing power however, time dependent nuclear physics models have become more popular for studying transient cases in nuclear reactor research. Deterministic models (codes that try to directly solve the neutron transport equation) are popular choices in industry because of the speed at which they can obtain results. However, the accuracy of these simulations is limited by the approximations that the programs make in order to solve the neutron transport equation. Even the more complicated space-energy dependent three dimensional codes suffer from inaccuracies caused by the necessary discretization of space and energy.

Monte Carlo simulations follow neutrons through space, energy and time in a given reactor geometry, without the need for discretization and the other approximations required by a deterministic approach. The only factors that limit the accuracy of a Monte Carlo model is the nuclear data set used to sample what processes will occur and how they will occur, and the statistical uncertainty caused by the random sampling of events (dependent on the number of neutrons simulated). As newer computing technology lets researchers use higher precision data sets and simulate more neutrons, these sources of inaccuracy continue to become less significant.

Unfortunately, the added accuracy that the Monte Carlo method offers comes at a cost of speed. Monte Carlo codes run far slower than many deterministic codes, making them unsuitable replacements for static, near static, or long duration experiments (testing refueling schemes). In these cases, the small gain in accuracy is not worth the extra run time. However, accident scenarios, during which the properties of the neutron population are changing rapidly over time, often require a high degree of accuracy, but do not require a lot of simulation time, and are thus particularly suited to the Monte Carlo method. As mentioned before, safety is paramount when it comes to nuclear technology and thus refining the methods that researchers use to assess the safety of a nuclear reactor is very important. Improving the accuracy with which we can assess the safety of the reactor does not only serve to assure the regulators and the public, it also opens the possibility of allowing operators to adjust the safety margins (ex: the regulated power threshold) and thus increase the potential revenue of the reactor.

## 1.2 Canadian Super Critical Water Reactor

The current Canadian SCWR concept is meant to combine the advantages of heavy water reactors, boiling water reactors and supercritical water fossil-fired plants. The main design goal of the SCWR is the improvement of the thermodynamic efficiency over current Canadian CANDU reactors, and take advantage of the superior heat transfer properties of supercritical water [2]. The use of supercritical water coolant allows the SCWR design to achieve a thermodynamic efficiency of 45-50%, which is greatly improved from the typical 33% of the CANDU6. Since the supercritical water coolant remains in a single state throughout its journey through the core, a simpler thermodynamic cycle can be used (since there is no need for evaporators or condensers), lowering the capital cost of the reactor [3]. Being in a super critical state, the coolant is unable boil along the fuel channels, which means that dryout (a prominent safety concern in most nuclear reactors) cannot occur as long as operational coolant pressures are maintained. The near elimination of dryout as a concern is a major safety advantage of the SCWR design that sets it apart from the majority of operating nuclear reactors. Finally, the use of light water coolant instead of the heavy water coolant (which is worth roughly $100,000 a barrel) used by the CANDU6, helps to reduce the upfront cost of the SCWR reactor relative to CANDU6.

In the CANDU, pressure tubes are used in the SCWR design to keep the high pressure coolant separate from the low pressure moderator. This is in contrast to most LWR designs (light water reactors), which use a single high pressure reactor vessel to contain the coolant and the moderator. The use of pressure tubes greatly lessens the volume of water that needs to be maintained at a high pressure, eliminating the need for pressure vessels that are difficult to manufacture (reducing the cost of the design). It also allows moderator to be different from the coolant, which is a requirement of the SCWR design. Unlike a CANDU, the pressure tubes in a SCWR reactor are vertically aligned. This allows natural convection to assist the coolant flow through the core and fixes the flow direction during an accident scenario. As expected from a Canadian design, the SCWR uses heavy water as moderator, which costs significantly more than light water, but should allow the SCWR to achieve better burnup (fuel economy) then similar LWRs. The use of a heavy water moderator should also significantly increase the reactor time period (the time it takes for changes in the reactor to affect the neutron population), making the reactor inherently more stable, and easier to control than an LWR. The current SCWR concept does not allow for online refueling, instead it uses batch refueling with 1/3 of the core being replaced every time it is refueled.

Another key aspect of the Canadian SCWR design is the use of MOX fuel made up of plutonium and thorium. The main advantages of using thorium are twofold: it is completely safe to handle before entering the core and it is more abundant than uranium, greatly extending the potential life span of nuclear power based on our current reserves [4]. The advantage of using plutonium over uranium is the potential to utilize plutonium

from the spent fuel of CANDU, LWRs, or from nuclear weapon reserves. The use of plutonium extracted from spent fuel would allow for better fuel economy and less expensive storage of the final waste (due to the lower potential energy left in the fuel). Burning the plutonium present in nuclear weapon reserves reduces the threat of the proliferation of weapons grade plutonium.

Since the SCWR design is very different from the current Canadian reactor fleet, determining its safety and performance will rely heavily on accurate computational modes. This is outlined by the following quote from CNL.

"The SCWR, as a synthesis of HWR, BWR, SCW-FFP and new innovations, presents challenges to current physics analysis and modeling methods that have evolved to suit the conditions that typify conventional reactor systems. Since there exists little or no experimental data available to test the application of current physics methods to SCWR operating conditions, modeling errors and uncertainties must be estimated via comparisons among various codes and methods. Such comparisons will aid in the development of these methods to provide supporting calculations for the SCWR concept." CNL SCWR report [2].

Some of the key parameters of the SCWR design can be seen below in Table 1.1.

| | |
|---|---|
| Thermal Power | 2540 MW |
| Electric Power | 1200 MW |
| Inlet / Outlet temperatures | 350ºC / 625ºC |
| Inlet / Outlet pressures | 26 MPa / 25 MPa |
| Channels | 336 |
| Lattice Pitch | 25 cm |
| Core Radius | 355 cm |
| Core Height | 650 cm |
| Upper axial reflector thickness | 75 cm |
| Lower axial reflector thickness | 75 cm |
| Fuel Assembly Length | 500 cm |
| Fuel batches | 3 |
| Target exit burnup | 40 MWd / kg |

| | |
|---|---|
| Target CVR | < 0 |

*Table 1.1: Key characteristics of the SCWR design*

## 1.3 Objectives

Since G4STORK (Geant4 STocastic Reactor Kinetics) is a Monte Carlo code that uses minimal assumptions when determining the neutron distribution in a geometry, it was an ideal candidate for being part of the SCWR simulation comparison study [1]. The part of the SCWR geometry that CNL wanted to be investigated was a single fuel assembly in the reactor divided into 10 axial segments [2]. Each axial segment was treated as a lattice cell (by applying the periodic boundary condition at all boundaries), with temperatures and densities being assumed to be homogenous within the same materials. Since the fuel cells are rotationally symmetric around the z axis a single quarter of the geometry could be used to represent the whole fuel cell. This effectively increased the neutron density in the geometry four fold for the same computation time, improving the accuracy of the results. The simulated reactor geometry can be seen below in Figure 3.1.

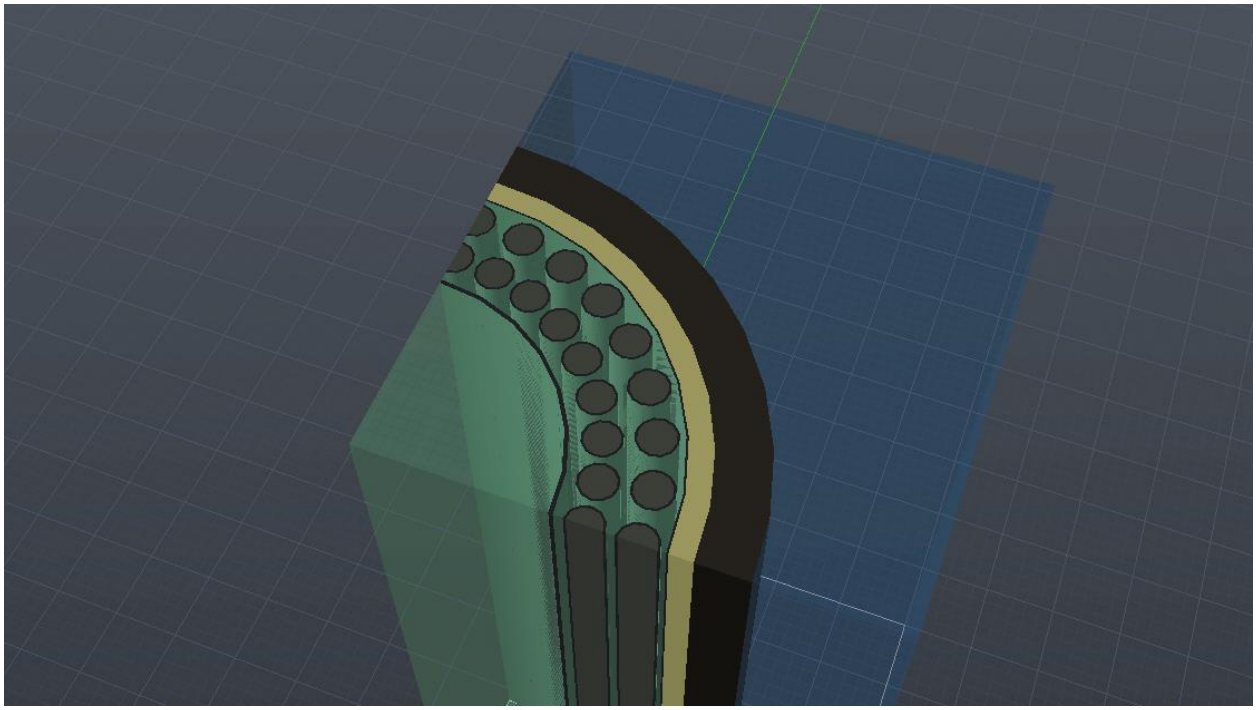

*Figure 1.1: Shows an axonometric view of the SCWR quarter lattice cell geometry*

In the comparison, both short duration reactivity calculations and long term fuel depletion calculations were asked for; however, at present G4STORK cannot perform depletion calculations, so only reactivity calculations were executed using G4STORK. The four reactivity cases that were under investigation were the fully cooled case, the voiding of the inner coolant, the voiding of the outer coolant, and the voiding of all the coolant, all of

which used fresh fuel at full power. The reason why these cases were of particular interest is that one of the major design requirement of the SCWR concept was to have a negative coolant void reactivity (CVR); the neutron population should decrease in the event that there is a loss of coolant. This is meant to improve the safety of the reactor, but more importantly would allow the design to sell in USA, (probably the primary target customers for this design given the use of enriched plutonium in the fuel) where a negative CVR is required by the local nuclear reactor regulators.

The initial results produced by G4STORK were significantly different from the other codes used in the comparison [5]. Given that different codes use of different datasets and different assumptions about the physics involved, some differences were expected. The other codes involved in the comparison had already been benchmarked against each other in a large international benchmark, so the fact that they agreed within the bounds that they did was not surprising [6]. However, the results produced using G4STORK were clearly the outsider of the comparison, as can be seen in Table 1.2. This suggested that there were flaws with G4STORK, flaws shared by the other software in the comparison, or both. Since the primary purpose of the code comparison was to estimate the margin of uncertainty for the reactor design, the reason behind what was causing this particular discrepancy was important. To identify the cause of the discrepancy, a detailed comparison of the neutron physics (more importantly the algorithms) used by G4STORK, and those of another code in the comparison, had to be undertaken. MCNP6 was chosen for this task due to its similar methodology and its international reputation in the field of nuclear physics. This comparison would not only serve to improve the understanding and assist the advancement of the SCWR design, but would also provide a great opportunity to validate a relatively young piece of software against a trusted veteran.

| $K_{inf}$ Cases | Fully Cooled | Void Inner Coolant | Void Outer Coolant | Void All Coolant |
|---|---|---|---|---|
| G4STORK | 1.253±0.0006 | 1.206±0.0002 | 1.258±0.0006 | 1.215±0.0004 |
| MCNP6.1 | 1.2863±0.0001 | 1.2497±0.0003 | 1.2984±0.0003 | 1.2666±0.0002 |
| DrIAEA | 1.27559 | 1.247399 | 1.284174 | 1.259823 |
| Dr7.1 | 1.28387 | 1.254724 | 1.292628 | 1.267557 |
| KENO238 | 1.28629 | 1.255601 | 1.296471 | 1.269283 |
| KENOCE | 1.28735 | 1.25917 | 1.296439 | 1.27279 |
| TRITON | 1.286854 | 1.25781 | 1.29675 | 1.268812 |

*Table 1.2: Shows the $K_{inf}$ for the SCWR lattice cell at 1.75m from the bottom of the core for four reactivity cases. Ask Jason for raw data, (MCNP data is our own data produced using Michel's geometry) [5].*

# 2 Background and Theory

In this section of the report, we will discuss the fundamentals of nuclear reactor theory, and advanced topics that are relevant to the physics being examined in this thesis. In particular, we will cover how nuclear power works, what are the neutron involved interactions, how we measure reactor behavior, how we model it using computational methods, and how we analyzed the results.

## 2.1 Liquid Drop Model

It is possible for an object in a high energy state (a ball on top of a skyscraper) to transition to a lower energy state (ball falls to the ground). However, according to quantum mechanics, it is far less probable for an object to transition to a higher energy of its own accord (ball jumps into the air). Thus all things (including nuclei) move towards, the lowest energy state that they can exist in. As a result of the extremely attractive strong force, nucleons can exist in a lower energy state when bound together then they can apart [7]. The difference in energy between a group of nucleons freely roaming and the equivalent nucleons in a bound configuration is referred to as the binding energy. The higher the binding energy (the lower the energy state of the nucleus is relative to free particles), the more stable the nucleus becomes, because more energy would be required to separate the particles. The liquid drop model is a semi-empirical model that predicts the binding energy of a nucleus for different nucleon configurations. The equation describing the liquid drop model is shown below in Equation 2.1.

$$Eb = a_V \times A - a_S \times A^{\frac{2}{3}} - a_C \times \frac{Z^2}{A^{\frac{1}{3}}} - a_A \times \frac{(A-2Z)^2}{A} + \delta(A,Z)$$

*Equation 2.1: Shows the relation between the binding energy ($E_b$), the total number of nucleons (A), and the number of protons (Z)*

The first term in Equation 2.1 is called the volume term, and it accounts for the increase in binding energy as the number of nucleons bound together increases. The second term is the surface term; it accounts for the nucleons at the surface having fewer bonds and thus a lower binding energy (this is why most nuclei are spherical to maximize the volume to surface area ratio). The third term is the coulomb term; it accounts for the loss of binding energy due to the electric field repulsion of the protons. The fourth term is the asymmetry term; it accounts for the loss of binding energy as result of having a non-ideal (1:1) ratio of neutrons and protons (this is a quantum mechanical effect). The final term is the parity term and it accounts for the gain or loss of binding energy by having even or odd numbers of neutron and protons (this is a quantum mechanics effect). Minimizing the negative effects of the last three terms, the dependence of the binding energy on the number of nucleons can be seen below in Figure 2.1.

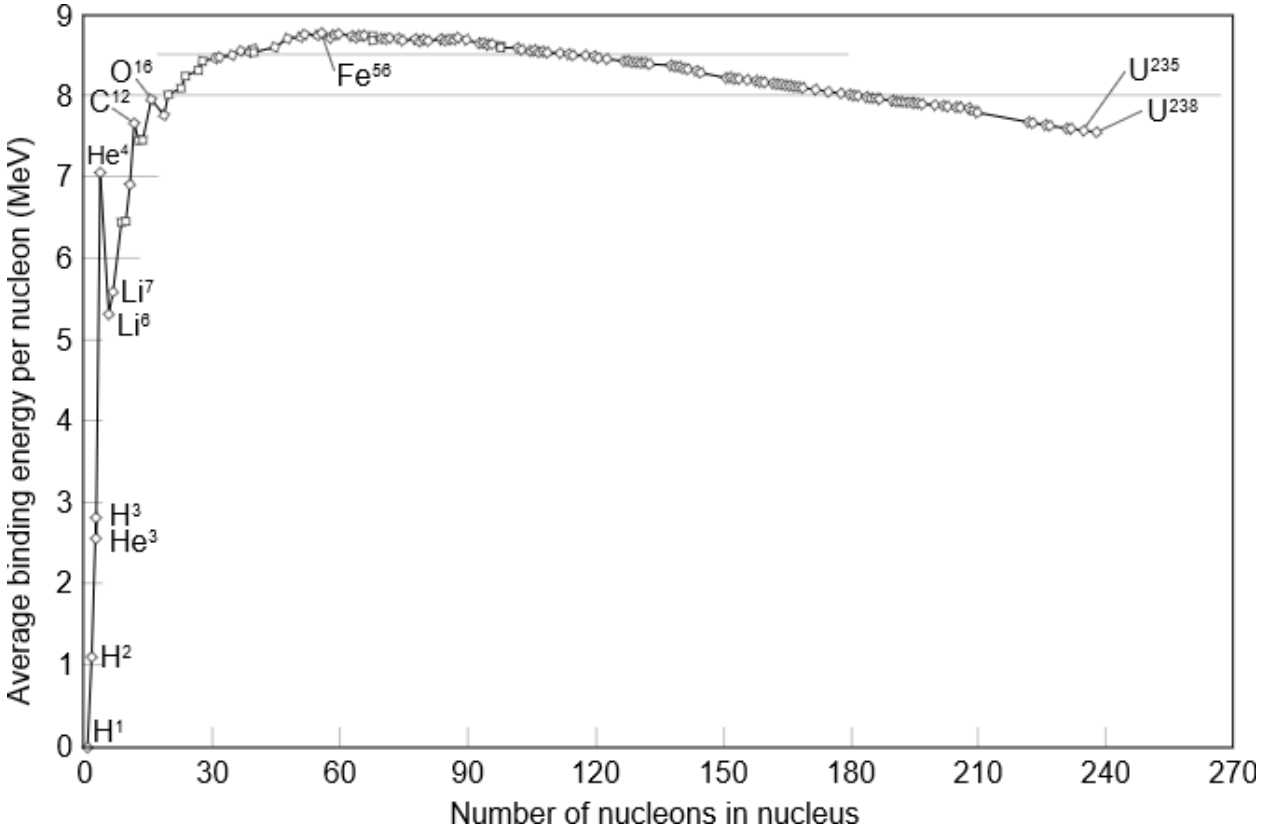


*Figure 2.1: shows the binding energy per number of nucleons as a function of the number of nucleons when the last three terms of the liquid drop model are minimized [8].*

We can see that the maximum binding energy occurs at iron-56, and that isotopes that are much heavier or lighter than iron-56, have significantly less binding energy (require more stored energy to hold them together). Thus by breaking apart (fissioning) a heavy isotope (such as uranium-235) we would create two nuclei that have significantly higher total binding energy than the original isotope. This means that during the transition there would be a release of potential energy that was stored in the original isotope. It is this released potential energy that we take advantage of in a nuclear reactor to produce electricity.

## 2.2 Cross-section

Consider a beam of neutrons traveling in the same direction through a material made of a single nuclide. It would be natural to assume that the rate at which the incident neutrons will undergo a reaction is dependent on the flow rate (number density multiplied by average speed) of the neutrons through the material, and the probability of a neutron undergoing a reaction (which we expect to be dependent on the energy of the incident neutron) per distance traversed in the material [7].

$$\iiint R(E,\vec{r})\,dV = \Sigma(E) \times \iiint J(\vec{r})\,dV \rightarrow R(E,\vec{r}) = \Sigma(E) \times J(\vec{r})$$

*Equation 2.2: Shows the relation between the reaction rate density ($R(E,\vec{r})$), neutron flow rate density (J), and the probability of interaction per distance ($\Sigma$ (E)). The position is described by $\vec{r}$ =(x,y,z), and the energy of the incoming neutron is set by E.*

If we allow the neutrons to flow in all directions, the J term in Equation 2.2 above becomes the neutron flux. By realizing that Σ (E) will depend on the number density of atoms (N) in the material the neutrons are transversing we get.

$$R(E,\vec{r}) = \Sigma(E) \times \emptyset(\vec{r}) = N \times \sigma(E) \times \emptyset(\vec{r})$$

*Equation 2.3: Shows the relation between the reaction rate density ($R(E,\vec{r})$), neutron flux ($\emptyset(\vec{r})$), macroscopic cross-section ($\Sigma$ (E)), number density of atoms (N), and the microscopic cross-section ($\sigma(E)$).*

Σ (E) is commonly referred to as the macroscopic cross-section, and $\sigma(E)$ is commonly refered to as the microscopic cross-section. Since the microscopic cross-section is independent of the density of the material, which can fluctuate with temperature and pressure, it is the most commonly used cross-section. For the sake of this report, mentions of 'the cross-section' will refer to the microscopic cross-section, unless it is explicitly stated otherwise. The microscopic cross-section has units of $cm^2$ (more commonly quoted in barns $10^{-24}$ $cm^2$), and is officially defined as "the effective cross-sectional area per nucleus seen by the neutron". The microscopic cross-section is a function of the kinetic energy of the incoming neutron relative to the nucleus.

### 2.2.1 Mean free path

The mean free path is the average distance that a neutron with a particular energy will go before undergoing an interaction in a uniform and infinite material. It can be derived using the classic integrated average equation [7].

$$\overline{x(E)} = \int_0^\infty x \times dP(E,x) = \int_0^\infty x \times \frac{d\mathrm{P}(E,x)}{dx} \times dx = \frac{\int_0^\infty x \times N \times \sigma(E) \times \emptyset(E,x) \times dx}{\emptyset_0}$$

*Equation 2.4: Shows the relation between the mean free path ( $\overline{x(E)}$), the distance traveled (r), the atomic density (N), the microscopic cross-section ($\sigma(E)$ ), the uncollided flux ( $\emptyset(E,\vec{x})$), and the starting uncollided flux ($\emptyset_0$).*

The uncollided flux can be derived in this equation by realizing that the reaction rate described in Equation 2.3 will be equal to the negative derivative of the uncollided population.

$$\frac{d\emptyset(E,x)}{dx} = -N \times \sigma(E) \times \emptyset(E,x) \rightarrow \emptyset(E,x) = \emptyset_0 e^{-N \times \sigma(E) \times x}$$

*Equation 2.5: Shows the relation between the uncollided flux ( $\emptyset(E,\vec{x})$), the distance traveled (r), the atomic density (N), and the microscopic cross-section ($\sigma(E)$).*

Substituting Equation 2.5 into Equation 2.4, we get that the mean free path is equal to the following.

$$\overline{x(E)} = \int_0^\infty x \times N \times \sigma(E) \times e^{-N \times \sigma(E) \times x} \times dx = \frac{1}{N \times \sigma(E)} = \frac{1}{\Sigma(E)}$$

*Equation 2.6: shows the relationship between the mean free path ($\overline{x(E)}$), the atomic density (N), and the microscopic cross-section ($\sigma(E)$).*

### 2.2.2 Resonance

When a neutron is absorbed by an atom, it combines with it to create an excited state. Quantum mechanics tells us that these excited state only occur at fixed energies (which are dependent on the atom), and thus the only way these excited states can be formed is by incoming neutrons with fixed energies (with a little bit of wiggle room due to the probabilistic nature of quantum mechanics) [7]. Thus the cross-section (probability) of an absorption reaction will sharply increase and decrease around the neutron energies needed to form an excited state, creating what is known as resonant peaks. As the nucleus gets larger there are more available spaces for the incoming neutron to fill and thus there are more resonant peaks in the cross-section. Likewise, as the excitation energy of the atom increases, the number of available excitation energy states increase, and so does the frequency of cross-sectional resonant peaks. Eventually the peaks become so close together that they cannot be told apart (due to the limitations of current measurement equipment), and the cross-section is just measured as a flattened line. These effects can be seen below in Figure 2.2.

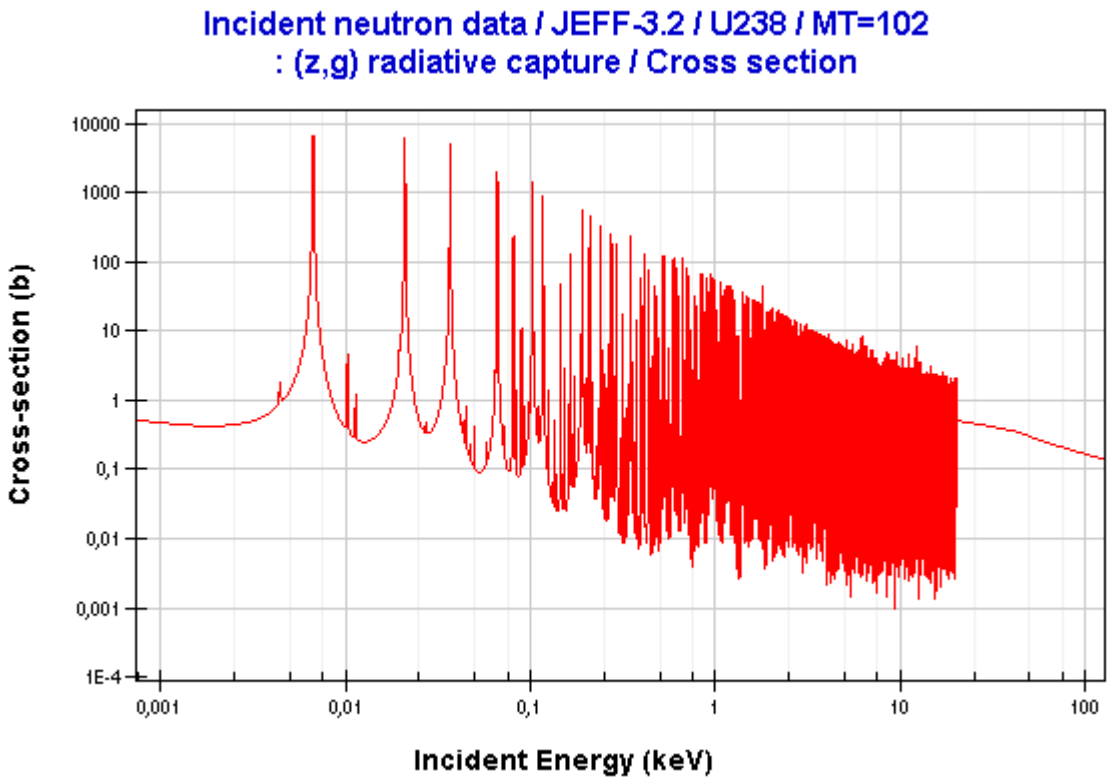


*Figure 2.2: Shows the radiative capture cross-section data for uranium-238 to illustrate the dependence of resonance on incoming neutron energy*

### 2.2.3 Doppler Broadening

Temperature is a macroscopic measure of the average kinetic energy of particles within a material; the larger the temperature, the stronger the vibrations of the particles [1]. In a free gas, the speed of the atoms relative to the lab frame can be described using the Maxwell-Boltzmann distribution, shown below.

$$f(E) = \sqrt{\frac{m}{2kT}} e^{-\frac{E}{kT}}$$

*Equation 2.7: The Maxwell Boltzmann distribution, m=mass, E=kinetic energy, k is the Boltzmann constant, T=temperature, and f is probability [7]*

For solid and liquid materials, the Maxwell-Boltzmann distribution is often used as an approximate method of determining the movement of the atoms. Thus when a neutron is traveling towards a material of high temperature, the incoming velocity of the neutron relative to the nucleus randomly varies around the value of the neutron's velocity in the lab frame with the addition of the velocity of the nucleus. Since the cross-section is a function of the incoming kinetic energy, the blurring of the incoming kinetic energy

causes the cross-section to become blurred (a random sampling of those around it) as well. This effect primarily affects resonant peaks, causing them to become flatter and wider. This is why the effect is known as Doppler broadening, Doppler because this effect is caused by the relative movement of the source (neutron) and the observer (nucleus), and broadening because it has the effect of broadening (flattening) peaks in the cross-section. A visualization of what the cross-section data looks like after being broadened can be seen below in Figure 2.3.

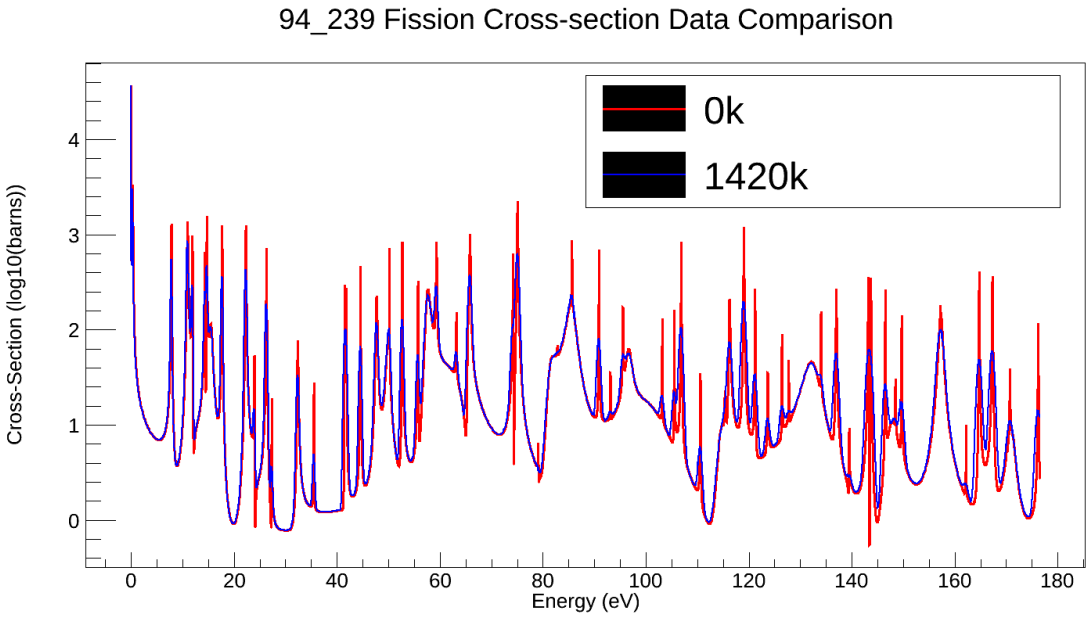


*Figure 2.3: Cross-section data files of 94-239 at 0k and at 1420k for the fission process.*

As the resonant peaks become smaller, self-shielding becomes less and less important, which mean that Doppler broadening causes the probability of absorption (particularly neutron capture) to increase. As the temperature of the fuel goes up, neutron absorption increases as well, causing the reactor power to decrease, and the fuel temperature to decline. This negative temperature feedback mechanism is an important safety component of any nuclear reactor.

## 2.3 Neutron Interactions

There are four interactions that neutrons can undergo with a nucleus while transiting a geometry: elastic, inelastic (here we are including all non-fission, and non-elastic neutron producing reactions), fission, and neutron capture (here we are including any reaction with an incident neutron that does not produce neutrons) [9]. The balancing of these four interactions (particularly fission and capture) is what allows a nuclear reactor to sustain a

steady neutron population and produce power. It is possible to derive exact theoretical descriptions for each of these interactions using the principals of quantum mechanics.

However, as the size of the atom becomes larger, the difficulty of finding a solution increases rapidly. Eventually finding a solution becomes impossible using current computing technology. Instead scientists develop distributions to match empirical data collected from high precision experiments and use these to describe the more complex interactions. This means that there are many different distributions that are meant to model the same interaction and the burden is left on the software developer to select which one better meets their needs. This usually involves compromising between speed and accuracy. While determining which algorithm is faster is usually obvious and easy work, determining which distribution represents the empirical data the best is much more difficult, and is dependent on the situation that the data will be used in (the energy range of the neutrons, which materials are being used, and at what temperatures).

### 2.3.1 Elastic Scattering

In this case both the momentum and the kinetic energy of the particles involved are conserved by the reaction, as a result no new particles are created. This often results in the neutron losing kinetic energy and changing direction but it can result in the neutron gaining kinetic energy (this commonly occurs when a high temperature material is being transverse by thermal neutrons). Elastic scattering can occur at any incoming neutron kinetic energy since the nucleus does not need to absorb the neutron and form a compound state and thus there is no threshold energy.

### 2.3.2 Inelastic Scattering

In the case of inelastic scattering the momentum of the system is conserved but the kinetic energy is not. The incoming neutron is temporarily absorbed by the nucleus, creating an unstable compound state [7]. A neutron then exits the nucleus, with less kinetic energy than the incoming neutron, putting the nucleus in an excited state. The nucleus then de-excites by emitting photons with energy equal to the difference of the initial neutron and the out-going neutron.

There are distinct incoming neutrons energies needed to create an excited state. These energies are blurred by the vibration of the target nucleus relative to the neutron (as described here 2.2.3). The probability of this interaction occurring is dependent on the number of excited states available for the incoming neutron to create with the target nucleus. From our earlier discussion about resonance we know that the number of excited states generally increases with the size of the nucleus and with incoming neutron kinetic energy, which means that the probability of the inelastic scattering occurring also increases with these two parameters. Subsequently we can also determine that the lower threshold energy at which this reaction will occur is slightly greater than the energy required by the lowest excited energy state of the nucleus (lower for larger atoms).

However, there are more factors to consider then just the number of excited states, when determining the probability of an interaction occurring.

In the field on of computational nuclear physics it is common to specify any reaction that produces neutrons, and is not fission or elastic, as inelastic scattering. Even though many of the included reaction do not exactly match the above description exactly, they have similar enough properties and effects that it is simpler to treat them this way.

### 2.3.3 Neutron Capture

Neutron capture is very similar to inelastic scattering, (an incoming neutron is absorbed by a target nucleus, the target nucleus de-excites by emitting particles) except that neutron capture does not emit any neutrons. In fact, it is the only interaction which does not emit a neutron and is thus vital for controlling the chain reaction of neutrons. Due to the similarity of neutron capture to inelastic scattering, the actual separation point is somewhat ambiguous in nuclear literature (the G4NDL library classifies many interactions that do not emit neutrons as inelastic for instance) but for the purpose of this report the above definition is what will be used.

### 2.3.4 Fission

As described in the previous section about the 2.1, as nucleus becomes larger it has a harder time holding itself together due to the increasing internal columbic repulsive force. At some point, either due to the random movement of the neutrons and protons within the nucleus, or because of the absorption of an incoming particle (in our case a neutron) the nucleus is spatially deformed to such a state that the coulombic force over whelms the strong force and the nucleus splits apart. This causes two smaller nuclei to be produced along with multiple prompt (immediately released) neutrons, photons, and neutrinos. The bulk of the energy is in the fission fragment kinetic energy (which usually deposit their energy fairly close to the site of the fission), heating up the fuel, and allowing us to harvest the energy using a thermodynamic cycle. Unlike inelastic scattering and neutron capture, fission can occur spontaneously without an incoming neutron. While most isotopes can undergo inelastic scattering and neutron capture, fission is only possible in very large and unstable nuclei. Isotopes capable of undergoing fission are called fissionable while isotopes capable of undergoing fission from neutrons at thermal energies (the vast majority of neutrons in a thermal reactor) are called fissile.

## 2.4 Criticality

Each reactor has a maximum power limit at which it can safely transfer heat from the fuel to the coolant. If a reactor is allowed to go far beyond this point it can lead to severe damage to the core, and potentially reactor meltdown. The amount of power the reactor produces is also directly linked to the amount of revenue it brings in, so that there is also a minimum power level that it must operate at in order for the operation to be economical. The power of a reactor is primarily dependent on the rate at which fission is occurring

within it, which we can see from Equation 2.3 is linearly dependent on the neutron flux (assuming the energy and spatial distribution of the flux don't change). Thus both the safety and performance of any nuclear reactor is highly dependent on controlling the neutron population. Because of this, many important measures have been devised to describe how the neutron population is changing and at what rate. One of the most important parameters for measuring the rate at which the neutron population is increasing or decreasing is K*eff* (described in Equation 2.8) [7].

$$K_{eff} = \frac{\dot{N}_{production}}{\dot{N}_{absorption}}, \quad N = neutron\ population$$

*Equation 2.8: Shows the relationship between $K_{eff}$, the rate of neutron production, and the rate of neutron absorption.*

The neutron multiplication factor (also called the $K_{eff}$) is used to determine the degree to which the neutron population is growing ($K_{eff}>1$, supercritical), declining ($K_{eff}<1$, subcritical), or is stable ($K_{eff}=1$, critical). If the average neutron life time is known, it can be used to determine the exact ratio of the current population compared to a previous population ($K_{run}$) over a given time step, using Equation 2.9.

$$K_{run} = \frac{\int_{t0}^{t1} K_{eff}(t) \times dt}{T_{life}}$$

*Equation 2.9: shows the relationship between $K_{run}$, $K_{eff}$, time (t), and the average neutron life time ($T_{life}$).*

Another parameter that is related to $K_{eff}$ is reactivity (ρ), described by Equation 2.10 below.

$$\rho = \frac{K_{eff} - 1}{K_{eff}}$$

*Equation 2.10: shows the relationship between reactivity (ρ), and $K_{eff}$*

Reactivity is used to describe how offset the system is from a critical state; the reactor is supercritical when ρ >0, subcritical when ρ <0, and critical when ρ=0.

## 2.5 Delayed neutrons

The fission fragments are born in an excited state and will eventually decay to produce delayed (because they are created hundreds of milliseconds to minutes after the fission process has occurred) neutrons and photons [7]. The prompt and delayed neutrons then go on to induce more fissions elsewhere, continuing the chain reaction. Although delayed neutrons only make up a small percentage of the yield (0.64% in uranium-235) they are essential for controlling the chain reaction. Any changes in the reactor causing variations in the probability of neutrons inducing fission will be reflected in the next generation of neutrons produced by fission (daughter neutrons). Starting from birth, prompt neutrons

only take microseconds (tens of microseconds in CANDU) to be absorbed somewhere else in the core (inducing fission or neutron capture). Thus, if the neutron population was only made up of prompt neutrons, the neutron population would be multiplied by $K_{eff}$ every few microseconds (the average fission generation time of the neutrons), making the neutron population impossible to control. However, if the delayed neutrons make up ~0.6% of the neutron population then the average neutron generation time = 0.994*(~$10^{-6}$s) + 0.006*(~1.0s) = .006s and thus neutron population will be multiplied by $K_{eff}$ at a much more manageable frequency. If $K_{eff}$ increases beyond the point where the prompt neutrons are able to sustain their own population (this point is called prompt critical), the number of prompt neutrons will be multiplied every few microseconds causing the delayed neutrons to become a very insignificant percentage of the total neutron population, and the average neutron generation time to drop to the prompt neutron generation time.

## 2.6 Computational Nuclear Physics

In this section of the report, we will cover computational nuclear physics theory relevant to this thesis. In particular, we will cover methods of calculating criticality, on the fly Doppler broadening, Shannon Entropy, the ENDF datasets, the high precision neutron interaction model (NeutronHP), MCNP6.1, and G4STORK.

### 2.6.1 Methods of Calculating Criticality

As we can see from Equation 2.8, $K_{eff}$ is the ratio of the rate of neutron production to the rate of neutron loss.

#### *2.6.1.1 Dynamic Criticality Method*

The dynamic criticality method works by measuring the ratio of the average rate of neutron production to the average rate of neutron loss over a step in time (as is shown below in Equation 2.11) [1]. Thus, the dynamic criticality method assumes that the $K_{eff}$ is constant over the time step, which is true if the reactor geometry does not change significantly during the time step. The shorter the time interval the more accurate this approach becomes. However, the shorter the time interval, the larger the neutron flux will have to be, to ensure that enough neutron productions and losses will happen within that time interval, for $K_{eff}$ to be statistically significant.

$$K_{dyn} = \frac{N_{production}(T1)}{N_{absorption}(T1)} = \frac{\dot{N}_{production}}{\dot{N}_{absorption}} \times \frac{\Delta T}{\Delta T}$$

*Equation 2.11: shows the dynamic criticality method that G4STORK uses to calculate $K_{eff}$*

#### *2.6.1.2 Generational Criticality Method*

The generational criticality method approximates $K_{eff}$ by taking the ratio of the number of neutrons produced at the end of one fission generation to the number of neutrons present at the start of the neutron fission generation (this is shown below in Equation 2.12) [1].

$$K_{gen} = \frac{Neutron\ population\ in\ generation\ i+1}{Neutron\ population\ in\ generation\ i}$$

*Equation 2.12: shows the generational criticality method that MCNP uses to calculate $K_{eff}$*

The neutron lifetime can vary greatly within a reactor. So pretending that the counting of surviving neutrons after a fission generation is equivalent to counting of surviving neutrons in steps of time, is only accurate when the neutron population is nearly independent of time ($K_{eff}$~1). One can easily prove this, by pretending that the neutron population within a reactor is split into two energy groups. We expect each energy group to have its own neutron production rate, loss rate, and average life time within the reactor. By subbing these criteria into Equation 2.12, we get Equation 2.13 shown below.

$$K_{gen} = \frac{(P_1 \times T_1 + P_2 \times T_2)}{(L_1 \times T_1 + L_2 \times T_2)}$$

*Equation 2.13: shows the generational criticality method when there are only two neutron energy groups in the reactor. Pi is rate of neutron production for group i, Li is rate of neutron loss for group i, and Pi is the average neutron lifetime for group i.*

The only ways that Equation 2.13 can be equivalent to Equation 2.11 are if $T_1$=$T_2$ or if $P_1/L_1$=$P_2/L_2$. Thus we can determine that in order for $K_{gen}$ to be equivalent to $K_{dyn}$, all the different neutron groups (we could have divided the neutrons based off position as well) would either have to have the same average life time (this does not occur in nature), or have the same neutron multiplicity (which can only occur when $K_{eff}$=1 if the average neutron life times are different). By closely studying Equation 2.13, one will notice that the importance of the neutron multiplicity for each neutron group will be weighted by their average lifetime. The neutrons groups that take a longer time to be absorbed (typically the thermal neutrons) will influence $K_{eff}$ more than they should. This makes the generational criticality method a far less suitable candidate then the dynamic criticality method, for simulating very sub or supercritical cases (such as reactor shutdown or startup). An illustration of the difference between the dynamic criticality method and the generational criticality method can be seen below in Figure 2.4. The generational criticality method is often used in reactor kinetic codes because it allows the calculation of $K_{eff}$ without the need for keeping track of the time the neutrons have spent in the reactor.

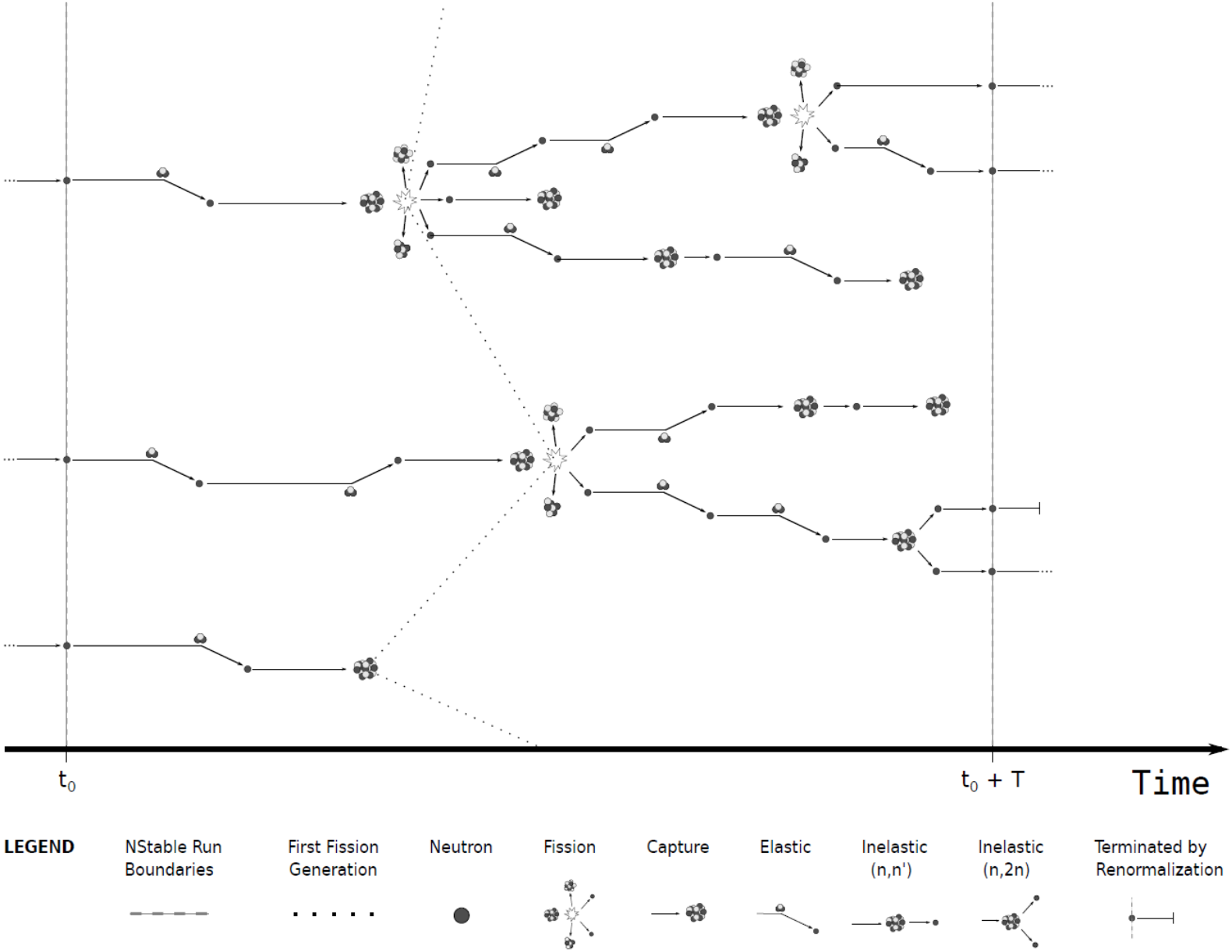
$t_0$
$t_0 + T$
Time
LEGEND
NStable Run Boundaries
First Fission Generation
Neutron
Fission
Capture
Elastic
Inelastic (n,n')
Inelastic (n,2n)
Terminated by Renormalization

*Figure 2.4: Illustrates the difference between the dynamic criticality and generational criticality method [1]*

### 2.6.2 On the Fly Doppler Broadening

From our discussion of Doppler broadening (here, 2.2.3), we know that the temperature of the material has a significant effect on the cross-section of the isotopes. To take this into account, many codes including G4STORK use an on the fly Doppler broadening algorithm, which works as follows [1]. The velocity of the nuclei relative to the material is determined by randomly sampling over a Gaussian distribution, centered on zero, with a width that is proportional to the temperature of the material (this is equivalent to using a Maxwell-Boltzmann distribution). Then, the kinetic energy of the neutron relative to a nucleus is determined, the corresponding cross-section is located and corrected for the relative speed of the neutron (since neutrons travelling faster relative to the nuclei will encounter more nuclei and thus have a higher chance of interacting). In order for the sampled nuclei velocity to be representative of all the isotopes in the material, the nuclei velocity is sampled many times until the variance in the resulting cross-sections is within a small percentage of the mean value. This is a very time-consuming process which is exacerbated if the temperatures used in the user defined reactor geometry are high (since more samples will have to be taken to cover the wider Gaussian distribution).

### 2.6.3 Shannon Entropy

Shannon entropy is a measure of the minimum amount of information (number of symbols) needed to express each unique element in a given data set, on average [10]. The equation used to calculate Shannon entropy is shown below in Equation 2.14.

$$H = -\sum_{i=1}^{N} P_i \times \log_b P_i$$

*Equation 2.14: Shows the relationship between the Shannon entropy (H), the number of unique elements in the dataset (N), the abundance of element i in the dataset ($P_i$), and the number of unique symbols that can be used to represent the element (b) (ex: b=2 if a binary number is being used)*

Shannon entropy is most commonly used by computer scientists to quantitatively assess the performance of lossless data compression algorithms. However, the datasets used for Shannon entropy calculations, do not have to be limited to characters and numbers. For instance, let us assume that each element in a dataset represents a unique region space, and that $P_i$ is the probability of a neutron being present in region 'i'. When we take the Shannon entropy of this data set, we are measuring how spread out the neutron population is in the geometry. Thus, if the spatial distribution of the neutrons has converged (i.e. not shifting between run iterations), then the Shannon entropy will be constant between iterations. By assuming that the previous statement is true in reverse (which is nearly always the case), the spatial convergence of the neutron population can be said to occur when the Shannon entropy is nearly constant between run iterations.

### 2.6.4 Evaluated Nuclear Data Files

The evaluated nuclear data files (ENDF) contain experimental data describing the properties of selected neutron interactions. ENDF was created by a group of U.S and Canadian nuclear laboratories, called the Cross-section Evaluation Working Group (CSEWG) in the 1960s [11]. The purpose of ENDF was to supply a flexible formatting standard to be adopted by the many preexisting nuclear sets. This would allow for easier comparisons between nuclear datasets, and better simulation reproducibility. It also made it easier for nuclear reactor physics programmers to create an interface capable of accepting many different nuclear data sets. Most importantly it allowed laboratories to pool their experimental data, and come up with a much more complete description of the physics involved. The raw ENDF data has far more information than is wanted by most nuclear reactor physics simulations. So instead of using the ENDF data directly, preprocessing codes are used to thin out the ENDF data to a given accuracy, and put it in a more useful format for the given nuclear reactor physics simulation.

### 2.6.5 The High Precision Neutron Interaction Model

The high precision neutron interaction model (called the NeutronHP model in GEANT4, and will be referred to as such for the rest of the thesis) is built off the ENDF data, and data libraries like it. By interpolating empirical data tables or sampling distributions fitted to laboratory measurements, the NeutronHP model is able to accurately reproduce experimental results, over the $10^{-5}$eV to 20MeV energy regime. Since neutron energies within a nuclear reactor typically fit well within this energy range, it can be accurately assumed that the NeutronHP model contains all of the physics relevant to nuclear reactor research. As discussed in the previous section (2.6.4), before the ENDF data can used in a nuclear reactor physics program, it must be converted into a neutron interaction library. During this preprocessing of the data, different representations (types of distributions) of the same raw data, are often selected by different programs. Even when the same type of representation is chosen, differing formats and levels of data precision are often used between libraries. This leads to each nuclear reactor physics program having a unique neutron interaction library. Subsequently, since the NeutronHP models are built around these libraries, they are also unique for each reactor physics program.

### 2.6.6 Thermal Scattering

In the previous section when we talked about neutron interactions (here, Neutron Interactions), we neglected to talk about the effect the surrounding atoms have on the nucleus undergoing the process. We did this because at incoming neutron energies above 4eV, the bonding effects of the surrounding atoms becomes negligible [12]. However, for energies less than 4eV, the effect of the intramolecular bonds on the interaction cross-section, and the outgoing particle energy and angular distribution becomes significant. In nuclear reactor physics we only really care about how the out-going neutrons are affected, so we ignore the intramolecular effects for neutron capture (since no neutrons are produced). After fission has occurred, the produced neutrons have kinetic energies of

~2MeV. Thus, the effects of the intermolecular bonds on the outgoing fission neutron energy and angular distribution are insignificant. So, we are only really interested in the intramolecular effects on the elastic and inelastic scattering interactions. In order to take these effects into account we use what is called a thermal scattering model (thermal because it is only used for thermal incoming neutrons, and scattering because it only affects the scattering interactions). Laboratory measurements of the outcome of scattering interactions occurring, for many different target isotope and material combinations, are stored in libraries as the functions of energy transferred and momentum transferred. During a nuclear reactor physics simulation, these data sets are sampled by the thermal scattering model to determine the result of a scattering interaction.

### 2.6.7 MCNP6.1

#### *2.6.7.1 Summary*

MCNP (Monte Carlo N-Particle transport code) is a widely used and respected code in the field of nuclear reactor physics. Since its creation in 1977, from the merger of MCNG and MCP [13], MCNP has become a stable, efficient, and feature rich code. Like most particle physics codes, the user specifies what they want MCNP to simulate by creating an ASCII text file in a format specified by MCNP. This input file contains everything about the simulation, the geometry, which physics to use, which libraries to use, and what results to store. At the start of an MCNP simulation, the code reads in the selected input file and uses it to gather the necessary data and set up the simulation. At the beginning of a criticality simulation, MCNP creates neutrons at positions in the fuel specified by the user with energies sampled from a generic Watt thermal fission distribution by default. Each neutron is then followed through the geometry until it has been absorbed (not including absorptions due to inelastic scattering). Once all the starting neutrons have been absorbed, the next generation of neutrons is created from the fission daughter neutrons, and the cycle is repeated. After every generation the $K_{\text{eff}}$ is calculated using the generational criticality method described here 2.6.1.2.

Every time a neutron enters a new material, the total cross-section (the summed cross-section of any relevant neutron interaction occurring) is looked up and used to sample the distance at which an interaction will occur. If this distance is less than the distance the particle will need to travel to exit the material, then an interaction occurs. Following this, the cross-section for each relevant neutron process is looked up and used to sample which interaction will occur. Then, the selected interaction is applied to the neutron in the way described by the user selected neutron data libraries, and simulation information (including tallies) are updated. In order to stop the neutron population from increasing or decreasing exponentially, the fission neutron yield is divided by the $K_{\text{gen}}$ calculated from the previous run.

#### 2.6.7.2 Convergence

Although MCNP does calculate the Shannon entropy of the neutron population after every generation, it does not use it to determine when convergence has occurred during a criticality simulation. Instead the user must specify how many neutron generations must pass until the neutron population can be said to be converged, beginning the collection of important data. Although this gives the user full control over when data will start to be collected, it can result in data being collected too early compromising the results, or too late, resulting in the needless waste of computational time.

#### 2.6.7.3 Assumptions

MCNP6.1 uses several assumptions and simplifications to in order to improve its performance. In this section, the most important simplifications that are relevant to this project will be discussed.

##### 2.6.7.3.1 Variance Reduction Techniques

By default, MCNP uses variance reduction techniques to reduce the simulation time and the statistical variance between results.

###### 2.6.7.3.1.1 Fission Yield Sampling

One of the most impactful variance reduction techniques used by MCNP, is the lack of fission yield sampling. What MCNP does instead is, every time there is an interaction, the average fission yield is looked up, multiplied by the probability of the interaction being fission, and then added on to a sum. At the end of each time step, this sum is used as the total number of daughters. Although this greatly reduces the statistical variance in the results caused by sampling the neutron yield, it also makes it impossible for MCNP to account for the effect of different neutron yields creating different energy distributions.

###### 2.6.7.3.1.2 Implicit Absorption

Another important variation reduction technique used by MCNP is implicit absorption. Implicit absorption works by multiplying the weight (importance) of the particle by the cross-section of absorption occurring divided by the summed cross-section of all the possible interactions every time an interaction occurs (as is shown in Equation 2.15).

$$W_{new} = \left(1 - \frac{\sigma_a}{\sigma_T}\right) \times W_{old}$$

*Equation 2.15: Shows how the weight of particles in MCNP are adjusted by implicit capture*

When the weight of the particle drops below the cutoff weight set by the user, MCNP rolls to determine whether the particle will be killed. Although this method works as a variance reduction technique it also can cause biasing in the number of absorptions depending on what the weight cut off is set to (if the cutoff is very high then the neutrons will be assumed to be absorbed more often than they should.This is not a problem when the default settings are used) [1].

#### 2.6.7.3.2 Neutron Physics Simplifications

As another way of improving the efficiency of the code, MCNP also makes some simplifications to the physics.

##### 2.6.7.3.2.1 No Enforcement of the Conservation Laws

Conservation of energy and momentum are not enforced by MCNP. Instead it is assumed that the average results achieved from sampling the neutron data libraries many times will be in line with these laws.

##### 2.6.7.3.2.2 Instantaneous Delayed Neutron Creation

In MCNP, delayed neutrons are created immediately following fission (like a prompt neutron) by default. This essentially assumes that the neutron population is independent of time and limits the ability of MCNP to simulate truly transient cases. The main reason MCNP uses this assumption is the enormous amount of simulation time it would take to build up a stable delayed neutron population. However, even if this was not a factor, MCNP would still have to make this assumption, since MCNP does not accurately keep track of time when performing a criticality calculation (making it impossible to know when to insert a delayed neutron).

##### 2.6.7.3.2.3 Classical Relativity

MCNP uses Galilean relativity rather than special relativity when transferring data between different reference frames. This assumes that the neutrons are not traveling at speed near that of light, making Galilean relativity and special relativity essentially equivalent.

##### 2.6.7.3.2.4 Quasi Criticality

Since MCNP uses the fission generational criticality method for determining $K_{eff}$, it is also making the assumption that the neutron population is independent of time (as described here, 2.6.1.2). This is probably the most limiting assumption that MCNP makes, since it ensures that the results of MCNP will only be accurate for simulating cases near critical.

### *2.6.7.4 NeutronHP Model*

In this section we will describe the different distributions used by the NeutronHP model (described here, 2.6.5) of MCNP6.1, to sample the outgoing neutron energy, and angle, for each neutron interaction. The neutron capture reaction is ignored since it only produces particles other than neutrons, which are not of interest for this project. Except for a few exceptions, each of these distributions work by first calculating the differential cross-sections (probability of occurrence), for each of the possible incoming and outgoing neutron combination. Then, one of the possible combinations is randomly chosen, with more probable combinations being chosen proportionately more often. Since the last step of this process is the same for all of the distributions, only the calculation of the differential cross-section will be discussed. The preprocessed ENDF files used by the

NeutronHP model in MCNP6.1 are stored in the XSDIR set by the user. For a full description about how the final state data is stored in the MCNP6.1 library, refer to volume three of MCNP's user manual [13].

#### 2.6.7.4.1 Elastic Scattering

There are two different representations used by MCNP6.1 to determine the outcome of elastic scattering [13]. Both of the representations describe the relationship between the differential cross-section (probability of occurrence) and the deflection angle. The energy of the outgoing neutron is obtained from the laws of conservation after the deflection angle has been sampled.

The first representation is the tabular distribution. In this case the differential cross-section is interpolated from a two dimensional table, indexed by the incoming neutron energy and the cosine of the deflection angle (described here, Equation 2.16).

$$\frac{d\sigma}{d\theta} = \frac{d\sigma}{d\theta}\,(E, cos\theta)$$

*Equation 2.16: Shows the relationship between the differential cross-section $\left(\frac{d\sigma}{d\theta}\right)$, the cosine of the scattering angle ($cos\theta$), and the incoming neutron energy (E) for the tabular distribution.*

The second representation is the equiprobable bin distribution. For this type of distribution, the differential cross-section is dependent on incoming energy and the outgoing neutron angle region (or bin). For each incoming neutron energy, the differential cross-sections are set for each region of the outgoing angular spectrum, such that the integrated probability for each region is uniform (described by Equation 2.17).

$$Const = \int_{\theta_{i-1}}^{\theta_i} \frac{d\sigma}{d\theta}\,(E, i)\,d\theta = \frac{d\sigma}{d\theta}\,(E, i) \times (\theta_i - \theta_{i-1})$$

*Equation 2.17: Shows the algorithm used to determine the differential cross-section $\left(\frac{d\sigma}{d\theta}\,(E, i)\right)$, using the upper and lower angular boundary ($\theta_i$ and $\theta_{i-1}$) of region 'i', Note: E is the incoming neutron energy.*

The differential cross-section can then be interpolated from the two dimensional table indexed by the incoming neutron energy and the outgoing neutron angular region (described here Equation 2.18).

$$\frac{d\sigma}{d\theta} = \frac{d\sigma}{d\theta}\,(E, i)$$

*Equation 2.18: Shows the relationship between the differential cross-section $\left(\frac{d\sigma}{d\theta}\right)$, the incoming neutron energy (E), and the outgoing angular region (i) for the equiprobable bin distribution.*

#### 2.6.7.4.2 Fission

For the fission process, three different representations are used to describe the outgoing neutron angular distribution, and nine different representations used to describe the outgoing neutron energy distribution [13].

##### 2.6.7.4.2.1 Outgoing Angular Representations

The first representation of the outgoing neutron angular distribution, is the tabular distribution. In this case the differential cross-section is interpolated from a two dimensional table, indexed by the incoming neutron energy, and the cosine of the outgoing neutron angle (described here, Equation 2.19).

$$\frac{d\sigma}{d\theta} = \frac{d\sigma}{d\theta}\,(E, cos\theta)$$

*Equation 2.19: Shows the relationship between the differential cross-section $\left(\frac{d\sigma}{d\theta}\right)$, the cosine of the scattering angle ($cos\theta$), and the incoming neutron energy (E) for the tabular distribution.*

The second representation is the equiprobable bin distribution. For this type of distribution, the differential cross-section is dependent on incoming energy and the outgoing neutron angle region (or bin). For each incoming neutron energy, the differential cross-sections are set for each region of the outgoing angular spectrum, such that the integrated probability for each region is uniform (described by Equation 2.20).

$$Const = \int_{\theta_{i-1}}^{\theta_i} \frac{d\sigma}{d\theta}\,(E, i)\,d\theta = \frac{d\sigma}{d\theta}\,(E, i) \times (\theta_i - \theta_{i-1})$$

*Equation 2.20: Shows the algorithm used to determine the differential cross-section $\left(\frac{d\sigma}{d\theta}\,(E, i)\right)$, using the upper and lower angular boundary ($\theta_i$ and $\theta_{i-1}$) of region 'i', Note: E is the incoming neutron energy.*

The differential cross-section can then be interpolated from the two dimensional table indexed by the incoming neutron energy and the outgoing neutron angular region (described here Equation 2.21).

$$\frac{d\sigma}{d\theta} = \frac{d\sigma}{d\theta}\,(E, i)$$

*Equation 2.21: Shows the relationship between the differential cross-section $\left(\frac{d\sigma}{d\theta}\right)$, the incoming neutron energy (E), and the outgoing angular region (i) for the equiprobable bin distribution.*

The third representation of the outgoing neutron angular distribution, is an isotropic distribution. As can be seen in Equation 2.22, every possible outgoing neutron angle in the center of mass reference, is given an equal probability, and thus is equally likely to occur.

$$\frac{d\sigma}{d\theta} = Const$$

*Equation 2.22: Shows how the differential cross-section $\left(\frac{d\sigma}{d\theta}\right)$ is calculated for an isotropic distribution.*

#### 2.6.7.4.2.2 Outgoing Energy Representations

The first representation is the equiprobable bin distribution. For this type of distribution, the differential cross-section is dependent on incoming energy and the outgoing neutron energy region (or bin). For each incoming neutron energy, the differential cross-sections are set for each region of the outgoing energy spectrum, such that the integrated probability for each region is uniform (described by Equation 2.23).

$$Const = \int_{\theta_{i-1}}^{\theta_i} \frac{d\sigma}{dE'}(E,i)\,d\theta = \frac{d\sigma}{dE'}(E,i) \times (E'_i - E'_{i-1})$$

*Equation 2.23: Shows the algorithm used to determine the differential cross-section $\left(\frac{d\sigma}{dE'}(E,i)\right)$, using the upper and lower energy boundary ($E'_i$ and $E'_{i-1}$) of region 'i', Note: E is the incoming neutron energy.*

The differential cross-section can then be interpolated from the two dimensional table indexed by the incoming neutron energy and the outgoing neutron energy region (described here Equation 2.24).

$$\frac{d\sigma}{dE'} = \frac{d\sigma}{dE'}(E,i)$$

*Equation 2.24: Shows the relationship between the differential cross-section $\left(\frac{d\sigma}{d\theta}\right)$, the incoming neutron energy (E), and the outgoing angular region (i) for the equiprobable bin distribution.*

The second representation of the outgoing neutron energy distribution, is level scattering. In this case the outgoing neutron energy is not sampled. Instead it is calculated from a linear transformation of the incoming neutron energy (shown here, Equation 2.25). The parameters $C_1$ and $C_2$ are constants that are unique to this particular distribution.

$$E' = C_2(E - C_1)$$

*Equation 2.25: Shows the relationship between the outgoing neutron energy (E'), the incoming neutron energy (E), parameter 1 ($C_1$), and parameter 2 ($C_2$) for the level scattering.*

The third representation of the outgoing neutron energy distribution, is the tabular distribution. In this case the differential cross-section $(\frac{d\sigma}{dE'})$ is interpolated from a two dimensional table, indexed by the incoming neutron energy, and the outgoing neutron energy (described here, Equation 2.26).

$$\frac{d\sigma}{dE'} = \frac{d\sigma}{dE'}(E, E')$$

*Equation 2.26: Shows the relationship between the differential cross-section $\left(\frac{d\sigma}{dE'}\right)$, the incoming neutron energy (E), and the outgoing neutron energy ($E'$) for the tabular distribution.*

The fourth representation of the outgoing neutron energy distribution, is the general evaporation spectrum shown in Equation 2.27. The multiplier (X(ε)) shown in Equation 2.27, is randomly sampled from a multiplier probability distribution (X(x), defined in the high precision dataset). This X(x) distribution is what defines the shape of the outgoing neutron energy distribution at every incoming energy. The T(E) variable shown in Equation 2.27, is interpolated from a one dimensional table, indexed by the incoming neutron energy.

$$E' = \mathrm{X}(\varepsilon) \times \mathrm{T}(\mathrm{E})$$

*Equation 2.27: Shows the relationship between the outgoing neutron energy, the sampled outgoing neutron energy multiplier ($X(\varepsilon)$), and the interpolated outgoing neutron energy multiplier ($T(E)$).*

The fifth representation of the outgoing neutron energy distribution, is the Maxwell fission spectrum shown in Equation 2.28. This representation is derived in the framework of quantum mechanics, by treating the constituents of the nucleus as particles in a free gas [14]. The average kinetic energy of the constituents is determined by the effective temperature of the nucleus (the same way the kinetic energy of molecules in a material are determined using the materials temperature). The effective temperature ($\Theta(E)$) used in Equation 2.28, is interpolated from a one dimensional table indexed by the incoming neutron energy.

$$\frac{d\sigma}{dE'} \propto \sqrt{E'} e^{E'/\Theta(E)}$$

*Equation 2.28: Shows the relationship between the differential cross-section $\left(\frac{d\sigma}{dE'}\right)$, the effective temperature ($\Theta(E)$), the incoming neutron energy (E), and the outgoing neutron energy ($E'$) for the Maxwell fission spectrum.*

Additionally, a restriction energy (U) is used to limit the maximum outgoing neutron energy possible (see Equation 2.29).

$$E'_{max} = E - U$$

*Equation 2.29: Shows the relationship between the maximum outgoing energy possible ($E'_{max}$), the restriction energy (U), and the incoming neutron energy (E).*

The sixth representation of the outgoing neutron energy distribution, is the evaporation spectrum shown in Equation 2.30. This representation was also derived by treating the constituents of the nucleus as particles in a free gas, but it uses a different fitting function around the exponent to better match different experimental data [15]. The effective

temperature ($\Theta(E)$) used in Equation 2.30, is interpolated from a one dimensional table indexed by the incoming neutron energy.

$$\frac{d\sigma}{dE'} \propto E' e^{E'/\Theta(E)}$$

*Equation 2.30: Shows the relationship between the differential cross-section $\left(\frac{d\sigma}{dE'}\right)$, the effective temperature ($\Theta(E)$), the incoming neutron energy (E), and the outgoing neutron energy ($E'$) for the evaporation spectrum.*

Additionally, a restriction energy (U) is used to limit the maximum outgoing neutron energy possible (see Equation 2.31).

$$E'_{max} = E - U$$

*Equation 2.31: Shows the relationship between the maximum outgoing energy possible ($E'_{max}$), the restriction energy (U), and the incoming neutron energy (E).*

The seventh representation of the outgoing neutron energy distribution, is the energy dependent Watt spectrum shown in Equation 2.32. This representation was also derived by treating the constituents of the nucleus as particles in a free gas. Except it uses different fitting function around the exponent to better match different experimental data [15]. The parameters a(E) and b(E) used in Equation 2.32, are interpolated from separate one dimensional tables indexed by the incoming neutron energy.

$$\frac{d\sigma}{dE'} \propto E' e^{-E'/a(E)} \sinh\sqrt{b(E)E'}$$

*Equation 2.32: Shows the relationship between the differential cross-section $\left(\frac{d\sigma}{dE'}\right)$, parameter $a(E)$ , parameter $b(E)$, the incoming neutron energy (E), and the outgoing neutron energy ($E'$) for the energy dependent Watt spectrum.*

Additionally, a restriction energy (U) is used to limit the maximum outgoing neutron energy possible (see Equation 2.33).

$$E'_{max} = E - U$$

*Equation 2.33: Shows the relationship between the maximum outgoing energy possible ($E'_{max}$), the restriction energy (U), and the incoming neutron energy (E).*

The eighth representation of the outgoing neutron energy distribution, is the tabular linear functions distribution. Each incoming neutron energy points to a list of linear functions with corresponding probabilities of being selected. Every time this representation is called, a linear function is sampled from the list corresponding to the current incoming neutron energy. The sampled linear function is then applied to the incoming neutron energy to determine the outgoing neutron energy (described by Equation 2.34)

$$E' = C_2(E,\varepsilon)\big(E - C_1(E,\varepsilon)\big)$$

*Equation 2.34: Shows the relationship between the outgoing neutron energy (E'), the incoming neutron energy (E), the sampled parameter 1 $C_2(E,\varepsilon)$, and the sampled parameter 2 ($C_1(E,\varepsilon)$) for the tabular linear functions distribution. Note ε is a randomly sampled number used to determine which linear function will be used.*

The ninth representation of the outgoing neutron energy distribution, is the modified tabular linear functions distribution. Each incoming neutron energy points to a list of linear functions with corresponding probabilities of being selected. Every time this representation is called, a linear function is sampled from the list corresponding to the current incoming neutron energy. The sampled linear function is then applied to the incoming neutron energy to determine the outgoing neutron energy (described by Equation 2.35)

$$E' = C_2(E,\varepsilon) \times E$$

*Equation 2.35: Shows the relationship between the outgoing neutron energy (E'), the incoming neutron energy (E), and the sampled parameter 1 $C_2(E,\varepsilon)$ for the modified tabular linear functions distribution. Note ε is a randomly sampled number used to determine which linear function will be used.*

#### 2.6.7.4.3 Inelastic Scattering

For the inelastic process the same angular representations and energy representations can be used as the fission process shown above 2.6.7.4.2 [13]. However, there are also four combined angular and energy representations.

The first representation of the combined outgoing neutron angular and energy distribution, is the Kalbach-87 formalism. First the differential cross-section of the outgoing energy spectrum $\left(\frac{d\sigma}{dE'}\right)$, is interpolated from a two dimensional table, indexed by the incoming neutron energy, and the outgoing neutron energy (described here, Equation 2.36).

$$\frac{d\sigma}{dE'} = \frac{d\sigma}{dE'}(E,E')$$

*Equation 2.36: Shows the relationship between the differential cross-section $\left(\frac{d\sigma}{dE'}\right)$, the incoming neutron energy (E), and the outgoing neutron energy (E′) for the center of mass 3D tabular distribution.*

Next the differential cross-section of the outgoing angular spectrum $\left(\frac{d\sigma}{d\theta}\right)$, is sampled from a Kalbach distribution shown here, Equation 2.37. The angular distribution slope (A) and the precompound fraction (R), are interpolated from two dimensional tables, indexed by the incoming neutron energy, and the outgoing neutron energy.

$$\frac{d\sigma}{d\theta}\,(E,E',\theta) = \frac{1}{2}\frac{A(E,E')}{\sinh(A(E,E'))}\left(\cosh(A(E,E')\theta) + R(E,E')\sinh(A(E,E')\theta)\right)$$

*Equation 2.37: Shows the relationship between the differential cross-section $\left(\frac{d\sigma}{d\theta}\right)$, the angular distribution slope (A), the precompound fraction (R), the incoming neutron energy (E), the outgoing neutron energy ($E'$), and the outgoing neutron angle ($\theta$) for the center of mass 3D tabular distribution.*

The second representation of the combined outgoing neutron angular and energy distribution, is the center of mass three dimensional tabular distribution. In this case the differential cross-section $\left(\frac{d\sigma}{dE'd\theta}\right)$, is interpolated from a three dimensional table, indexed by the incoming neutron energy, the outgoing neutron energy, and the outgoing neutron angle, in the center of mass frame (described here, Equation 2.38).

$$\frac{d\sigma}{dE'd\theta} = \frac{d\sigma}{dE'd\theta}\,(E,E',\theta)$$

*Equation 2.38: Shows the relationship between the differential cross-section $\left(\frac{d\sigma}{dE'}\right)$, the incoming neutron energy (E), the outgoing neutron energy ($E'$), and the outgoing neutron angle ($\theta$) for the center of mass 3D tabular distribution.*

The third representation of the combined outgoing neutron angular and energy distribution, is the N-body phase-space distribution. The outgoing neutron energy multiplier is sampled from the probability distribution described in Equation 2.39. The maximum outgoing neutron energy ($E_{max}$) is determined using the number of outgoing products, the total mass of the outgoing products, Qvalue of the reaction, and the conservation laws.

$$P(T) = C(n)\sqrt{T}\,(E'_{max} - T)^{\frac{3n}{2}-4}$$

*Equation 2.39: Shows the relationship between the probability of this outgoing energy multiplier ($P(T)$), the outgoing energy multiplier (T), the maximum outgoing neutron energy ($E'_{max}$), parameter ($C(n)$), and the number of products (n) for the N-body phase-space distribution.*

The outgoing neutron energy is then calculated by multiply the maximum possible outgoing neutron energy by the sampled outgoing neutron energy multiplier (described by Equation 2.40)

$$E' = T \times E'_{max}$$

*Equation 2.40: Shows the relationship between the outgoing neutron energy (E'), the incoming neutron energy (E), the outgoing energy multiplier (T), and the maximum outgoing neutron energy ($E'_{max}$) for the tabular linear functions distribution.*

The outgoing neutron angle is sampled from an isotropic distribution (described by Equation 2.41).

$$\frac{d\sigma}{d\theta} = Const$$

*Equation 2.41: Shows how the differential cross-section $\left(\frac{d\sigma}{d\theta}\right)$ is calculated for an isotropic emission distribution.*

The final representation of the combined outgoing neutron angular and energy distribution, is the laboratory three dimensional tabular distribution. In this case the differential cross-section $\left(\frac{d\sigma}{dE'd\theta}\right)$, is interpolated from a three dimensional table, indexed by the incoming neutron energy, the outgoing neutron angle, and the outgoing neutron energy, in the laboratory frame (described here, Equation 2.42).

$$\frac{d\sigma}{dE'd\theta} = \frac{d\sigma}{dE'd\theta}\ (E, \theta, E')$$

*Equation 2.42: Shows the relationship between the differential cross-section $\left(\frac{d\sigma}{dE'}\right)$, the incoming neutron energy (E), the outgoing neutron angle, and the outgoing neutron energy ($E'$) for the laboratory 3D tabular distribution.*

### 2.6.8 G4STORK

#### *2.6.8.1 Summary*

G4STORK, which stands for Geant4 Stochastic Reactor Kinetics, was initially created by Liam Russell, under the supervision of Dr. Adriaan Buijs at McMaster University [1]. The purpose of G4STORK was to create a truly time dependent Monte Carlo reactor physics code, using minimal assumptions. This would allow G4STORK to model rapid transients (such as accident scenarios) with as much accuracy as possible, which for reasons described in the previous subchapter 1.1, is greatly needed by the nuclear industry. G4STORK and GEANT4 are both open source projects, allowing researchers to easily acquire them for free and make changes to the code as they see fit. Both GEANT4 and G4STORK have been created using a strict hierarchy structure to make it easier for developers to insert their own code so that they can both be adapted to a wide range of situations with minimal effort.

Starting with an initial guess as to what the equilibrium neutron position, and energy distribution will look like, the G4STORK code works by tracking individual neutrons through a geometry provided by the user in steps of time. At each time step, the important parameters such as $K_{\text{eff}}$ (calculated using the dynamic criticality method shown in Equation 2.11) and Shannon entropy are determined and the neutron population is renormalized to the initial number of neutrons. These important parameters are not recorded into the final results until the spatial distribution of the neutrons has converged. Since G4STORK is a stochastic simulation, the processes that the neutrons undergo as they move throughout the geometry are randomly selected from a list of potential processes that are weighted based on their cross-section. This is dependent on the isotopic

composition of the material that the neutron is currently traversing and the kinetic energy of the neutron relative to the nuclei in its path. While the composition of the material is defined in the geometry by the user, the relative kinetic energy of the neutron to the nuclei is determined by the on the fly Doppler broadening algorithm described in 2.6.2.

#### *2.6.8.2 Convergence*

As discussed in 2.6.3, the neutron population is often assumed to be converged, when the Shannon entropy of the neutron spatial distribution is constant between simulation iterations. In G4STORK the Shannon entropy is said to be constant when the Shannon entropies of each of the last 25 time steps do not deviate from the mean Shannon entropy (taken from the last 25 time steps) beyond a set limit (by default the limit is 2%). As mentioned in the summary, G4STORK does not begin collecting results until the neutron population has been flagged as converged. The larger the difference between the initial neutron population and the converged neutron population, the longer it will take for convergence to occur. Thus, for the sake of code performance, it is important that the user choose an initial neutron population that is as close to the converged neutron population as possible.

#### *2.6.8.3 Assumptions*

##### 2.6.8.3.1 Unresolved Resonance Regime

When the width of the resonant peaks in the cross-section data become too thin to be measured precisely, G4STORK uses the average cross-section instead. This is a fairly minor assumption for nuclear reactor research, since this effect occurs at energies much larger than the usual neutron energy within a nuclear reactor.

##### 2.6.8.3.2 Discretization of Time

While some event can occur continuously with time in G4STORK, such as particle creation, changes to the geometry can only occur in between time steps. Thus during a time step, important feedback effects such as changes in fuel temperature or composition cannot occur. The inaccuracy of this assumption becomes negligible, if the time step used is significantly smaller than the period of time needed for the effect (heating, burn up, …) to significantly impact the results.

#### *2.6.8.4 The NeutronHP Model of GEANT4*

In this section we will describe the different distributions used by the NeutronHP model (described here, 2.6.5) of GEANT4, to sample the outgoing neutron energy, and angle, for each neutron interaction. The neutron capture reaction is ignored since it only produces particles other than neutrons, which are not of interest for this project. In general, each of these distributions work by first calculating the differential cross-sections (probability of occurrence), for each of the possible incoming and outgoing neutron combination. Then, one of the possible combinations is randomly chosen, with more probable combinations being chosen proportionately more often. Since the last step of

this process is the same for all of the distributions, only the calculation of the differential cross-section will be discussed. The preprocessed ENDF files used by the NeutronHP model in GEANT4 are stored in the G4NDL library. For a full description about how the final state data is stored in the G4NDL library, refer to our manual about the decryption of G4NDL [16].

#### 2.6.8.4.1 Elastic Scattering

There are two different representations used by GEANT4 to determine the outcome of elastic scattering [17]. Both of the representations describe the relationship between the differential cross-section (probability of occurrence) and the deflection angle. The energy of the outgoing neutron is obtained from the laws of conservation after the deflection angle has been sampled.

The first of representation is the tabular distribution. In this case the differential cross-section is interpolated from a two dimensional table, indexed by the incoming neutron energy and the cosine of the deflection angle (described here, Equation 2.43).

$$\frac{d\sigma}{d\theta} = \frac{d\sigma}{d\theta}(E, cos\theta)$$

*Equation 2.43: Shows the relationship between the differential cross-section $\left(\frac{d\sigma}{d\theta}\right)$, the cosine of the scattering angle ($cos\theta$), and the incoming neutron energy (E) for the tabular distribution.*

The second representation used is a series of Legendre polynomials, shown below in Equation 2.44. This representation comes from the derivation of neutron scattering using the partial waves method in the framework of quantum mechanics [18]. The Legendre polynomial coefficients ($a_l(E)$) used in Equation 2.44, are interpolated from a one dimensional table indexed by the incoming neutron energy.

$$\frac{2\pi}{\sigma(E)} \times \frac{d\sigma}{d\theta}(E, cos\theta) = \sum_{l=0}^{n} \frac{2l + 1}{2} a_l(E) P_l(cos\theta)$$

*Equation 2.44: Shows the relationship between the differential cross-section $\left(\frac{d\sigma}{d\theta}\right)$, the cosine of the scattering angle ($cos\theta$), and the incoming neutron energy (E) for the Legendre polynomial series representation. Note: l represents the order of the Legendre polynomial, n represents the number of Legendre polynomials, $a_l$ represents the coefficient of the lth Legendre polynomial, and $P_l$ is the lth Legendre polynomial.*

#### 2.6.8.4.2 Fission

For the fission process, three different representations are used to describe the outgoing neutron angular distribution, and six different representations used to describe the outgoing neutron energy distribution [17].

#### 2.6.8.4.2.1 Outgoing Angular Representations

The first representation of the outgoing neutron angular distribution, is the tabular distribution. In this case the differential cross-section is interpolated from a two dimensional table, indexed by the incoming neutron energy, and the cosine of the outgoing neutron angle (described here, Equation 2.45).

$$\frac{d\sigma}{d\theta} = \frac{d\sigma}{d\theta}\,(E, cos\theta)$$

*Equation 2.45: Shows the relationship between the differential cross-section $\left(\frac{d\sigma}{d\theta}\right)$, the cosine of the scattering angle ($cos\theta$), and the incoming neutron energy (E) for the tabular distribution.*

The second representation of the outgoing neutron angular distribution, is a series of Legendre polynomials, shown below in Equation 2.46. This representation comes from the derivation of neutron scattering using the partial waves method in the framework of quantum mechanics [18]. The Legendre polynomial coefficients ($a_l(E)$) used in Equation 2.46, are interpolated from a one dimensional table indexed by the incoming neutron energy.

$$\frac{2\pi}{\sigma(E)} \times \frac{d\sigma}{d\theta}\;(E, cos\theta) = \sum_{l=0}^{n} \frac{2l\ +\ 1}{2} a_l(E) P_l(cos\theta)$$

*Equation 2.46: Shows the relationship between the differential cross-section $\left(\frac{d\sigma}{d\theta}\right)$, the cosine of the scattering angle ($cos\theta$), and the incoming neutron energy (E) for the Legendre polynomial series representation. Note: $l$ represents the order of the Legendre polynomial, $n$ represents the number of Legendre polynomials, $a_l$ represents the coefficient of the lth Legendre polynomial, and $P_l$ is the lth Legendre polynomial.*

The third representation of the outgoing neutron angular distribution, is an isotropic distribution. As can be seen in Equation 2.47, every possible outgoing neutron angle in the center of mass reference, is given an equal probability, and thus is equally likely to occur.

$$\frac{d\sigma}{d\theta} = Const$$

*Equation 2.47: Shows how the differential cross-section $\left(\frac{d\sigma}{d\theta}\right)$ is calculated for an isotropic distribution.*

#### 2.6.8.4.2.2 Outgoing Energy Representations

The first representation of the outgoing neutron energy distribution, is the tabular distribution. In this case the differential cross-section ($\frac{d\sigma}{dE'}$) is interpolated from a two dimensional table, indexed by the incoming neutron energy, and the outgoing neutron energy (described here, Equation 2.48).

$$\frac{d\sigma}{dE'} = \frac{d\sigma}{dE'}(E, E')$$

*Equation 2.48: Shows the relationship between the differential cross-section $\left(\frac{d\sigma}{dE'}\right)$, the incoming neutron energy (E), and the outgoing neutron energy ($E'$) for the tabular distribution.*

The second representation of the outgoing neutron energy distribution, is the general evaporation spectrum shown in Equation 2.49. The multiplier (X(ε)) shown in Equation 2.49, is randomly sampled from a multiplier probability distribution (X(x), which is defined in the G4NDL data file). This X(x) distribution is what defines the shape of the outgoing neutron energy distribution at every incoming energy. The T(E) variable shown in Equation 2.49, is interpolated from a one dimensional table, indexed by the incoming neutron energy.

$$E' = \mathrm{X}(\varepsilon) \times \mathrm{T}(\mathrm{E})$$

*Equation 2.49: Shows the relationship between the outgoing neutron energy, the sampled outgoing neutron energy multiplier ($X(\varepsilon)$), and the interpolated outgoing neutron energy multiplier ($T(E)$).*

The third representation of the outgoing neutron energy distribution, is the Maxwell fission spectrum shown in Equation 2.50. This representation is derived in the framework of quantum mechanics, by treating the constituents of the nucleus as particles in a free gas [14]. The average kinetic energy of the constituents is determined by the effective temperature of the nucleus (the same way the kinetic energy of molecules in a material are determined using the materials temperature). The effective temperature ($\Theta(E)$) used in Equation 2.50, is interpolated from a one dimensional table indexed by the incoming neutron energy.

$$\frac{d\sigma}{dE'} \propto \sqrt{E'} e^{E'/\Theta(E)}$$

*Equation 2.50: Shows the relationship between the differential cross-section $\left(\frac{d\sigma}{dE'}\right)$, the effective temperature ($\Theta(E)$), the incoming neutron energy (E), and the outgoing neutron energy ($E'$) for the Maxwell fission spectrum.*

The fourth representation of the outgoing neutron energy distribution, is the evaporation spectrum shown in Equation 2.51. This representation was also derived by treating the constituents of the nucleus as particles in a free gas. Except it uses different fitting function around the exponent to better match different experimental data [15]. The effective temperature ($\Theta(E)$) used in Equation 2.51, is interpolated from a one dimensional table indexed by the incoming neutron energy.

$$\frac{d\sigma}{dE'} \propto E' e^{E'/\Theta(E)}$$

*Equation 2.51: Shows the relationship between the differential cross-section $\left(\frac{d\sigma}{dE'}\right)$, the effective temperature $(\Theta(E))$, the incoming neutron energy (E), and the outgoing neutron energy $(E')$ for the evaporation spectrum.*

The fifth representation of the outgoing neutron energy distribution, is the energy dependent Watt spectrum shown in Equation 2.52. This representation was also derived by treating the constituents of the nucleus as particles in a free gas. Except it uses different fitting function around the exponent to better match different experimental data [15]. The parameters a(E) and b(E) used in Equation 2.52, are interpolated from separate one dimensional tables indexed by the incoming neutron energy.

$$\frac{d\sigma}{dE'} \propto E' e^{-E'/a(E)} \sinh\sqrt{b(E)E'}$$

*Equation 2.52: Shows the relationship between the differential cross-section $\left(\frac{d\sigma}{dE'}\right)$, parameter $a(E)$ , parameter $b(E)$, the incoming neutron energy (E), and the outgoing neutron energy $(E')$ for the energy dependent Watt spectrum.*

The final representation of the outgoing neutron energy distribution, is the Madland Nix spectrum shown in Equation 2.53. This representation is derived by treating the constituents of the nucleus as particles in a free gas, and by taking the different sizes of the fission products into account [15]. The heavy and light fragment kinetic energies ($K_l$ and $K_H$) used in Equation 2.53, are assumed to be independent of the incoming neutron energy. The effective temperature ($\Theta(E)$) used in Equation 2.53, is interpolated from a one dimensional table indexed by the incoming neutron energy.

$$\frac{d\sigma}{dE'} = \frac{1}{2}\left(g(E', K_l) + g(E', K_H)\right)$$

$$g(E', K) = \frac{1}{3\sqrt{K\Theta}}\left(u_2^{\frac{3}{2}}E_1(u_2) - u_1^{\frac{3}{2}}E_1(u_1) + \gamma\left(\frac{3}{2}, u_2\right) - \gamma\left(\frac{3}{2}, u_1\right)\right)$$

$$u_1(E', K) = \frac{\left(\sqrt{E'} - \sqrt{K}\right)^2}{\Theta}$$

$$u_2(E', K) = \frac{\left(\sqrt{E'} + \sqrt{K}\right)^2}{\Theta}$$

*Equation 2.53: Shows the relationship between the differential cross-section $\left(\frac{d\sigma}{dE'}\right)$, the kinetic energy of the light fission fragment ($K_l$), the kinetic energy of the heavy fission fragment ($K_H$), The effective temperature $(\Theta(E))$, the incoming neutron energy (E), and the outgoing neutron energy $(E')$ for the energy dependent Watt spectrum. Note $E_1(x)$ is the exponential integral function, and $\gamma(x)$ is the incomplete gamma function.*

#### 2.6.8.4.3 Inelastic Scattering

For the inelastic process the same angular representations and energy representations can be used as the fission process shown above 2.6.8.4.2 [17]. However, there are also five combined angular and energy representations.

The first representation of the combined outgoing neutron angular and energy distribution, is the isotropic emission distribution. As can be seen in Equation 2.54, every possible outgoing neutron angle in the center of mass reference, is given an equal probability, and thus is equally likely to occur. The outgoing neutron energy is then determined using the Qvalue of the reaction and the conservation laws.

$$\frac{d\sigma}{d\theta} = Const$$

*Equation 2.54: Shows how the differential cross-section $\left(\frac{d\sigma}{d\theta}\right)$ is calculated for an isotropic emission distribution.*

The second representation of the combined outgoing neutron angular and energy distribution, is the discrete two-body kinematics distribution. As can be seen in Equation 2.55, the outgoing angle probability is determined using a series of Legendre polynomials. This representation comes from the derivation of neutron scattering, using the partial waves method, in the framework of quantum mechanics [18].The outgoing neutron energy is then determined using the Qvalue of the reaction and the conservation laws.

$$\frac{2\pi}{\sigma(E)} \times \frac{d\sigma}{d\theta}\ (E, cos\theta) = \sum_{l=0}^{n} \frac{2l\ +\ 1}{2} a_l(E) P_l(cos\theta)$$

*Equation 2.55: Shows the relationship between the differential cross-section $\left(\frac{d\sigma}{d\theta}\right)$, the cosine of the scattering angle ($cos\theta$), and the incoming neutron energy (E) for the discrete two-body kinematics representation. Note: l represents the order of the Legendre polynomial, n represents the number of Legendre polynomials, $a_l$ represents the coefficient of the lth Legendre polynomial, and $P_l$ is the lth Legendre polynomial.*

The third representation of the combined outgoing neutron angular and energy distribution, is the N-body phase-space distribution. The outgoing neutron energy is sampled from the probability distribution described in Equation 2.56. The maximum outgoing neutron energy ($E_{max}$) is determined using the number of outgoing products, the total mass of the outgoing products, Qvalue of the reaction, and the conservation laws. The outgoing neutron angle is sampled from an isotropic distribution (see Equation 2.54).

$$\frac{d\sigma}{dE'}\ (E') = \sqrt{E}\ (E_{max} - E)^{\frac{3n}{2}-4}$$

*Equation 2.56: Shows the relationship between the differential cross-section $\left(\frac{d\sigma}{dE'}\right)$, the incoming neutron energy (E), the maximum outgoing neutron energy ($E_{max}$), the number of products (n), and the outgoing neutron energy ($E'$) for the N-body phase-space distribution.*

The fourth representation of the combined outgoing neutron angular and energy distribution, is the center of mass three dimensional tabular distribution. In this case the differential cross-section $\left(\frac{d\sigma}{dE'd\theta}\right)$, is interpolated from a three dimensional table, indexed by the incoming neutron energy, the outgoing neutron energy, and the outgoing neutron angle, in the center of the mass frame (described here, Equation 2.57).

$$\frac{d\sigma}{dE'd\theta} = \frac{d\sigma}{dE'd\theta}(E, E', \theta)$$

*Equation 2.57: Shows the relationship between the differential cross-section $\left(\frac{d\sigma}{dE'}\right)$, the incoming neutron energy (E), the outgoing neutron energy ($E'$), and the outgoing neutron angle for the center of mass 3D tabular distribution.*

The final representation of the combined outgoing neutron angular and energy distribution, is the laboratory three dimensional tabular distribution. In this case the differential cross-section $\left(\frac{d\sigma}{dE'd\theta}\right)$, is interpolated from a three dimensional table, indexed by the incoming neutron energy, the outgoing neutron angle, and the outgoing neutron energy, in the laboratory frame (described here, Equation 2.58).

$$\frac{d\sigma}{dE'd\theta} = \frac{d\sigma}{dE'd\theta}(E, \theta, E')$$

*Equation 2.58: Shows the relationship between the differential cross-section $\left(\frac{d\sigma}{dE'}\right)$, the incoming neutron energy (E), the outgoing neutron angle, and the outgoing neutron energy ($E'$) for the laboratory 3D tabular distribution.*

## 2.7 Why Validate G4STORK?

G4STORK is still a relatively young code and thus does not have the decades of case by case validation that MCNP has. So even though G4STORK has the potential to become a valuable tool for modeling accident scenarios and other transient case, it will not be used by researchers unless they trust the results it produces. So the validation of G4STORK is top priority for the advancement of the code. The goal of validating G4STORK wasn't just to ensure that using the same assumptions that G4STORK would produce the same answers as a trusted code, we also wanted to show that using better assumptions (G4STORK's default minimal assumptions) G4STORK would be able to produce more accurate results.

The SCWR was chosen as the model that would be used to benchmark G4STORK. Part of the reason that the SCWR was selected was simply because our research group was being funded on the condition that we investigate it. However, that was not the only

reason. The high $K_{eff}$ of the SCWR lattice cell, made it ideal for testing the cost in accuracy of using the generational criticality method instead of the dynamic criticality method. Another reason that the SCWR was selected, was because of the large amount of results available for us to compare those produced by G4STORK against (due to the benchmark paid for by CNL).

Since there was only enough time for G4STORK to be benchmarked against one code in such a detailed fashion, selecting the appropriate code was crucial. In order for a comparison of the methodology to be possible, the selected software had to be Monte Carlo. Since at the end of this comparison, the validity of G4STORK would depend on the reputation of the code chosen, the second most important trait of the candidate software was its notoriety (how trusted and popular is it in the scientific community). The third most important characteristic of the chosen software would be its similarity to G4STORK (similarity of its assumptions and methodology). The more in common the two codes had the less adjustments that would have to be made to G4STORK to make it so. MCNP6 was judged as the code that filled these requirements the best. While some of its methodology does differ from that of G4STORK, most notably the calculation of $K_{eff}$, MCNP6 is one of the most widely used and respected codes in the nuclear physics community. Both MCNP and G4STORK primarily use the ENDF libraries, which limits any differences between the physics methodology of both codes, and allows the data sets to be manipulated and converted.

## 2.8 Methods for Comparing Physics Models

In order to compare G4STORK to MCNP6.1, we needed a method for analyzing the difference between the NeutronHP models of the two programs.

### 2.8.1 Results Storage Format

However, before a scheme for analyzing the results of the two different NeutronHP models could be determined, a storage format had to be chosen for the results. The main criterion for choosing the format are as follows. The chosen format of the results had to be optimized for the speed at which it could be analyzed by the computer, but still have obvious physical meaning to the reader. The chosen format also had to be capable of accurately representing the energy, and multiplicity of the outgoing neutrons as functions of incoming neutron energy (since this was of particular interest to us). The chosen format had to allow for easy visual comparisons between different isotopes and reactions for the same outgoing neutron property.

Based off this criterion, it was decided that the outgoing neutrons (the results) would be recorded into two dimensional histograms, indexed by the incoming neutron energy, and an outgoing neutron property. The bin widths for the outgoing neutron properties were kept constant for all isotopes and reactions to allow for faster analysis of the data and simpler visual comparisons. The total number of bins used in each histogram was also

kept constant to ensure that the resolution of the resulting images would be suitable for viewing. This type of representation of the results is in line with what other researchers (including SLAC [19]) have used to validate or compare the NeutronHP model of GEANT4.

### 2.8.2 Methods for Measuring the Statistical Distance

The primary purpose of analyzing the results was to quantify the differences in the outgoing neutron property distributions created by the different NeutronHP models, when the same target isotope, neutron interaction, and incoming neutron energy were used. To do this, a method for quantifying the statistical distance between the outgoing neutron property distributions (stored in each row (incoming neutron energy) of the recorded two dimensional histograms) had to be chosen. Two statistical methods commonly used for this purpose are the Pearson's chi-squared test, and the two-sample Kolmogorov–Smirnov test. The first step for all three of these test is to generate a number (we will refer to as Y) that represents the difference between the two distributions. Then Y is passed into the relevant cumulative probability distribution function, along with the number of data points (bins), to determine the probability of the difference between the two datasets being less than Y (this is equal to one minus the p-value).

#### *2.8.2.1 Pearson's Chi-Squared Test*

The Pearson's chi-squared test quantitatively assess the difference of two distributions using Equation 2.59.

$$x^2 = \sum_{i=1}^{n} (O_i - E_i)^2 / E_i$$

*Equation 2.59: Shows the relationship between the generated difference ($x^2$), the observed dataset ($O_i$), and the expected dataset ($E_i$). Note: 'n' is equal to the number of points in both datasets.*

The cumulative distribution function used to determine the probability of the difference being less than $x^2$ for this method, is shown in Equation 2.60.

$$F(x,k) = \frac{\gamma\left(\frac{k}{2}, \frac{x}{2}\right)}{\Gamma\left(\frac{k}{2}\right)}$$

*Equation 2.60: Shows the relationship between the probability of the difference being less than $x^2$ ($F(x,k)$), the square root of the generated difference (x), and the degrees of freedom (k=n-1). Note: $\gamma\left(\frac{k}{2}, \frac{x}{2}\right)$ is the lower incomplete gamma function, and $\Gamma\left(\frac{k}{2}\right)$ is the upper incomplete gamma function.*

By squaring the difference of the bin values, Pearson's chi-squared test gives more weight to large differences in a few bins then small differences in many bins. This behavior is important because, it minimizes the effect of small variations in the data caused by random sampling. Since the squared difference is divided by the expected value, the

expected dataset cannot contain any zeros. Unfortunately, since the bin widths of the outgoing neutron properties are fixed for all incoming neutron energies; we cannot enforce that there will not be bins containing zeros in the expected dataset. Thus Pearson's chi-squared test is unsuitable for our purposes.

#### 2.8.2.2 The Kolmogorov–Smirnov Test

The Kolmogorov–Smirnov test calculates the difference of two distributions using Equation 2.61.

$$D = \sup(|O_i - E_i|)$$

*Equation 2.61: Shows the relationship between the generated difference ($D$), the observed dataset ($O_i$), and the expected dataset ($E_i$). Note: sup() is the supremum function which essentially finds the maximum value in the inputted dataset..*

The cumulative distribution function used to determine the probability of the difference being less than $D$ for this method, is shown in Equation 2.62.

$$F\left(x = D \times \sqrt{n}\right) = \frac{\sqrt{2\pi}}{x} \sum_{i=1}^{\infty} e^{-(2i-1)^2\pi^2/(8x^2)}$$

*Equation 2.62: Shows the relationship between the probability of the difference being less than D ($F(x)$), the generated difference (D), and the number of bins (n).*

By simply taking the difference of the two datasets, the Kolmogorov–Smirnov test avoids errors caused by dividing by zero, and is efficient to calculate. However, the Kolmogorov–Smirnov test only takes the maximum difference between any two bins in the entire distribution into account. Which means in our case that it unfairly weights large but rare discrepancies caused by physical effects (such as energy boundaries) or numeric imprecision. Thus Pearson's chi-squared test is unsuitable for our purposes.

#### 2.8.2.3 Frobenius Normalization Method

Instead a less commonly used method of taking the Frobenius norm of the difference matrix (described by Equation 2.63) was implemented. Note, that each row of both the observed dataset ($O_{ij}$), and the expected dataset ($E_{ij}$), are normalized to one.

$$D = \frac{\sqrt{\sum_{i=1}^{n} \sum_{j=1}^{m} \left(O_{ij} - E_{ij}\right)^2}}{n}$$

*Equation 2.63: Shows the relationship between the generated difference D, the observed dataset ($O_{ij}$), and the expected dataset ($E_{ij}$). Note: 'n' is equal to the number of rows (incoming neutron energies) in both datasets.*

This method has similar properties as Pearson's chi-squared test discussed earlier, where large differences are weighted more than small differences. Like the Kolmogorov–

Smirnov test, it is capable of handling datasets with zeros in it. It also quantifies the difference of the entire two dimensional histograms, instead each row individually like the previous methods. The main drawback of this approach was that it does not provide any criteria that would allow us to quantitatively determine whether the two distributions are the same. To compensate for this, visualizations of the histograms that were determined to differ the most were created, allowing the user to visually assess whether the distributions differed beyond a reasonable limit. The effect of minor discrepancies in the shape of the outgoing neutron property distributions, was found to have an insignificant effect on the reactor coefficients being compared. The absence of having a mathematical framework to explicitly prove whether two datasets came from the same distribution was deemed acceptable for the purposes of this thesis.

# 3 Contribution and Methodology

In this section of the report, we will cover the methodology used to validate G4STORK, and model the SCWR reactor. In particular, we will cover the software created to meet the goals of this project, how the SCWR was modelled, and how G4STORK was validated.

## 3.1 Improving the Capabilities of G4STORK

Several extensions had to be programmed into G4STORK, in order for us to model the SCWR reactor, and validate G4STORK. In this section we will discuss the most important extensions made to G4STORK for this project.

### 3.1.1 Custom Boundary Conditions

The SCWR lattice cell model relies upon quarter symmetry as a way of improving the efficiency. For this to be physically accurate, a periodic boundary condition which rotates the position and velocity of the neutron, or a reflective boundary condition was needed. G4STORK only allowed for all boundaries to be either, a zero boundary condition (all particles that reach this boundary are killed) or a periodic boundary condition, which moved neutrons from one boundary to the opposite boundary. In order to allow G4STORK meet the needs of a larger variety of models, new boundary conditions were implemented along with a new structure for implementing new boundary conditions. The previous periodic boundary was updated so that the user could specify which boundaries would be linked together (how the neutron position and velocity will be transformed), and a new reflective boundary condition was implemented in G4STORK as well. G4STORK was also modified so that the user can select which surfaces followed which boundary conditions (some surfaces could be periodic, some could be reflective and some could be a zero boundary condition). Finally, if the above options were not sufficient to cover the needs of the user, by simply adding the 3D transform matrix required by the user into the provided template class, the user can easily implement a custom boundary condition.

### 3.1.2 Uniform Source Distribution

G4STORK does not start collecting results until the neutron population has met the convergence criteria. This means that the faster the neutron population meets this criterion the faster the user is able to get results. To make the neutron population converge as fast as possible the user must define an initial neutron population that is close to that of the converged population. While G4STORK already has some techniques for expediting this process (using the converged source distribution from a previous simulation), the user was only allowed to select a point source or a uniform spatial distribution throughout the entire geometry. In order to improve the performance of the SCWR model, the uniform source distribution was updated so that the user can specify the region over which the uniform source distribution will occur.

### 3.1.3 Criticality Calculation Options

While G4STORK uses the dynamic criticality method to calculate criticality, many other codes use the generational or the alpha eigenvalue criticality method [1]. For instance, MCNP, the code against which we wish to validate G4STORK, uses the generational criticality method. The generational criticality method is only accurate (equivalent to the dynamic criticality method) when $K_{\text{eff}}$ is near one (as explained here, 2.6.1.2). Since this is not the case in a lattice cell calculation, the generational criticality method had to be added to G4STORK in order for the comparison between the two codes to proceed.

### 3.1.4 Pre-Doppler Broadening

By default, G4STORK uses an on the fly Doppler broadening algorithm (explained here, 2.6.2). Unfortunately, the SCWR design uses very high temperatures, which means that a lot of the simulation time is spent in the on the fly Doppler broadening algorithm. To fix this problem, an interface was created with in G4STORK that would allow the use of cross-section data at temperatures higher than 0K (the default G4NDL temperature). By using temperatures equal to (or close to) those of the materials used in the SCWR geometry, the on the fly Doppler broadening algorithm could be avoided, dramatically improving the speed of G4STORK. To create these higher temperature cross-section data files, a program called dopplerbroadpara (described in 3.2.3) was used.

## 3.2 Extending the Softwarre Complement of G4STORK

In order for us to model the SCWR reactor, and validate G4STORK, many new programs had to be created. In this section we will discuss the most important programs created for the purpose of this project.

### 3.2.1 Comparing Geometries

During the creation of a new geometry in G4STORK, much of the coder's time is spent creating the material list and ensuring that it is correct. This is a very tedious and error prone process which limits the speed at which new cases can be modelled in G4STORK. To help speed up this process, the ExtractMaterialComposition program was created. This program works by reading in a geometry source and header file in the GEANT4 format

(same as G4STORK), extracting all of the relevant material data, and writing the extracted data to a text file. Using the created text file allows the user to ignore unnecessary data, and specify the data formats (ex: weight% vs abundance). This makes it easier for the user to compare the material information contained by the geometry file, to that of the reference manual which the geometry was built from.

Additionally, the program GEANT4MCNPMatComp was created to automatically compare the material composition of a GEANT4 geometry file to a geometry contained in an MCNP input file. The user must first generate a material composition file from the G4STORK geometry file using ExtractMaterialComposition, and another from the MCNP input file using ExtractMCNPMaterialComp. Then the user passes these two files to GEANT4MCNPMatComp program. GEANT4MCNPMatComp compares the material data contained by the two files, and then writes the data from both files, along with the differences between them, to a text file. This program further streamlined the process of validating G4STORK against MCNP.

### 3.2.2 Converting Neutron Data Libraries

One of the most commonly blamed sources for discrepancy between codes is the different neutron interaction libraries used. Even if the data libraries are based off the same data, as is the case with GEANT4 and MCNP, differences can still occur. This is largely due to the differing format and precision with which each library represents the original experimental data. In order to alleviate this as a concern when comparing G4STORK to MCNP, a program called ConvertMCNPToG4NDL was created to convert the MCNP libraries into the GEANT4 format. The output of ConvertMCNPToG4NDL is a series of temperature directories each of them containing directories for every neutron interaction with all the relevant isotope cross-section data and final state data (distributions of the out-going neutrons properties) files inside. Since GEANT4 was only set up to extract the G4NDL library from a particular directory, a new interface had to be created in G4STORK, to allow the user to select the directory containing the final state data (i.e. G4NDL or the converted data) that they wanted G4STORK to use.

### 3.2.3 Creating Higher Temperature Cross-section Data

#### *3.2.3.1 dopplerbroadpara*

dopplerbroadpara was created to pre-Doppler broaden the cross-section data to the material temperatures that are used in the simulation. This allows the simulation to retain the accuracy of using Doppler broadened cross-section data, without the overhead incurred by an on the fly Doppler broadening algorithm. The program works by first extracting a list of the isotope names and temperatures, used in the geometry to be simulated. Then, it takes in this list of isotopes and temperatures, and iteratively assigns the task of broadening the specified isotope data to each of the available processor cores. The algorithm used to broaden the data is the same as the on the fly Doppler broadening

algorithm described in 2.6.2. The broadened data is then stored in directories named after the temperature of the data so that G4STORK can figure out which data to use.

#### *3.2.3.2 InterpCSData*

MCNP offers the neutron cross-section data at 293.6K, 600K, 900K, 1200K, 2500K. In order to achieve temperatures in between these values, it was common practice to linearly interpolate the cross-section distribution from the two cross-section files with the closest temperature. In order for G4STORK to better match the methodology of MCNP simulations using this technique, the program InterpCSData was created. InterCSData works the exact same way as dopplerbroadpara (described here 3.2.3.1), except that new cross-section data is interpreted from the two cross-section files with the closest temperatures.

## 3.3 Modelling the SCWR in G4STORK

In this section we will outline all of the simplifications and assumptions, necessary for modeling the SCWR.

### 3.3.1 Geometry Simplifications

The point of the SCWR model, was to measure changes in the reactivity, along a central fuel channel, for various coolant cases. By choosing a fuel channel near the center of the core, periodic boundary conditions (zero neutrons escaping the core) could be assumed, simplifying the problem geometry to just one fuel channel. Fluctuations in the reactivity along the fuel assembly were expected, because of the changing temperature and density of the fuel channel components with height. Unfortunately, most reactor physics software require that materials have uniform properties. To accommodate this issue, the fuel channel was split up into 50cm chunks (lattice cells) along the z-axis, with each material being given its average temperature and density within the chunk, as a constant property. By taking advantage of the rotational symmetry of the SCWR lattice cells, a fraction of the lattice cell could be used to represent the whole cell, and in the case of G4STORK only ¼ of each lattice cell was simulated as can be seen in Figure 3.1. For the purpose of the validation of G4STORK against MCNP, only the lattice cell at 1.75m from the bottom of the fuel assembly was considered.

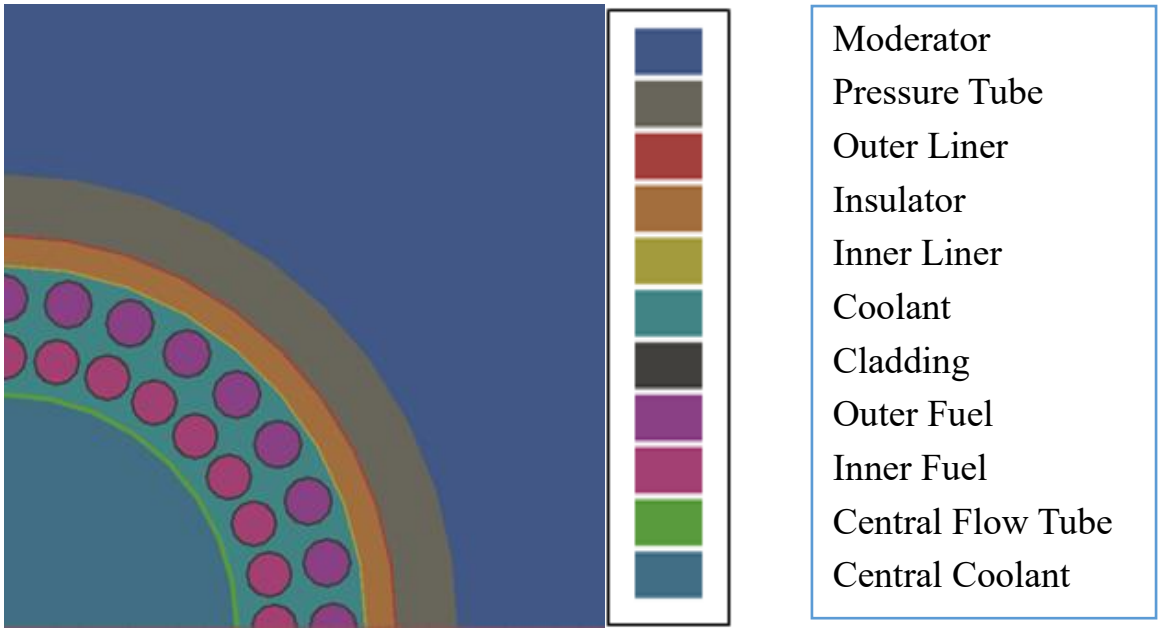


*Figure 3.1: Shows the quarter lattice cell simulated using G4STORK*

### 3.3.2 Material Composition Check

Small deviations in the isotopic composition of materials between model implementations is a common source for discrepancies between simulations. This can be caused by differing level of data input precision by the user, the lack of certain isotopes for certain codes, differing definitions of natural abundance, or isotope mass internal to the code. To ensure that this was not the case for the comparison of MCNP6 and G4STORK, the GEANT4MCNPMatComp program (described in 3.2.1) was used to pinpoint any differences between isotopic composition of the MCNP and G4STORK geometries so that they could be corrected.

### 3.3.3 Removal of Doppler Broadening

Since the SCWR uses materials with very large temperatures, on the fly Doppler broadening takes up a large percentage of the simulation time. Using the dopplerbroadpara code (described in 3.2.3.1) high temperature cross-section data files were created for each unique isotope and temperature combination in the reactor. By passing these files in through the newly created interface (described in 3.1.4), we were able to achieve a 10 times speed up with the code. This step was essential for us to be able to run the SCWR model with the required accuracy, in the available time frame.

### 3.3.4 Simulation Parameters

Whenever a new model is created within G4STORK, various simulation settings have to be adjusted, for the user to get the best compromise of speed and accuracy. The simulation parameters used for the initial modelling of the SCWR lattice cell are shown in Table 3.1.

| Simulation Parameters | Value |
|---|---|
| Lattice Cell Height | 1.75m |
| $K_{eff}$ calculation method | dynamic criticality method |
| # of time steps | 100 |
| Minimum convergence steps | 25 |
| Time step duration | 100μs |
| # of neutrons per time step | 300000 |
| Initial neutron source position | Uniform through outer coolant and fuel |
| Initial neutron energy distribution | Gaussian distribution centered on 2.0MeV, with a standard deviation of 0.5MeV |
| Delayed neutron creation | Instantaneous (created with the prompt neutrons) |
| Boundary conditions | Reflective (equivalent to a rotational periodic condition) |
| Neutron reaction library | G4NDL (the GEANT4 default mainly built off ENDF)GEANT4MCNP |
| Cross-section file library | User generated high temperature files |

*Table 3.1: Lists the run parameters used for the initial simulation of the SCWR lattice cell*

The initial source distribution was chosen to be representative of a new fission generation to reduce the number of time steps needed for the neutron population to converge. The delayed neutrons were set to be created immediately following the fission process with the prompt neutrons. This approximation significantly reduced the amount of simulation time by eliminating the time required to build up a stable precursor population. It also ensures that the delayed neutron population will remain a fixed percentage of the total neutron population (~0.64% in U-235). However, since the cases being simulated are well beyond prompt critical, the delayed neutron population would in reality become a very insignificant proportion of the total neutron population. Thus this is very inaccurate assumption to make for the cases being considered. This approximation is also used by default in MCNP, thus by making it we better matching the methodology of the two codes.

As described earlier (in 3.3.1 and 3.3.3), periodic boundary conditions and pre-Doppler broadened cross-section data were used to significantly reduce the simulation time. In the case of this lattice cell, the $K_{eff}$ is much greater than 1, and the fission generation lifetime

will be equivalent to the prompt neutron lifetime (on the order of microseconds). Thus the time step had to be limited to 100μs to keep the neutron population from exploding. Finally, in order to achieve a $K_{eff}$ with a statistical uncertainty less than 0.5mk, 100 times steps with 300000 neutrons per time step, had to be used.

## 3.4 Validating the G4STORK Model

The initial results of G4STORK were outliers compared to the results of the other codes used in the benchmark study (described here, 1.3). This is partly what gave rise to the need to validate G4STORK. For reasons described in, 2.7, MCNP6 was chosen as the program that G4STORK would be validated against. The neutron high precision model (NeutronHP, empirical model based off ENDF data), implemented in both G4STORK and MCNP (although it is not named in MCNP), covers the energy range of $10^{-5}$eV to $2.0x10^{7}$eV. Neutrons within a nuclear reactor rarely exceed the boundaries of this energy range, so it is fairly accurate to assume that all of the important physics is contained within the NeutronHP model. The thermal scattering model (described here, 2.6.6) is also contained within this energy regime with an energy range of $10^{-5}$eV to 4eV, however its effects on the neutron population are far less significant than the NeutronHP model. For the purpose of this validation, only the NeutronHP model will be compared and validated.

### 3.4.1 Matching the Simulation Settings

#### *3.4.1.1 Modifying the Settings of G4STORK*

The first step in the process of removing the methodological differences between MCNP and G4STORK, was to ensure that the same method to calculate $K_{eff}$, and the same empirical neutron physics models were being used. G4STORK determines $K_{eff}$ by using the dynamic criticality method (described here, 2.6.8.1). This is the most accurate way of determining the $K_{eff}$, since it only assumes that the geometry does not change significantly during a time step. MCNP determines $K_{eff}$ by using the generational criticality method described here, 2.6.1. By using the generational criticality method, MCNP is making the same assumption as G4STORK, and is also assuming that the neutron population is nearly independent of time ($K_{eff}$~1). In a lattice cell calculation such as this, the assumption made by the generational criticality method that the neutron population is near constant with time is very inaccurate. In fact, it was found to be responsible for the majority of the difference between the results produced by G4STORK and MCNP6.1 (compare Table 4.1 to Table 4.2). In order to continue looking for what was causing the remaining discrepancy, between the results of the two codes, the generational criticality method was implemented in G4STORK. Since we wanted to focus our efforts on comparing the NeutronHP model, the high energy (>20MeV) neutron models were turned off in G4STORK.

#### *3.4.1.2 Modifying the Settings of MCNP6*

For incoming neutrons with really low kinetic energy, < 1eV, the electromagnetic forces felt by the target nucleus from the surrounding atoms in a molecule, cannot be ignored.

To take this effect into account, thermal scattering data (described here 2.6.6) must be used. However, our goal is to compare the NeutronHP model by itself (as stated here, 3.4), so the thermal scattering model was taken out of MCNP. For the same reasons, the high energy (>20MeV) neutron models were turned off in MCNP as well. By default, MCNP uses implicit absorption (described here Equation 2.15) instead of analogue absorption. GEANT4 does not implement implicit absorption, because of the cost of accuracy to the result. Consequently, MCNP was set to use analogue absorption. Finally, since GEANT4 does not implement any special treatment for the unresolved resonance regime, such treatments had to be turned off in MCNP. Of all the adjustments to the MCNP simulation listed in this paragraph, the removal of the thermal scattering data was the only modification that significantly affected the $K_{\text{eff}}$.

### 3.4.2 Matching Neutron Interaction Libraries

The second step was to use the MCNP cross-section and final state data in place of the GEANT4 data, when running G4STORK. In order to convert the data from the MCNP format into the GEANT4 format, the program ConvertMCNPToG4NDL (described here, 3.2.2) was used. Once the MCNP data was converted into the GEANT4 format, the cross-section data files were Doppler broadened to the temperatures used in the SCWR geometry using the dopplerbroadpara program (described here, 3.2.3.1). While the cross-section data was relatively easy to transfer, the final state data was organized in a much more complex format. Often distributions and functions implemented in one code would not be present or compatible with the existing distributions in the other code. This can be seen by comparing the distributions used by MCNP6.1 (described here, 2.6.7.4) to those used by the NeutronHP model of GEANT4 (described here, 2.6.8.4). In order to fix these incompatibilities, the source code of GEANT4's NeutronHP had to be modified.

The combination of modifying the source code of GEANT4, along with the large probability of bugs due to the size and complexity of the conversion program, meant that the generated final state files had to be verified. The ValidateFSData program was created to go through all the possible neutron interaction for each isotope, and calculate the results of each isotope/neutron interaction pair, for a fixed incoming neutron energy distribution. The angle, energy, and yield of the outgoing neutrons and photons were stored in separate 2D histograms. The x-axis and y-axis of each histogram, corresponded to the incoming neutron energy, and an outgoing particle property respectively. The count for each bin in the histogram, was normalized by the sum of the counts of the entire row. Two sets of these histograms were produced by ValidateFSData. One was created using the original GEANT4 FS data (G4NDL), and the second from the converted MCNP data. The results obtained using the two data sets were compared, and then stored into text files for post-processing. During the comparison, the difference between histograms produced using G4NDL, and the converted MCNP data, was measured by taking the Frobenius norm of the absolute difference of the two matrices (described here 2.8). The values calculated using the Frobenius norm, were used to rank which isotopes and which

physical properties required the user's attention the most. After ValidateFSData had finished running, an Octave script was used to take in the generated text files, and create 2D histogram plots showing the measured GEANT4, MCNP, and calculated difference data. Some of the images can be viewed here, 6.1.

Since both the GEANT4 and MCNP data sets are based off the same ENDF version, any major discrepancy in the results should have been due to an error in the conversion of the MCNP data. Although this was true most of the time, this was not always the case. Several bugs were identified in the GEANT4 NeutronHP model, and several other major discrepancies in the results were caused by the different data storage formats used by MCNP and GEANT4 (i.e. polynomial versus a few linear regions). A list of the most significant bugs found in the GEANT4 NeutronHP model, is shown below in Table 3.2.

| Location | Bugs |
|---|---|
| G4NeutronHPContAngularPar.cc:262 | When only one photon energy is given by the neutron dataset a bad extraction error occurs. Ensure you provide more than one energy |
| G4NeutronHPPhotonDist.cc:392 | Error in the setting of the photon data, count is not incremented so only the first photon receives a kinetic energy. Add count++ at line 393, and move count++ at line 418 to 417 to fix the error. |
| G4NeutronHPMadlandNixSpectrum.cc | The Madland Nix spectrum is not implemented correctly; it does not give the same shape as the tabular MCNP data. By multiplying the temperature values by 10, a similar mean outgoing energy was achieved however the shape remained discrepant. MCNP |
| G4NDL | Elastic data for yttrium-90 is missing so an isotropic distribution is used which does not match with MCNP/theory at high energies. This was fixed by using the converted data instead of the G4NDL data. MCNPG4NDL |
| G4NeutronHPVector::Merge() | Merge() copies x=0 from both of the vectors being merged, which can cause serious errors. To fix this, Merge() had to be rewritten. |
| G4NeutronHPPartial::GetY() | GetY() does not properly merge the selected distributions, resulting in the duplication of zeros and other errors. To fix this, GetY() had to be rewritten. |
| G4NeutronHPLabAngularEnergy:: Sample() | The methodology used by Sample() is incorrect, because it does not take into account, that a shorter (fewer points) normalized probability vector, will have higher y-values than an equivalent longer probability vector. Thus, when a longer and shorter |

| | |
|---|---|
| | vector are mixed, the shorter vector will be unintentionally given a larger significance. To fix this, Sample() had to be rewritten. |
| G4NeutronHPProduct.cc:48 | The yield is floored when it should be sampled. Fixed this by randomly sampling the yield between the two surrounding integers, with the closest integer getting picked proportionately more often.Flooring |
| G4NeutronHPEnAngCorrelation.cc:101 | When the energy ranges of different distributions overlap, the neutron yield is unintentionally multiplied. This can be fixed by ensuring the energy ranges don't overlap or by forcing GEANT4 to pick a distribution. GEANT4 |

*Table 3.2: Displays bugs found in the high precision neutron physics model during the validation of the converted dataset*

It should be noted that newer versions of GEANT4 now exist and that some of these issues may no longer exist. Using the converted MCNP cross-section and final state data, the results of the G4STORK simulation became a little bit closer to those of MCNP. There was still some discrepancy between the results, as can be seen by comparing Table 4.2 to Table 4.3.

### 3.4.3 Matching the Neutron Physics Methodology

The third step was to compare the methodology used in the physics engine of each code, and determine whether there were any discrepancies. To do this two programs were created, the first program was a modified version of MCNP6 called MCNP6PV, and the second was a modified version of ValidateFSData called ValidateMCNPPhysics. MCNP6PV works the same as ValidateFSData (described here, 3.4.2), except it only produces one set of histograms, using the MCNP physics engine, with the original MCNP data. ValidateMCNPPhysics is then used to take in the results of MCNP6PV, and compare them to those produced by the GEANT4 physics engine, using the converted MCNP data. As before, the two sets of histograms were compared, by taking the Frobenius norm of the absolute difference between equivalent histograms produced by the two different programs (described here 2.8). Once the results had been compared and outputted, an Octave script was used to create visualizations of the results with the greatest discrepancy. Some of these visualizations are shown in, 6.2. The most important differences in the physics methodology between the two codes, that were observed during the comparison, are shown below in Table 3.3.

| Physics Affected | How the Physics is Affected |
|---|---|

| | |
|---|---|
| Neutron Yield and Energy | GEANT4 uses a Poisson distribution to sample the prompt, and delayed neutron yield for the fission process. MCNP samples between the two closest integers to the mean yield when creating the daughter neutrons (MCNP uses a separate algorithm when determining the number of neutron daughters for $K_{eff}$, described in 2.6.7.3.1.1). The average number of neutrons produced by the two different methods will be roughly the same. However, the energy spectrum will be different, since the law of conservation of energy demands that a larger yield of neutrons must have a lower average energy. |
| Neutron Energy and Angle | GEANT4 strictly enforces that the outcome of any reaction abides by the laws of conservation of energy and momentum, MCNP does not. |
| Neutron Yield | MCNP floors the neutron yield for inelastic processes, ensuring that the yield is at least one. GEANT4 also floors the yield but it does not ensure that the yield is at least one. Both of these approaches are incorrect, since a non-integer yield indicates that it should be sampled not floored. FlooringFlooring |
| Cross-section | MCNP modifies the elastic cross section in function neutron_elastic_xsec, if the incoming neutron energy is within a thermal boundary threshold of the target isotope. GEANT4 does not. |
| Neutron Energy | An error was found in the implementation of energy distribution laws 4, 44, and 61 in MCNP. The outgoing neutron energy was not properly sampled from the low incoming neutron energy spectrum.MCNP |

*Table 3.3: Shows what part of the physics was affected by the differences in methodology between MCNP and G4STORK, and how it is affected*

After adjusting the GEANT4 physics engine to match the MCNP physics engine, G4STORK produced the results show in Table 4.4.

### 3.4.4 Examining the Remaining Sources of Discrepancy

#### *3.4.4.1 Examining the Source Code*

Up to this point we have assured that MCNP and G4STORK are using the same geometry, simulation settings, neutron interaction models, and neutron interaction data. So if the rest of both programs has been written correctly, then we would expect them to

produce the same results after the previous step. This can clearly be seen to not be true when one analyzes Table 4.4. Thus the final step of the validation, was to systematically examine the methodology surrounding the NeutronHP model in both programs, and identify flaws/differences that could cause the observed discrepancy in the results. First the NeutronHP algorithms, written in the source code of G4STORK and MCNP, were examined. During this process, a few differences in the methodology between G4STORK and MCNP were discovered. Most of the differing MCNP methodology could not be easily implemented in GEANT4. Instead, the GEANT4 methodology was implemented in MCNP6 (which we were trying to avoid doing). A list of all the issues that were identified, along with which code was modified to match the other, is shown below in Table 3.4.

| Physics Affected | How the Physics is Affected | Changed Code |
|---|---|---|
| Cross-section | MCNP ignores certain inelastic reactions when it takes in the neutron interaction data for oxygen-16, GEANT4 does not. | G4STORK |
| Neutron Energy | Unlike GEANT4, for KCODE simulations (MCNP), the type of fission reaction is not sampled. Instead the fission reaction with the lowest MT that is present (18 or 19) is always used. MCNP | MCNP |
| $K_{eff}$ | During a KCODE calculation, the sampled yield is not used for calculating $K_{eff}$. Instead, the average fission yield is multiplied by the fission cross-section, divided by the total cross-section, and then added to the survivor sum every time any interaction occurs (even if it is not fission). This works as an effective variance reduction technique for $K_{eff}$, but it also ensures that the generational criticality method must be used. | MCNP |
| Neutron Energy and Yield | MCNP only samples the target velocity for elastic and inelastic scattering interactions. | MCNP |

*Table 3.4: Lists the differences in the physics methodology between G4STORK and MCNP, identified by reading through the source code of both programs*

#### 3.4.4.2 Toggling the Methodology

After addressing all of these potential causes of discrepancies, the results shown in Table 4.5 were produced. Since the results of MCNP and G4STORK were still clearly not

within statistical uncertainty, we had to continue checking the remaining methodology for suspect algorithms. Both MCNP and G4STORK are far too big to be analysed in the same way that the transportation and NeutronHP algorithms had been. Instead a coarser method of turning sections of the methodology/code in both programs on and off, and then examining how the results ($K_{eff}$) were affected, was used. The results of this toggling of the methodology can be seen below in Table 3.5. Large differences between the results of G4STORK and MCNP can be seen to occur when the stored cross-section data is used rather than an arbitrary fixed value, and when more than one material is used. Thus the most likely causes for the discrepancy observed in Table 3.5, are differences in the neutron energy distributions, and/or discrepancies in the cross-section distributions.

| Processes, Energy Spectrum | Modified Methodology | G4STORK | MCNP |
|---|---|---|---|
| Fission; fixed 2.0MeV[1]<br>Capture; turned off<br>Inelastic; turned off<br>Elastic; turned off | | 2.896±0.001 | 2.8967±0.0005 |
| Fission; fixed 2.0MeV<br>Capture;<br>Inelastic; fixed 2.0MeV<br>Elastic; fixed 2.0MeV | with isotope speed sampling turned off | 2.049±0.008 | 2.048±0.002 |
| Fission;<br>Capture;<br>Inelastic;<br>Elastic; | with everything made of inner fuel<br>with the cross-sections set to a constant<br>with all materials set to 293.606K | 0.590±0.002 | 0.5893±0.0005 |
| Fission;<br>Capture; | with all materials set to 293.606K<br>with isotope speed sampling turned | 0.7629+0.0004 | 0.7710+0.0006 |

| | | | |
|---|---|---|---|
| Inelastic;<br>Elastic; | off | | |
| Fission;<br>Capture;<br>Inelastic;<br>Elastic; | with all materials set to 293.606K | 1.296±0.002 | 1.302±0.001 |
| Fission;<br>Capture;<br>Inelastic;<br>Elastic; | | 1.264±0.003 | 1.2702±0.0003 |

*Table 3.5: Shows the $K_{eff}$ produced by G4STORK and MCNP, for the SCWR CVR case, as parts of the methodology are turned on and off. Note 1: the output neutron energy of this process is fixed to 2.0MeV. Note 2: the output neutron energy is randomly selected from 1.0eV to 1.0MeV.*

#### 3.4.4.3 Examining the Outgoing Neutron Energy Distribution

The NeutronHP models had been adjusted to produce the same results when used outside of the G4STORK and MCNP programs (discussed here, 3.4.3). However, there was still the possibility that during an actual simulation, parts of the NeutronHP model might behave differently. To test this, incoming and outgoing neutron energy histograms were collected for each isotope and neutron process (elastic, fission, and inelastic), while the G4STORK and MCNP simulations were running. The divergence between the outgoing energy spectra, was then measured by taking the Frobenius norm of the absolute difference between equivalent histograms. Note that all of the individual fission and inelastic interactions had to be lumped into a fission and an inelastic process in order to have enough total samples for each energy distribution to be statistically significant. This also served to focus the observer's attention on the interactions that were dominating a process (being called the most). Unfortunately, this also allowed the cross-section data of the subprocesses (neutron interactions within fission and inelastic scattering), to affect the results as well. The results of this comparison of the outgoing neutron energy spectrums, can be seen in section 6.3.1. During the comparison of the outgoing neutron energy spectrums, discrepancies were found to occur as a result of differences between the thermal isotope speed sampling algorithm.

#### 3.4.4.4 Examining the Extracted Cross Section Data

Next we had to look for discrepancies in the cross-section data being used by the two programs. By converting the MCNP neutron interaction data into the GEANT4 format,

we have already ensured that the cross-section data being passed into the two programs was exactly the same. However, the cross-section data had not yet been closely examined, after it had been extracted and manipulated by MCNP and G4STORK. So a program called CompareCSData was created to compare the original cross-section data (at 293.606K), to the cross-section data (also at 293.606K) produced by the extraction routines of G4STORK and MCNP. When the results of this comparison (shown in, section 6.3.2) were analyzed, minor differences in the extracted cross-section data were found.

#### *3.4.4.5 Final Adjustments to the Methodology*

In the previous two steps, we discovered that both the thermal isotope speed sampling algorithm, and the cross-section extraction routines were sources for discrepancy between the two codes. The problem of the differing thermal isotope speed sampling algorithms was rectified by removing the algorithms from both codes. In order to account for the discrepancy between the extracted cross-section data, the ConvertMCNPToG4NDL program (described here, 3.2.2) was modified. This modification, allowed G4STORK to use cross-section data that had already been modified by MCNP's extraction routine, instead of the raw cross-section data. The results of this final step in the validation, can be seen in section 4.1.5.

## 3.5 Creating the Final Results

In section 3.4, we validated that using the same methods, G4STORK produces results similar to those of MCNP6. Therefore, we expect that when G4STORK uses better assumptions, it should give more accurate results. Deciding which neutron physics dataset is a more accurate representation of the original ENDF data, would require yet another comparison. Although such a comparison would be beneficial for both codes, there was simply not enough time left to conduct it. Instead both datasets were separately used to simulate the final results, with the methodology tuned to be as accurate as possible for the given dataset. The MCNP simulations use the same methodology as the first step of the validation (described here, 3.4.1).

### 3.5.1 Final Results using G4NDL

For the final simulation of the SCWR model using the G4NDL data, the following methodology was used. First the original simulations settings (described here, 3.3) were implemented. Then the thermal scattering model was turned on in G4STORK. Finally, the bugs found during this step of the validation, 3.4.2, were fixed. A comparison of the results produced by G4STORK using this methodology and those produced by MCNP6 using its original methodology can be seen below in Table 4.7.

### 3.5.2 Final Results using the MCNP Data

For the final simulation of the SCWR model using the MCNP neutron interaction data, the following methodology was used. The original simulations settings (described here,

3.3) were implemented. The thermal scattering model was turned off since it seemed to conflict with the converted MCNP data, distorting the results. The MCNP energy distribution models that were missing in GEANT4 (described here 3.4.2), were implemented. Finally, the bugs found in GEANT4, listed in section 3.4.2, were fixed. A comparison of the results produced by G4STORK using this methodology and those produced by MCNP6 using its original methodology can be seen below in Table 4.8.

# 4 Results and Analysis

In this section we will state and analyze the results collected from the validation of G4STORK, and the simulation of the SCWR.

## 4.1 Validation of the G4STORK Physics Engine

The initial results produced by the G4STORK and MCNP6 simulations of the SCWR lattice cell (using the methodology described here, 3.3), are shown in Table 4.1. The results produced by G4STORK can be seen to be both quantitatively and qualitatively different from those generated from MCNP6, by comparing the $K_{inf}$ cases, and the CVR respectively. The difference between the results are clearly beyond statistical uncertainty, indicating that at least one of the simulations is not accurately representing the physics. The following subsections pinpoint exactly what parts of the methodology of each program are causing differences in the results, and inaccurate physics.

| $K_{inf}$ Cases | Fully Cooled | Void Inner Coolant | Void Outer Coolant | Void All Coolant | CVR (mk) |
|---|---|---|---|---|---|
| G4STORK | 1.253±0.0006 | 1.206±0.0002 | 1.258±0.0006 | 1.215±0.0004 | -25.0±0.47 |
| MCNP6.1 | 1.2863±0.0001 | 1.2497±0.0003 | 1.2984±0.0003 | 1.2666±0.0002 | -12.09±0.14 |

*Table 4.1: Initial simulation results before the adjustments for differing methodology were made*

### 4.1.1 The Results of Matching the Simulation Settings

Following the methodology described in 3.4.1, the results displayed in Table 4.2 were produced. The use of the generational criticality method in G4STORK can be seen to have significantly increased the calculated $K_{eff}$, for all coolant cases, bringing the results of G4STORK into much better quantitative and qualitative alignment with the results of MCNP. This was an expected result since the assumptions used by the generational criticality method, tend to overweight the thermal neutrons (as explained here 2.6.1.2). Which, in the case of the SCWR reactor, have a higher production to loss ratio than faster neutrons.

The results of MCNP can be seen to have changed as well. Although it is not shown in Table 4.2, the turning off of the thermal scattering data was the primary cause of the change in the MCNP results (all the other changes combined shifted the results by less than 1mk). The cases with the most coolant present, were also affected the most by the

turning off of the thermal scattering model. This made sense for two reasons. First, the thermal scattering model is only used for neutron energies less than 4eV, and the neutron energy distribution of the cooled cases is lower, because of the moderating effects of the coolant. Second, the thermal scattering model is part of the elastic scattering process, which primarily occurs in the coolant.

| $K_{inf}$ Cases | Fully Cooled | Void inner Coolant | Void Outer Coolant | Void all Coolant | CVR (mk) |
|---|---|---|---|---|---|
| G4STORK | 1.305±0.0006 | 1.2574±0.0006 | 1.313±0.0004 | 1.270±0.0006 | -21.1±0.51 |
| MCNP6.1 | 1.2979±0.0002 | 1.2562±0.0003 | 1.3085±0.0002 | 1.2708±0.0002 | -16.4±0.17 |

*Table 4.2: The generational criticality method has been implemented in G4STORK and in MCNP the thermal scattering data, unresolved resonance, and implicit capture models were turned off.*

### 4.1.2 The Results of Matching Neutron Interaction Libraries

By converting the MCNP ENDF data into the G4NDL format and performing the subsequent steps described here, 3.4.2, the results shown in Table 4.3 were produced. When comparing Table 4.2 to Table 4.3, it can be easily seen that the results of G4STORK have shifted down by ~7mk, so that the fully cooled case result is in near perfect alignment with that of MCNP. Similarly, the once matching results between G4STORK and MCNP for the case when all of the coolant is voided are no longer aligned. About half of the difference between the results shown in Table 4.2 and Table 4.3 was caused by errors in the GEANT4 code listed here Table 3.2. Three of the most important bugs in the GEANT4 code occurred in G4NeutronHPMadlandNixSpectrum.cc, G4NeutronHPLabAngularEnergy:: Sample(), G4NeutronHPEnAngCorrelation.cc:101. The first two bugs, caused large shifts in the outgoing neutron energy spectrum, and the third bug caused the neutron yield to be unintentionally multiplied for some inelastic reactions.

| $K_{inf}$ Cases | Fully Cooled | Void inner Coolant | Void Outer Coolant | Void all Coolant | CVR (mk) |
|---|---|---|---|---|---|
| G4STORK | 1.2978±0.0007 | 1.2499±0.0007 | 1.3057±0.0008 | 1.2639±0.0006 | -20.7±0.61 |
| MCNP6.1 | 1.2979±0.0002 | 1.2562±0.0003 | 1.3085±0.0002 | 1.2708±0.0002 | -16.4±0.17 |

*Table 4.3: Using the same adjustments described in Table 4.2 as well as having G4STORK use the MCNP cross-section and final state data*

The other half of the difference between the results of Table 4.2 and Table 4.3, was caused by the different data storage formats used by GEANT4 and MCNP. For example, MCNP uses a 2D table of incoming neutron energy and outgoing neutron angle to represent elastic scattering, whereas, GEANT4 often uses Legendre polynomials (one gets from solving elastic scattering using Quantum Mechanics, described here, 2.6.8.4.1),

to represent elastic scattering. Even if the format in which the data is stored were the same, the chosen precision (how many data points are used) is often different between programs. So despite the fact that both the GEANT4 and MCNP datasets are derived primarily from the same ENDF data, large differences in accuracy at which they represent the original experimental data exist. Graphical examples of where the data sets differed the most are shown in 6.1.

### 4.1.3 The Results of Matching the Neutron Physics Methodology

The physics methodologies of G4STORK and MCNP were matched and validated following the steps outlined here, 3.4.3. The physics methodology that differed the most, after the adjustments to GEANT4 (listed here Table 3.3) had been made, are shown here, 6.2. They demonstrate that there are no significant differences in the neutron physics methodology remaining between the two programs. Some of the more significant changes to the GEANT4 physics methodology was the flooring of the fission neutron yield, when no delayed data was present, and the removal of inelastic reactions for particular isotopes. The results produced by G4STORK after the modifications had been made are listed below in Table 4.4. The changes to the methodology caused the results of G4STORK to be quantitatively a worse match for the results of MCNP, but qualitatively a better match. The overall shift in the G4STORK results was relatively minor.

| $K_{inf}$ Cases | Fully Cooled | Void inner Coolant | Void Outer Coolant | Void all Coolant | CVR (mk) |
|---|---|---|---|---|---|
| G4STORK | 1.2964±0.0006 | 1.2503±0.0004 | 1.3053±0.0004 | 1.2636±0.0004 | -20.0±0.44 |
| MCNP6.1 | 1.2979±0.0002 | 1.2562±0.0003 | 1.3085±0.0002 | 1.2708±0.0002 | -16.4±0.17 |

*Table 4.4: Same adjustments as Table 4.3 except that the high precision neutron data model used in G4STORK has been adjusted to more closely match the methodology of MCNP, as described in the last paragraph of the methodology*

### 4.1.4 The Results of the Adjustments Made After Examining the Source Code

Following the methodology outlined in 3.4.4.1, the results shown in Table 4.5 were produced. Although the overall shift in the results is not large, they have come quantitatively and qualitatively (difference between CVR) closer together. The majority of the shifts in the MCNP results were caused by turning on thermal isotope speed sampling for all interactions, instead of just for the elastic and inelastic processes. The results for all four coolant cases, can be observed to still differ beyond statistical uncertainty.

| $K_{inf}$ Cases | Fully Cooled | Void inner Coolant | Void Outer Coolant | Void all Coolant | CVR (mk) |
|---|---|---|---|---|---|
| G4STORK | 1.2961±0.0005 | 1.250±0.0004 | 1.3054±0.0004 | 1.2642±0.0004 | -19.5±0.35 |
| MCNP6.1 | 1.2972±0.0005 | 1.2556±0.0004 | 1.3083±0.0004 | 1.2702±0.0003 | -16.4±0.35 |

*Table 4.5: Same adjustments as Table 4.4 except both MCNP6 and G4STORK have been adjusted to more closely match each other's methodology*

### 4.1.5 The Results of the Final Adjustments to the Methodology

Following the methodology outlined in 3.4.4.5, the results shown in Table 4.6 were produced. Significant shifts in the $K_{inf}$ can be seen to occur; These large shifts are primarily due to the removal of the thermal isotope speed sampling algorithms, which significantly reduces the relative probability of fission occurring for thermal neutrons. More importantly, the results can be seen to be significantly closer together, both quantitatively and qualitatively. Except for the fully cooled case, all of the results are almost within statistical uncertainty of each other.

| $K_{inf}$ Cases | Fully Cooled | Void inner Coolant | Void Outer Coolant | Void all Coolant | CVR (mk) |
|---|---|---|---|---|---|
| G4STORK | 0.7132±0.0004 | 0.7799±0.0004 | 0.7089±0.0004 | 0.7722±0.0004 | 107±1 |
| MCNP6.1 | 0.7164±0.0007 | 0.7813±0.0006 | 0.7108±0.0006 | 0.7710±0.0006 | 99±2 |

*Table 4.6: Same adjustments as Table 4.5, except that the thermal isotope speed sampling model has been removed from G4STORK and MCNP6, and the cross-section extraction routines of the two codes have been matched*

### 4.1.6 Hypothesis for the Remaining Discrepancy Between the Results of MCNP and G4STORK

The remaining discrepancy in the results is likely due to the different mathematical models used to describe elastic scattering by the two different codes. The primary difference between the models results from G4STORK uses of relativistic transforms to transfer between reference frames, instead of the Galilean transforms used by MCNP6. Although we had previously matched the neutron property distributions that the neutron interactions sample from, the outgoing energy distribution for elastic scatter could not be explicitly matched, since it is not sampled (it is calculated as a function of the relative incoming neutron energy, the masses of the reactants, and the scattering angle).

The presence of a discrepancy between the elastic scattering models, would also explain why greater discrepancies occur can be seen to occur, in the results of Table 4.6, with increasing moderation. The discrepancy between the elastic scattering models is likely small, otherwise there would have been a visible difference between the two when we compared them in section 6.2. However, since neutrons typically undergo many elastic scattering reactions before being absorbed, even small differences between elastic scattering models can cascade to have a significant effect. Unfortunately, there was not enough time left in this project to test this hypothesis. Thus further work will have to be done to complete the validation of G4STORK.

## 4.2 Final SCWR Results

As discussed in section 3.5, the SCWR was simulated using the G4NDL data, and the converted MCNP neutron interaction data, for the final results. In the following sections we will describe and analyze each set of results created using the different neutron interaction data sets.

### 4.2.1 Using G4NDL Data

Using the methodology described in 3.5.1, the results contained in Table 4.7 were produced. The addition of the thermal scattering model, and the fixes to GEANT4's NeutronHP model, can be seen to have had little effect on the results of G4STORK, when one compares Table 4.1 to Table 4.7. This observation conflicts with the significant change in $K_{eff}$ noticed when the thermal scattering data was turned off in MCNP (observed here, 4.1.1). This difference warrants two possible explanations. The first explanation is, that the thermal scattering models produce different results between programs. Given the large difference of the effects of the different thermal scattering models, and the relatively minor role the models play, there would have to be a vast discrepancy between the models. While this is possible it is not very likely. The second possibility is that the thermal scattering model is significantly affecting the neutron lifetime in the reactor. As explained here, 2.6.1.2, this would affect the generational criticality method (used by MCNP6) differently than a dynamic criticality method (used by G4STORK). This seems like the most plausible explanation for the observed discrepancy.

When examining Table 4.7, a large divergence between the results of MCNP and G4STORK is evident. As discussed before, this is largely caused by the inaccuracy of the generational criticality method used by MCNP6, when simulating highly sub or supercritical cases. The reactivity incurred by the complete voiding of the coolant (the CVR), is calculated to be far more negative in the G4STORK simulation than in the MCNP6 simulation. G4STORK also predicts a significantly smaller positive reactivity increase when the outer coolant of the fuel assembly is voided. In fact, the results of G4STORK indicate that the reactor will behave more desirably during any kind of coolant voiding accident, than any other code used in the comparison (as can be seen by comparing Table 1.2 to Table 4.7). However, all of the other programs in the comparison make the same inaccurate assumption that the neutron population is nearly independent of time. Which means that we expect the results of G4STORK to be a more accurate description of the SCWR lattice than the other codes in the comparison. This is good news for the designers, since our results indicate that their design has better safety characteristics than they would have been otherwise led to believe.

| $K_{inf}$ Cases | Fully Cooled | Void Inner Coolant | Void Outer Coolant | Void All Coolant | CVR (mk) |
|---|---|---|---|---|---|

| | | | | | |
|---|---|---|---|---|---|
| G4STORK | 1.252±0.0006 | 1.207±0.0008 | 1.259±0.0006 | 1.216±0.0004 | -23.6±0.47 |
| MCNP6.1 | 1.2863±0.0001 | 1.2497±0.0003 | 1.2984±0.0003 | 1.2666±0.0002 | -12.09±0.14 |

*Table 4.7: Compares the most accurate results achievable with G4STORK and MCNP, when G4STORK is using the G4NDL data*

### 4.2.2 Using Converted MCNP6 Neutron Interaction Data

Using the methodology described in 3.5.2, the results contained by Table 4.8 were produced. The same large discrepancies between the results of G4STORK and MCNP6.1, observed in section 4.2.1, can be seen to exist in Table 4.8 (refer to section 4.2.1, for further explanation about this phenomenon). However, a significant change in the results can be observed when one compares Table 4.7 to Table 4.8. A systematic drop in the $K_{eff}$ by ~4.5mk for all four reactivity cases can be seen. The CVR stays almost exactly the same indicating that the shift was purely quantitative.

| $K_{inf}$ Cases | Fully Cooled | Void Inner Coolant | Void Outer Coolant | Void All Coolant | CVR (mk) |
|---|---|---|---|---|---|
| G4STORK | 1.247±0.0006 | 1.202±0.0006 | 1.255±0.0006 | 1.212±0.0006 | -23.2±0.56 |
| MCNP6.1 | 1.2863±0.0001 | 1.2497±0.0003 | 1.2984±0.0003 | 1.2666±0.0002 | -12.09±0.14 |

*Table 4.8: Compares the most accurate results achievable with G4STORK and MCNP, when G4STORK is using the Converted MCNP data*

### 4.2.3 Possible Method for Analyzing Reactor Uncertainty

In section 4.2.1, we showed that the thermal scattering data had an insignificant effect on the results of the SCWR simulation, when the dynamic criticality method is being used. Thus, the difference between the results of Table 4.7 and Table 4.8, must be due to the use of the converted MCNP data with additional energy distributions (mentioned here, 3.4.3), instead of the default G4NDL data. This result is expected, since as we discussed here 3.4.2, the two data sets use different types of distributions to represent the data, and different levels of precision. Since neither one of the datasets has been proven to be more accurate than the other in all possible situations, both of them are equally valid to use. Thus, the difference between the results achieved using the two different datasets is a measure of the level of uncertainty that we must expect when modeling the SCWR in this situation.

By continuing to compare the results produced using the two different datasets for many different cases, a generic estimate of what the uncertainty of modeling the SCWR design could be calculated. The generic uncertainty calculated using this method, would not be affected by the differing methodology (physics, assumptions, bugs) present between differing programs. The generic uncertainty would also not be biased by the inaccurate assumptions of the generational criticality method used by the other nuclear reactor

physics programs, in the comparison of the SCWR. By using the suggested method, the only factors that would affect the calculated generic uncertainty would be the reactor design, the statistical uncertainty, and the effects of the different neutron interaction datasets. The effect of statistical uncertainty on the results, will always be present due to the random nature of Monte Carlo programs. The effect of the different neutron interaction datasets will also be an unavoidable factor (because they are all equally valid to use), until one dataset has been proven to be more accurate than the others in every possible situation. Since none of these sources of uncertainty can be eliminated, the generic uncertainty calculated from them would represent the minimum uncertainty achievable by the user when modelling a particular nuclear reactor.

Understanding what the minimum uncertainty is for modelling particular cases using the computational method, could help researchers decide which cases need to be measured experimentally, and which cases don't. Considering that determining the uncertainty involved in modelling the SCWR was CNL's main purpose for funding this cross nuclear reactor physics program comparison, we believe this conclusion to be an important result that warrants further investigation.

# 5 Conclusions

The intentions of this project were to validate the accuracy of G4STORK and compare its results to other codes used to investigate the SCWR lattice cell. By matching the assumptions of G4STORK and MCNP6, we were able to achieve close agreement between results of the two programs when simulating the SCWR lattice cell (shown here, 4.1.4). However, the final results were not quite within statistical uncertainty of each other, so the validation is not complete. It was hypothesized that the remaining discrepancy could be caused by differences in the elastic scattering models between G4STORK and MCNP6.

After implementing its more accurate default methodology, G4STORK was used to simulate the SCWR lattice cell using the G4NDL data and the converted MCNP data. When the results of G4STORK were compared to those of MCNP6, large differences were observed (shown here, 4.2). This difference was proven to be caused by MCNP's inaccurate use of the generational criticality method to model such a highly supercritical system. Based on these findings we conclude that, not only are the results of G4STORK accurate, they can be more accurate than those produced by many of the more commonly used programs for transient cases.

During the comparison of the results obtained when G4STORK is using the G4NDL dataset, and MCNP's neutron interaction library (4.2.2), a new method for estimating the uncertainty of modeling the SCWR was devised. The new method would determine the minimum uncertainty that CNL can achieve when modeling the SCWR using a particular code. This method could help researchers better understand how much uncertainty is

inherent (unavoidable) to their model, and would be less dependent on the specific implementation.

## 5.1 Future Work

There are still many important tasks that must be done before the G4STORK code will be able to meet its original goal (of being a minimal assumption tool for modeling transient scenarios). Some of the most important tasks are listed below.

### 5.1.1 Completing the Validation

Before researchers will trust and hopefully start using the G4STORK code, the validation of G4STORK must be finished. In order for the validation of G4STORK to be complete, the remaining discrepancy between the results of G4STORK and MCNP (when they are using the same methodology) must be eliminated. To do this, the transportation algorithms of MCNP and G4STORK should be compared since this is the most likely cause for the difference.

### 5.1.2 Heat Transfer Model

Although G4STORK allows for the temperature and density of a material to change in steps of time, there is currently no heat transfer model that dictates how it should do so. This stops G4STORK from being able to accurately account for temperature and density feedback during accident scenarios. To solve these inaccuracies, two different approaches for adding a heat transfer model should be taken. The first approach would be to implement a generic framework in G4STORK that the user can build their own heat transfer model from, specifically for the case they are studying. This would allow the user to determine the necessary accuracy of the heat transfer model and optimize the performance of the G4STORK code. The second approach would be to create a generic interface in G4STORK, through which external heat transfer programs could receive and send data (coupling the two programs together). This would allow the user to more accurately model heat transfer within their geometry than the previous approach and require less effort from the user.

### 5.1.3 GPU Acceleration

Currently G4STORK is only able to simulate cases that take place on the order of one tenth of a second in a reasonable timeframe. However, many important physical effects, such as the appearance of delayed neutrons, can take minutes to occur. Thus, for most accident scenarios we would like to study the reactor's behavior over a similar time period. In order for G4STORK to be able to able to simulate cases on the order of minutes, a massive speed up of the code is required. This kind of speed up is possible through the use of GPU parallelism. A high end CPU contains eight processors and runs at ~3.5GHz, a high end GPU contains ~3600 processors and runs at ~1.5GHz. There are certain issues with GPU computing that limit its performance, most notably that every processor within each 32 processor block must be following the same algorithm (but on

different data). Another limitation on GPU performance is the latency incurred by sending data back and forth between the CPU and the GPU. This was most noticed when a group of undergraduate students tried to port frequently used functions within GEANT4 to the GPU. This resulted in GEANT4 running slower due the GPU to CPU latency being larger than the time required to perform the functions on the CPU. If each group of 32 processors was independently responsible for tracking a group of neutrons, only two messages between the GPU and the CPU would be required per time step. Thus the few milliseconds taken up by GPU-CPU latency would be insignificant compared to the minutes to hours taken up by each time step. Since the section of the G4STORK code that would run on the GPU scales very efficiently with an increasing number of processes, a code speed up of roughly 30-80 would be expected with GPU acceleration.

# 6 Appendix

In this section, we will the results obtained from the comparison of the neutron interaction data sets and from the comparison NeutronHP models.

## 6.1 Comparison of the Converted MCNP6 Nuclear Data and the G4NDL Data

The figures below were produced to validate the converted MCNP neutron interaction data against G4NDL (during this step of the methodology, 3.4.2). They show measured properties of the out-going neutrons as functions of incoming neutron energy, for each isotope and process. For each distribution, the results collected using the G4NDL data, the converted MCNP data, are shown separately along with a visualization of the squared difference between the two. In the cases where there is no visible, difference between the two distributions, the difference plot is not shown. One dimensional histograms showing the point (incoming neutron energy) where the outgoing neutron energy distributions differed the most between datasets, are also shown for all process except for elastic scattering. The three sets of images show the distributions that differed the most between codes for the three most important reactions, elastic, fission (combined), and inelastic MT=4 (capture was not included since it doesn't produce any neutrons).

### 6.1.1 Elastic Scattering Comparison

It is easy to see from Figure 6.1 and Figure 6.3, that there is virtually no difference between the shown out-going energy distributions. Since these results were measured to be the most discrepant for the elastic scattering process, we can state that there are no major discrepancies between the G4NDL data, and the converted MCNP neutron interaction data, for this process.

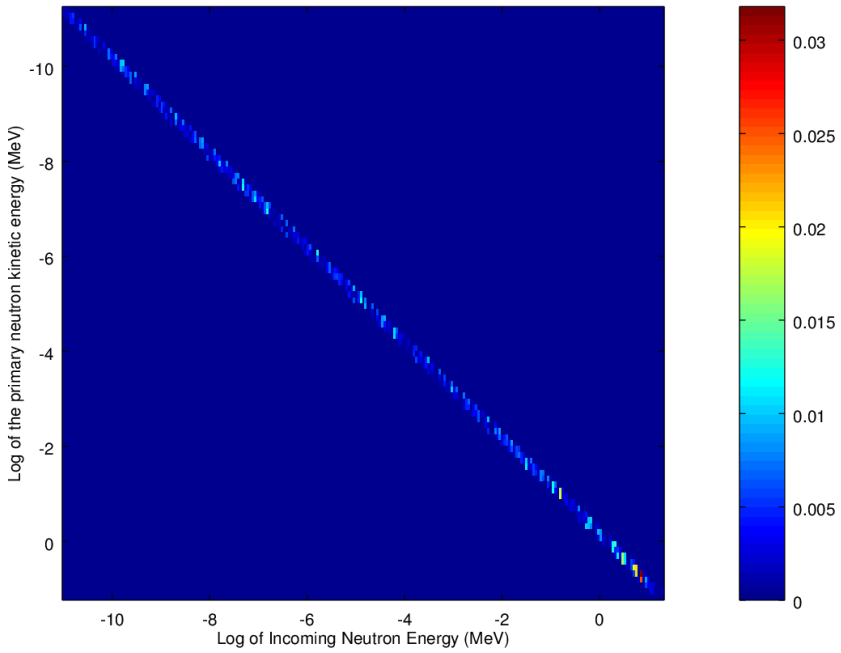


*Figure 6.1: The squared difference between the outgoing neutron energy for the elastic scattering of beryllium-9*

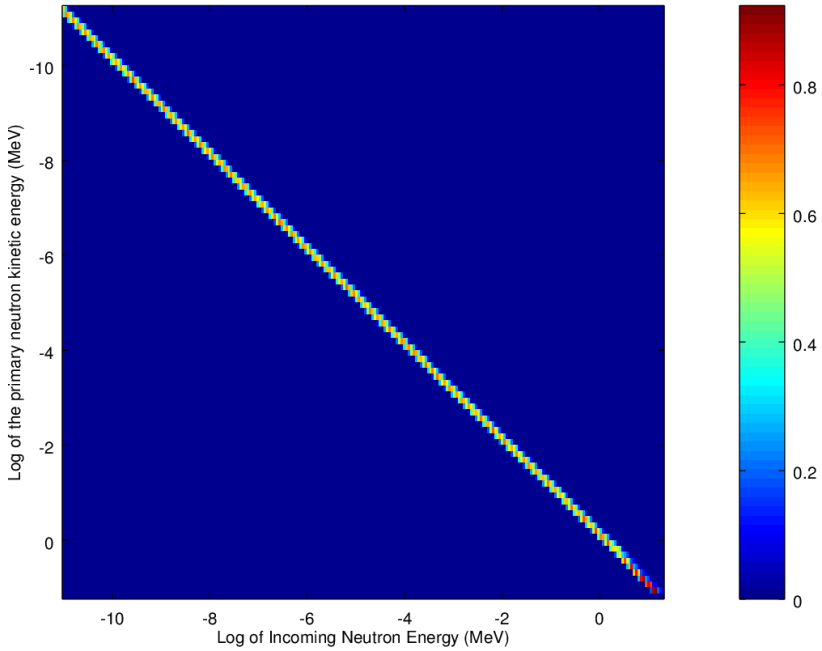


*Figure 6.2: The G4NDL outgoing neutron energy for the elastic scattering of beryllium-9*

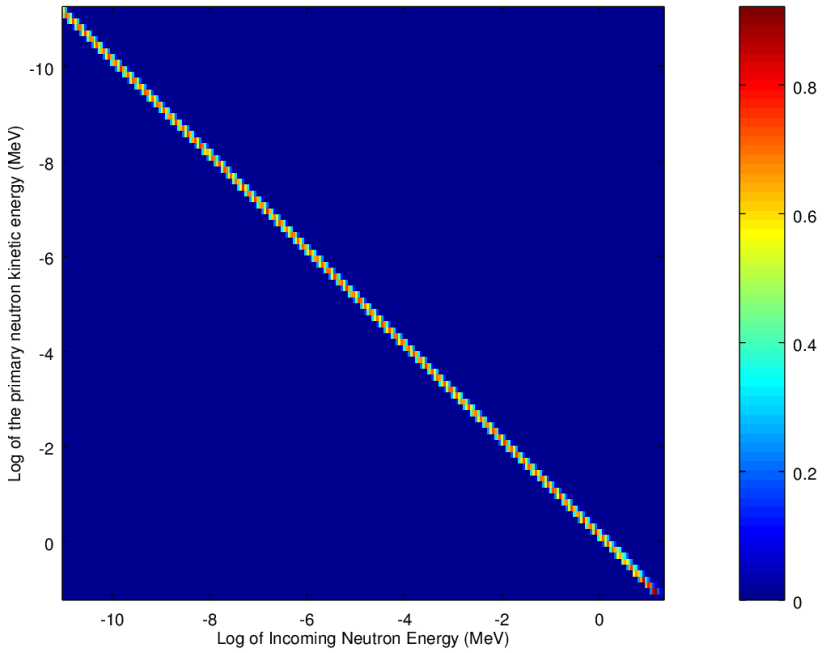


*Figure 6.3: The converted MCNP data outgoing neutron energy for the elastic scattering of beryllium-9*

### 6.1.2 Fission Comparison

By examining Figure 6.4 to Figure 6.13, we can see that the only significant difference for this process between the G4NDL data and the converted MCNP data occurs in the outgoing neutron energy distribution. While small differences in the yield distributions can be seen, these are primarily caused by statistical uncertainty, and not by differing methodology. The observed discrepancies between the energy distributions, is a result of the different types of representations being used by the two data sets, and is not a flaw in the conversion.

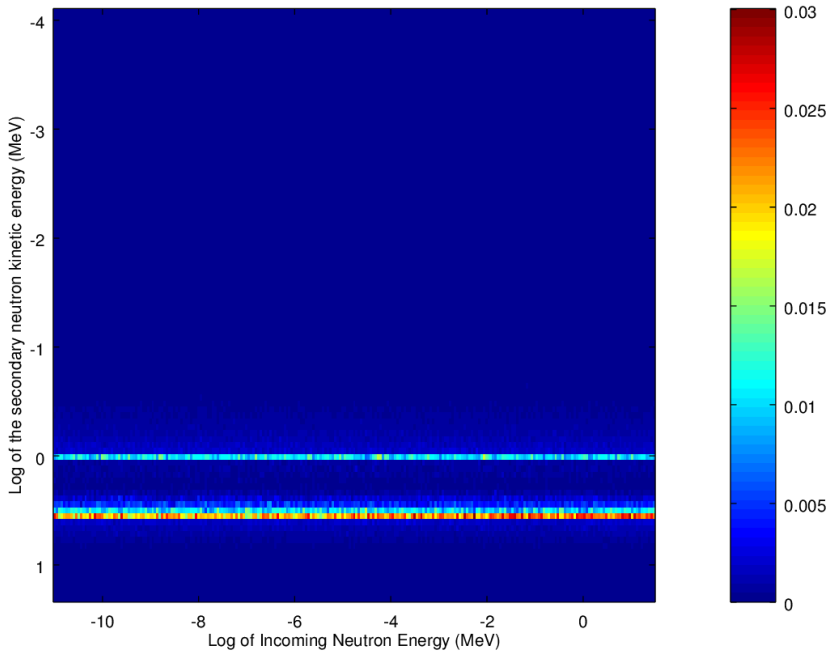


*Figure 6.4: The squared difference between the outgoing neutron energy for the combined fission process of americium-241.*

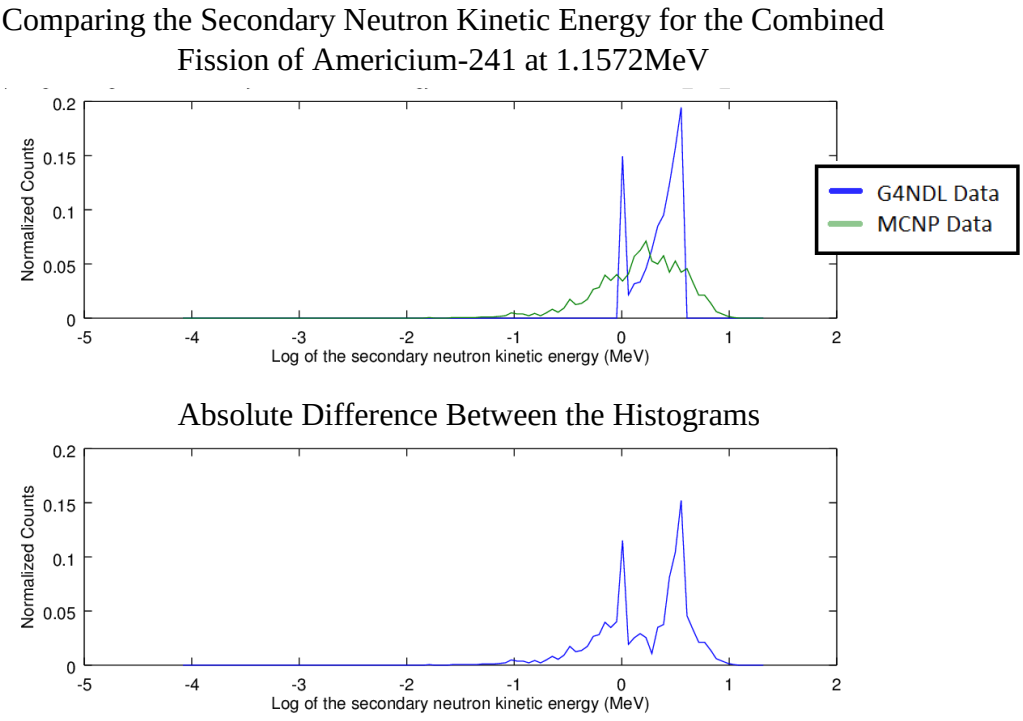


*Figure 6.5: The most discrepant outgoing neutron energy distributions for a fixed incoming neutron energy (column) of the combined fission process of americium-241.*

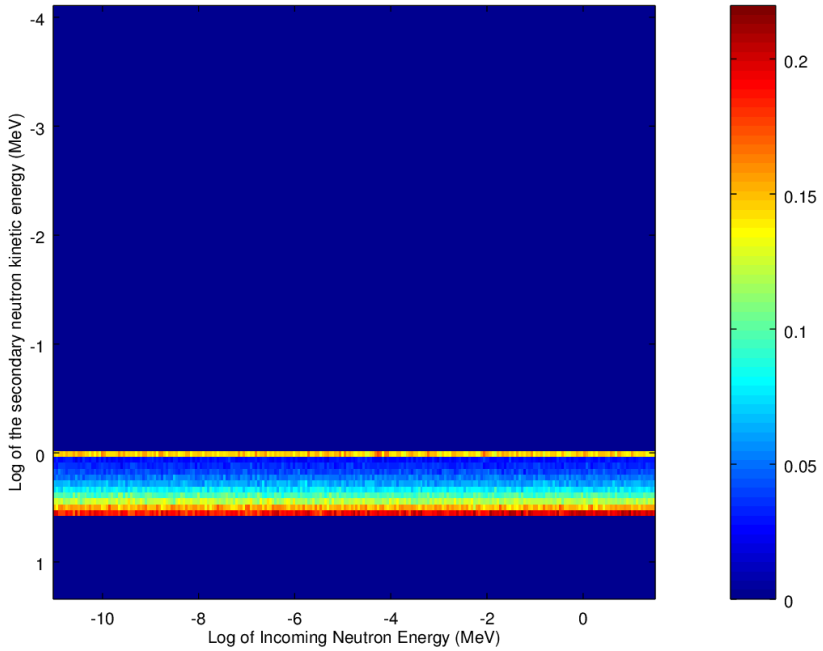


*Figure 6.6: The G4NDL outgoing neutron energy for the combined fission process of americium-241*

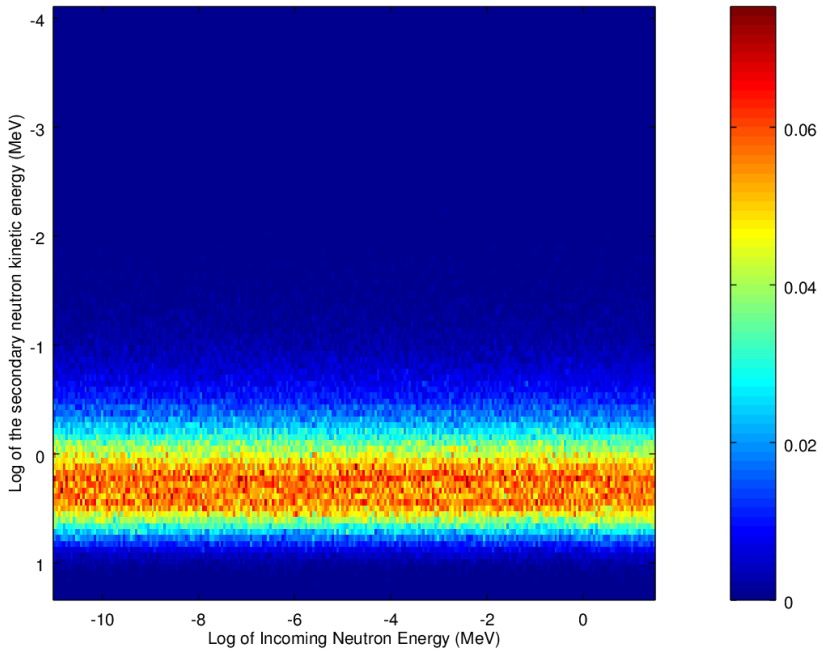


*Figure 6.7: The converted MCNP data outgoing neutron energy for the combined fission process of americium-241*

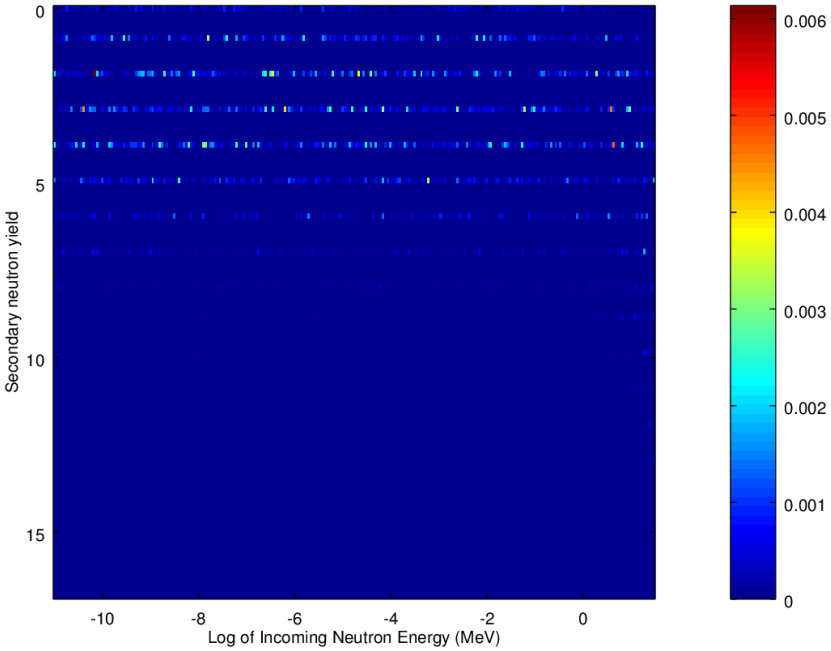

*Figure 6.8: The squared difference between the prompt neutron yield for the combined fission process of americium-241*

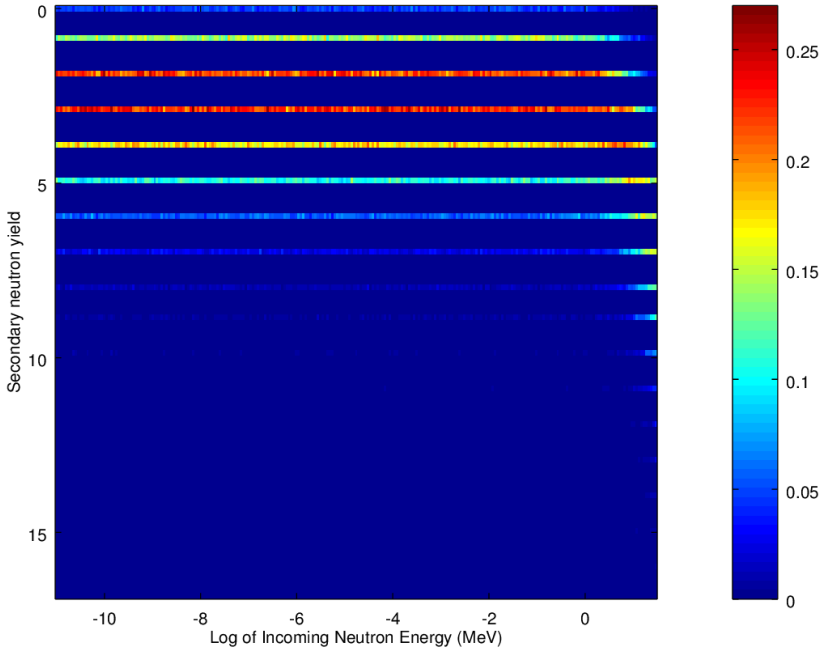


*Figure 6.9: The G4NDL prompt neutron yield for the combined fission process of americium-241*

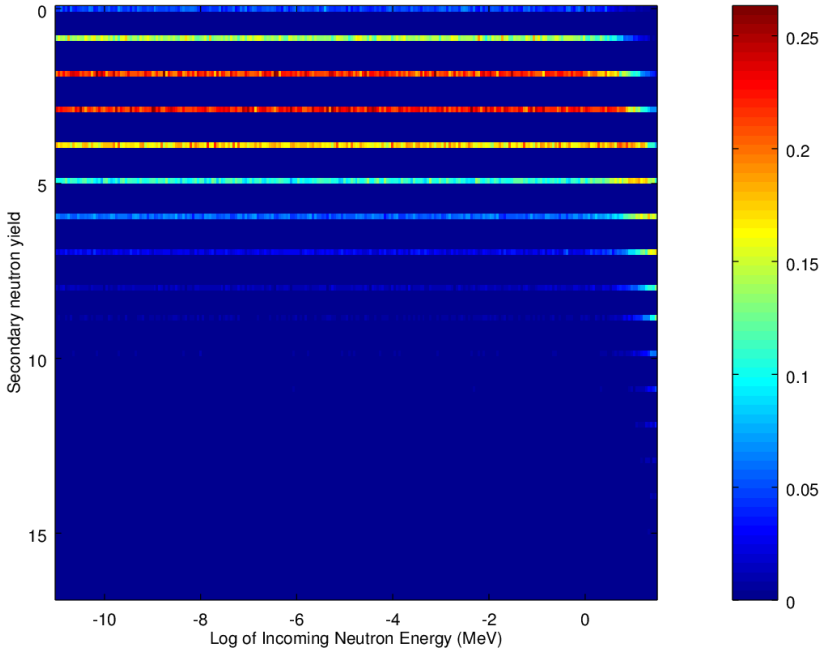

*Figure 6.10: The converted MCNP data prompt neutron yield for the combined fission process of americium-241*

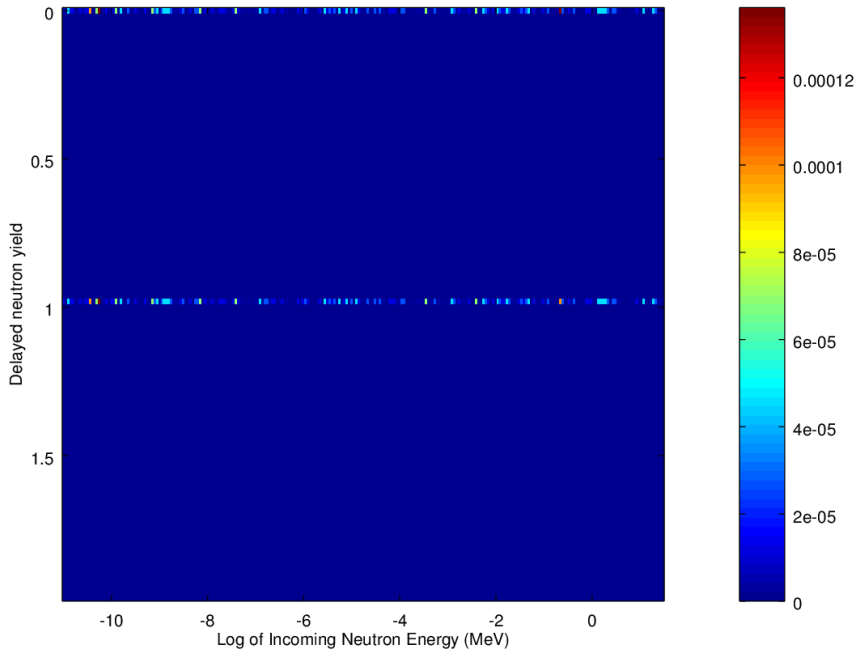


*Figure 6.11: The squared difference between the delayed neutron yield for the combined fission process of americium-241*

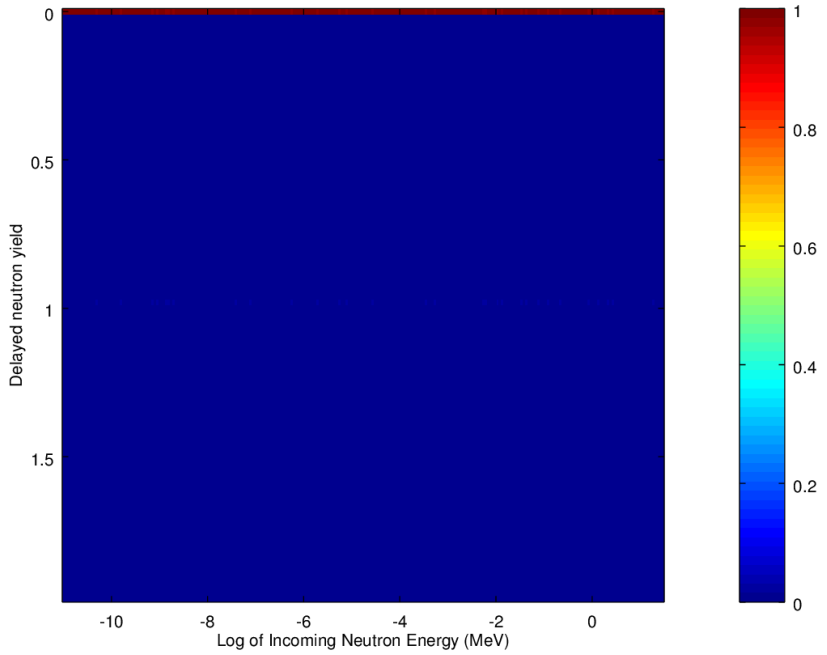


*Figure 6.12: The G4NDL delayed neutron yield for the combined fission process of americium-241*

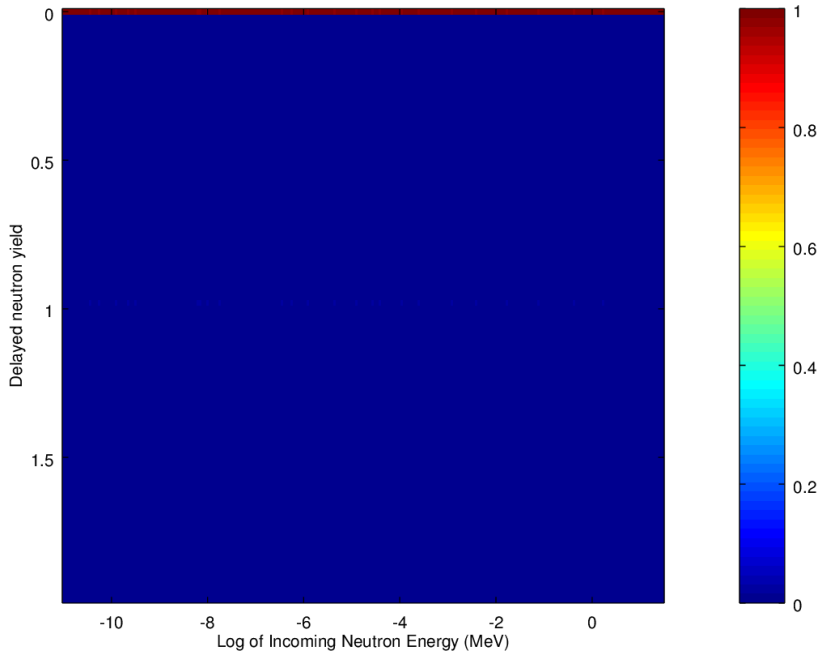


*Figure 6.13: The converted MCNP data delayed neutron yield for the combined fission process of americium-241*

### 6.1.3 Inelastic MT=4 Comparison

By examining Figure 6.14 to Figure 6.17, we can see that the only difference for this process, between the G4NDL data and the converted MCNP data, occurs in the outgoing neutron energy distribution. Again, the observed discrepancies between the energy distributions, is a result of the different datasets and types of representations being used by the two data sets, and is not a flaw in the conversion.

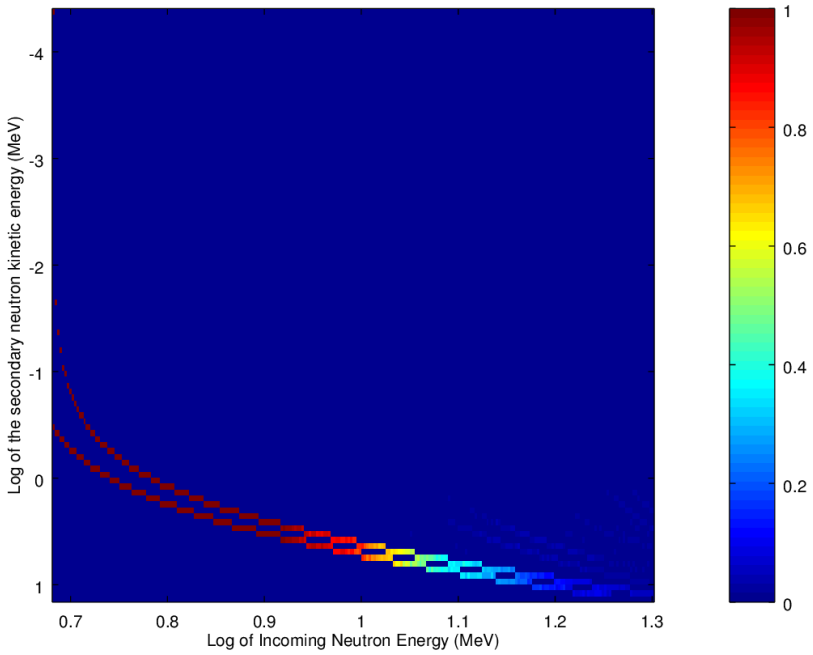


*Figure 6.14: The squared difference between the outgoing neutron energy for the MT=4 inelastic process of natural carbon*

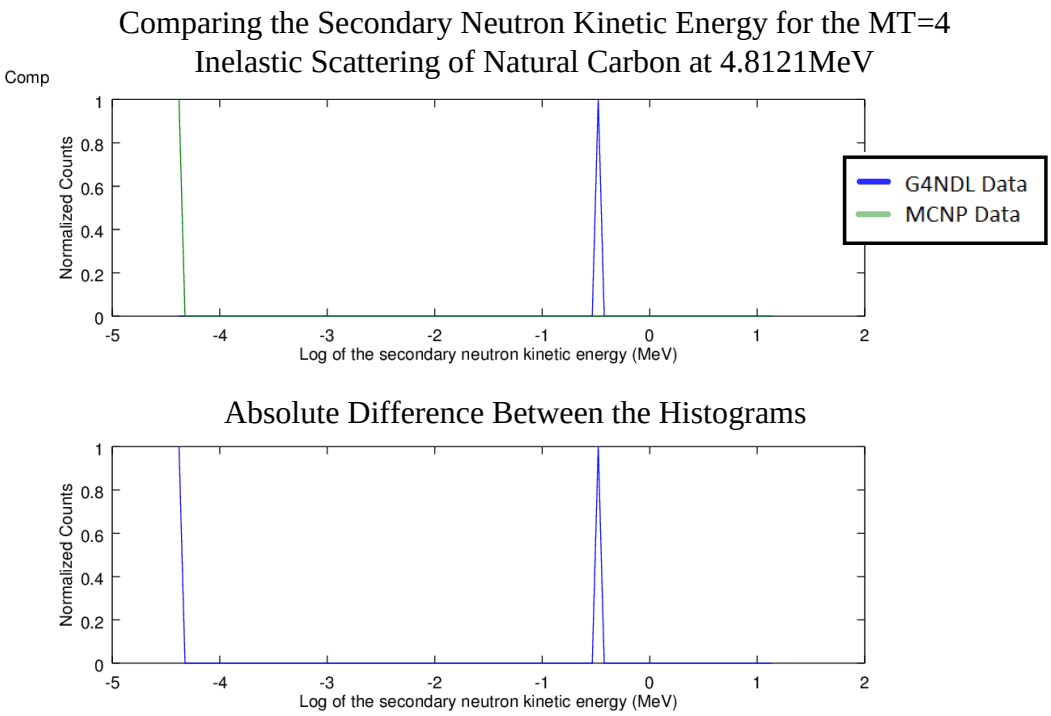


*Figure 6.15: The most discrepant outgoing neutron energy distributions for a fixed incoming neutron energy (column) of the MT=4 inelastic process of natural carbon*

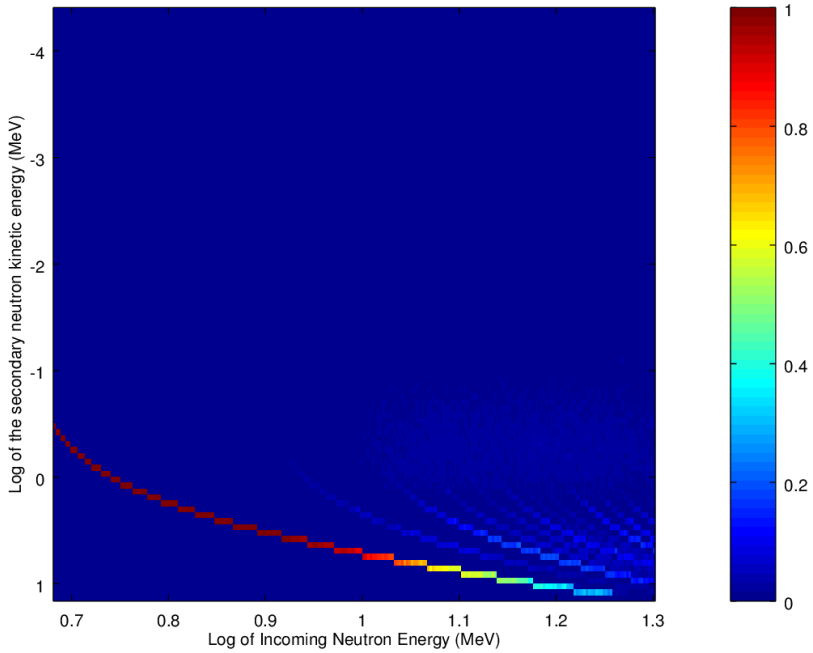


*Figure 6.16: The G4NDL outgoing neutron energy for the MT=4 inelastic process of natural carbon*

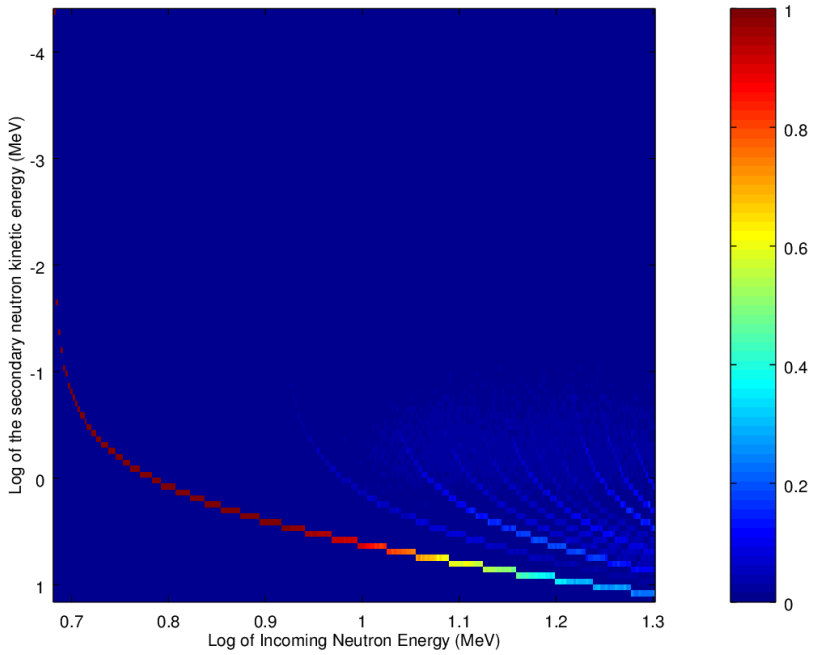


*Figure 6.17: The converted MCNP data outgoing neutron energy for the MT=4 inelastic process of natural carbon*

## 6.2 Comparison of the MCNP6 and GEANT4 Physics Engines

The figures shown below were produced to determine whether the methodology of the NeutronHP model, was the same between MCNP and G4STORK (explained here, 3.4.3). Measured properties of the out-going neutrons are shown as functions of incoming neutron energy, for each isotope and process. For each distribution the data collected from G4STORK, MCNP and the squared difference between the two are shown separately. In the cases where there is no visible difference between the two distributions, the difference plot is not shown. One-dimensional histograms showing the point (incoming neutron energy) where the outgoing neutron energy distributions differed the most between datasets, are also shown for all processes except for elastic scattering. The first three sets of images show the distributions that differed the most between codes for the three most significant reactions: elastic, fission (combined), and inelastic MT=4 (capture was not included since it does not produce any neutrons). The last set of data shows the isotope distributions that differed the most out of any of the possible processes.

### 6.2.1 Elastic Scattering Comparison

It is easy to see that there is almost no difference between the G4STORK and MCNP results shown in Figure 6.18 and Figure 6.19. Since these results were measured to be the most discrepant for the elastic process, we can state that there are no major discrepancies between the physics methodologies for elastic scattering.

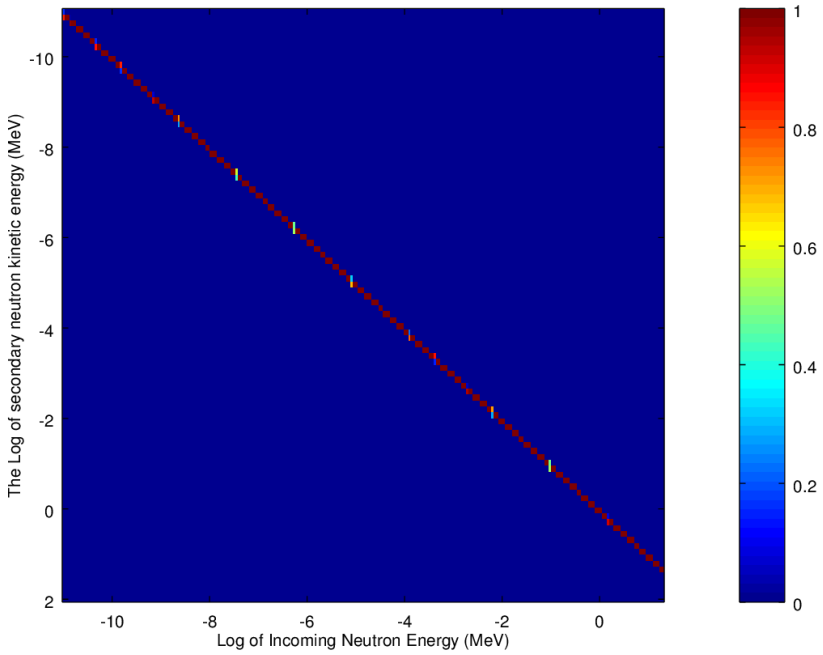


*Figure 6.18: The G4STORK outgoing neutron energy for the elastic scattering of plutonium-242*

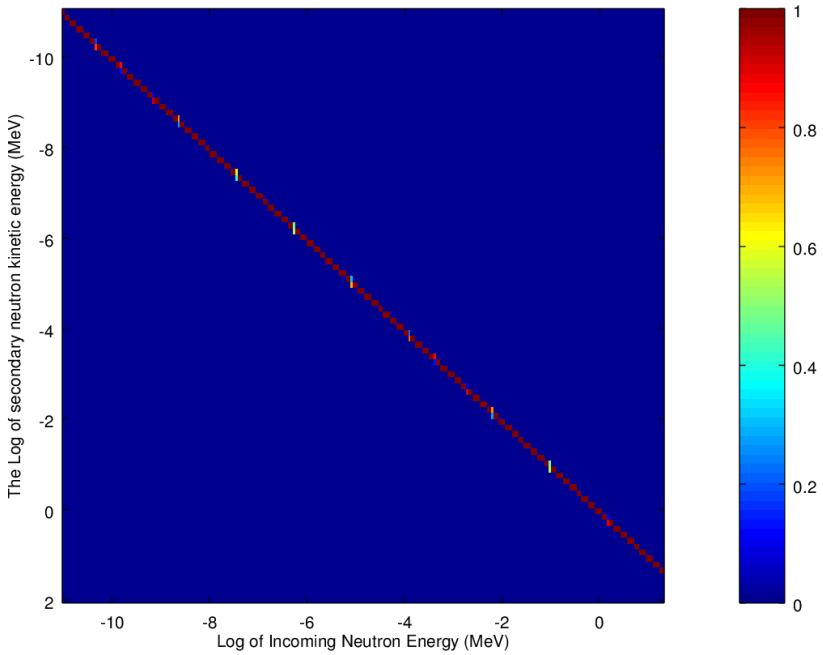


*Figure 6.19: The MCNP6 outgoing neutron energy for the elastic scattering of plutonium-242*

### 6.2.2 Fission Comparison

By examining Figure 6.20 to Figure 6.29, we can see that no significant differences exist between the physics methodology of G4STORK and MCNP, for the fission process. The small differences in the outgoing neutron energy spectrum will not significantly impact the results of SCWR simulation. While small differences in the yield distributions can be seen as well, these are primarily caused by statistical uncertainty, and not by differing methodology.

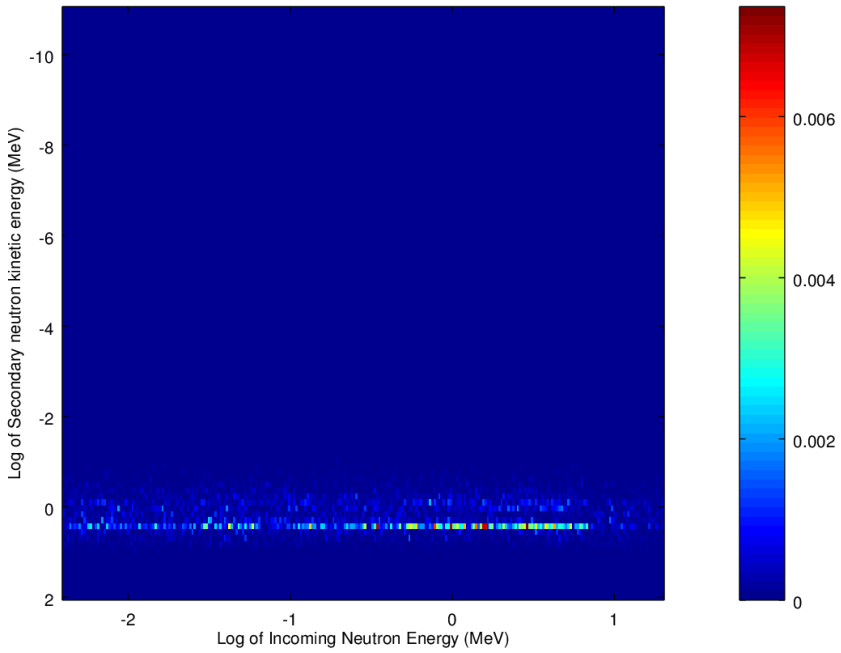


*Figure 6.20: The squared difference between the outgoing neutron energy for the fission of thorium-232*

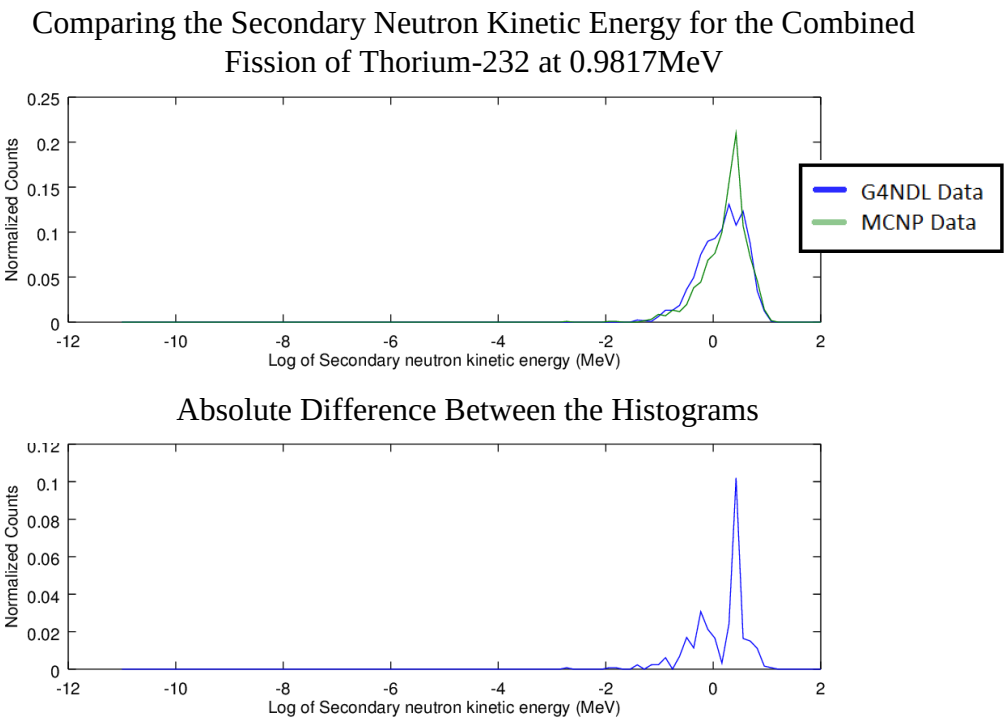


*Figure 6.21: The most discrepant outgoing neutron energy distributions for a fixed incoming neutron energy (column) of the fission of thorium-232*

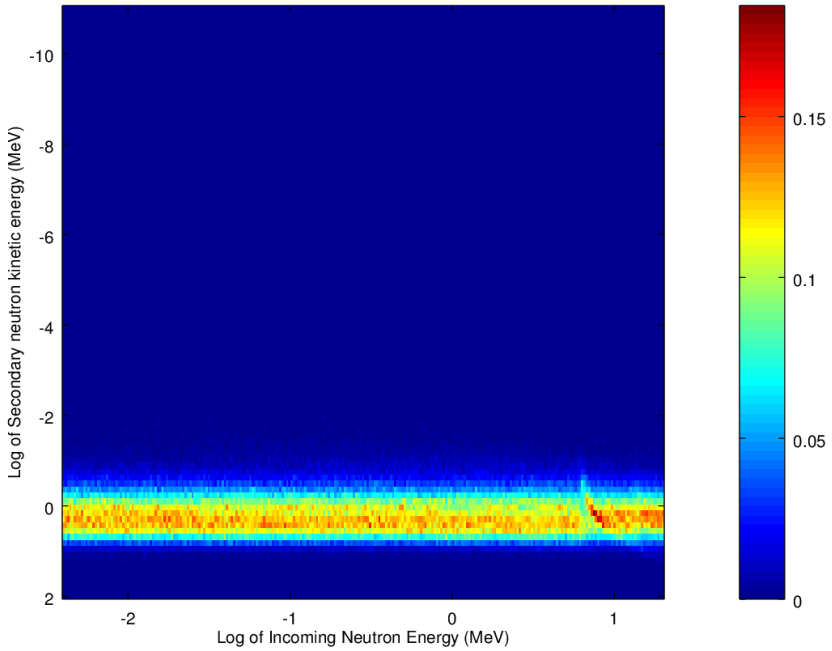


*Figure 6.22: The G4STORK outgoing neutron energy for the fission of thorium-232*

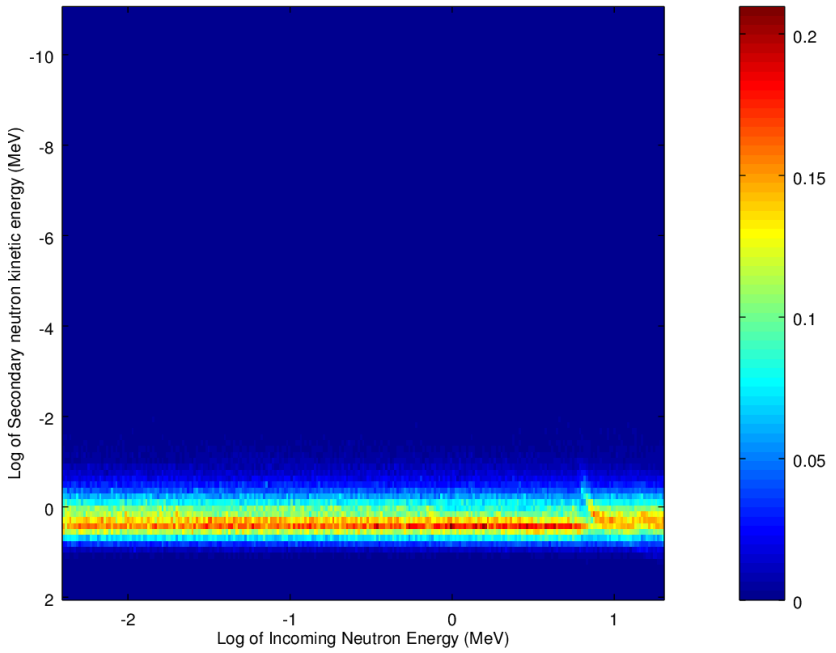


*Figure 6.23: The MCNP6 outgoing neutron energy for the fission of thorium-232*

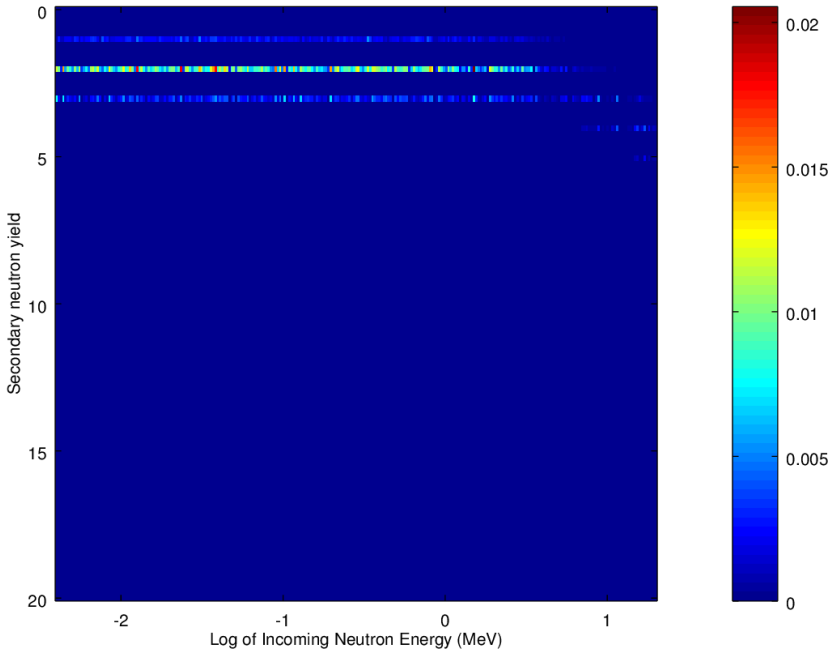


*Figure 6.24: The squared difference between the prompt neutron yield for the fission of thorium-232*

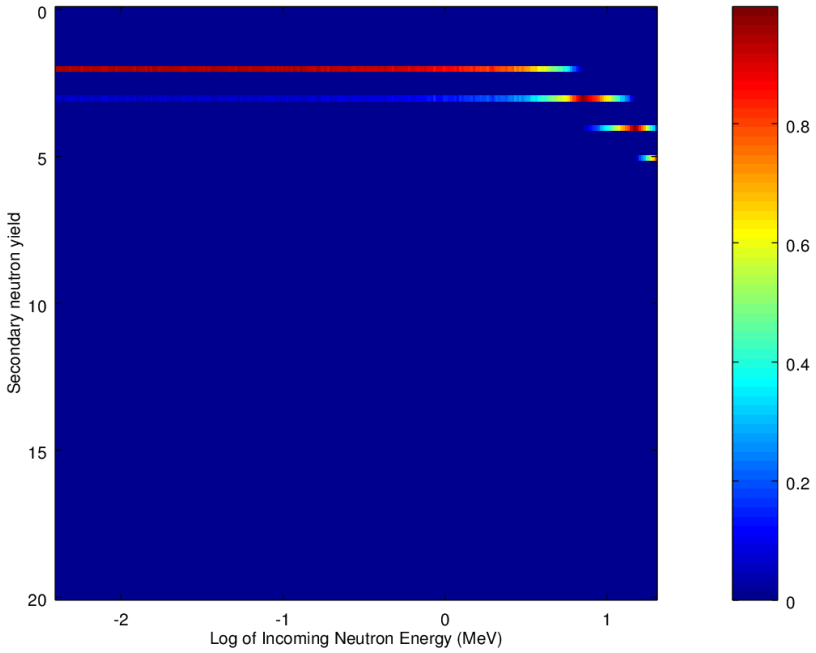


*Figure 6.25: The G4STORK prompt neutron yield for the fission of thorium-232*

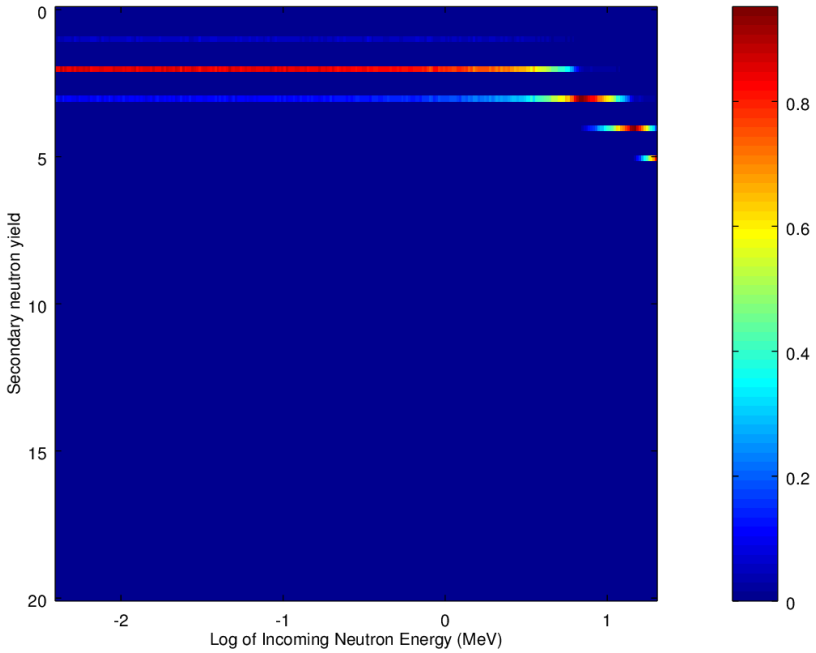


*Figure 6.26: The MCNP6 prompt neutron yield for the fission of thorium-232*

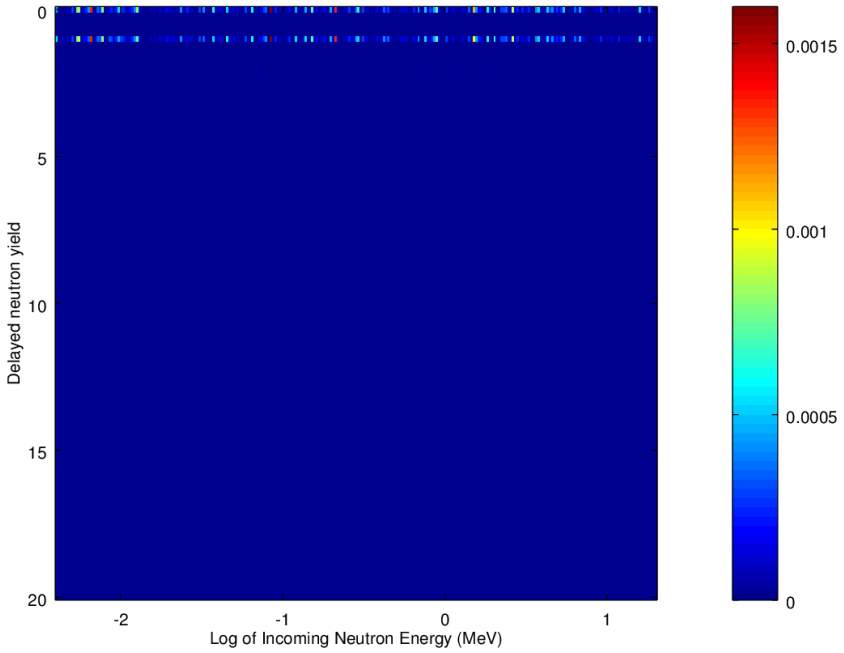


*Figure 6.27: The squared difference between the delayed neutron yield for the fission of thorium-232*

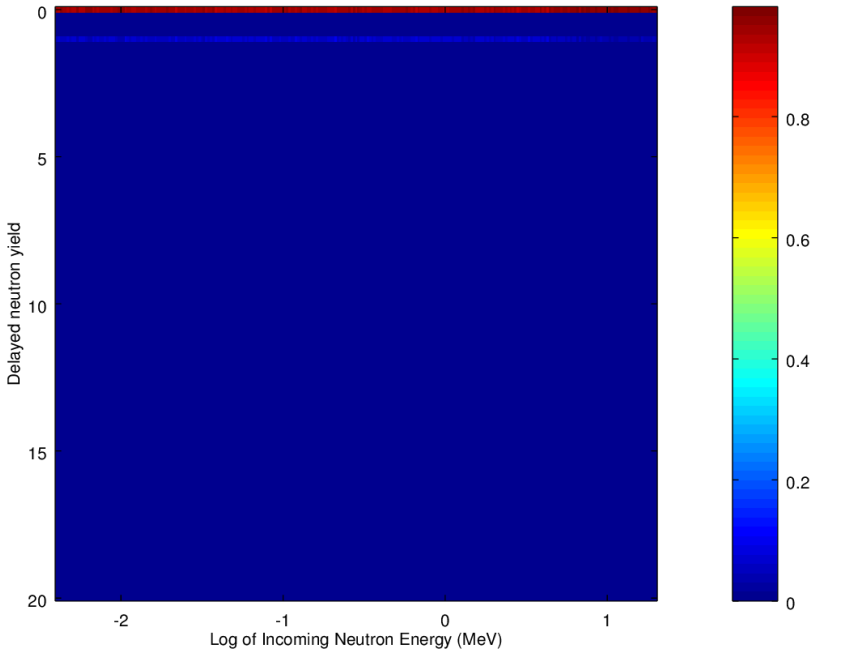


*Figure 6.28: The G4STORK delayed neutron yield for the fission of thorium-232*

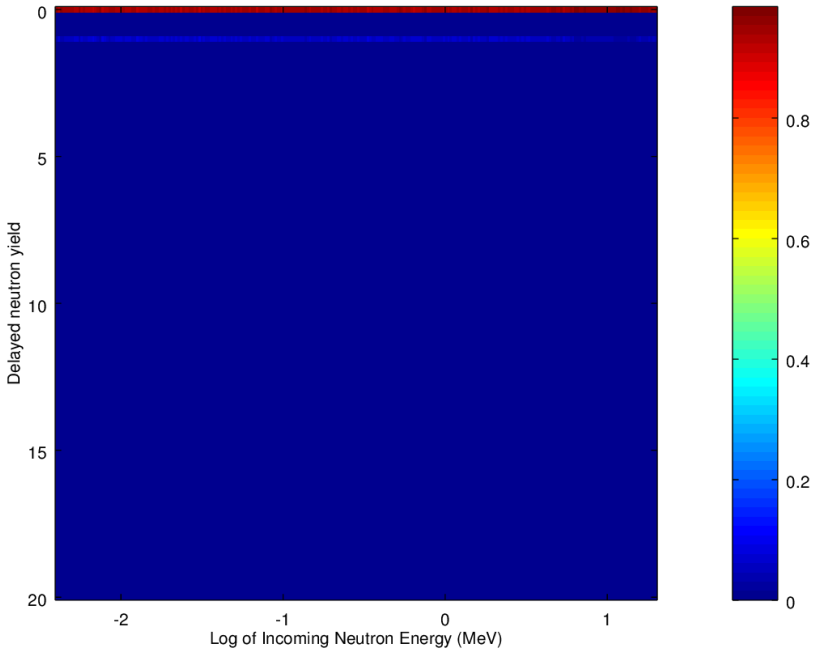


*Figure 6.29: The MCNP6 delayed neutron yield for the fission of thorium-232*

### 6.2.3 Inelastic MT=4 Comparison

By examining Figure 6.30 to Figure 6.32, we can see that no significant differences exist between the physics methodology of G4STORK and MCNP, for the inelastic MT=4 process.

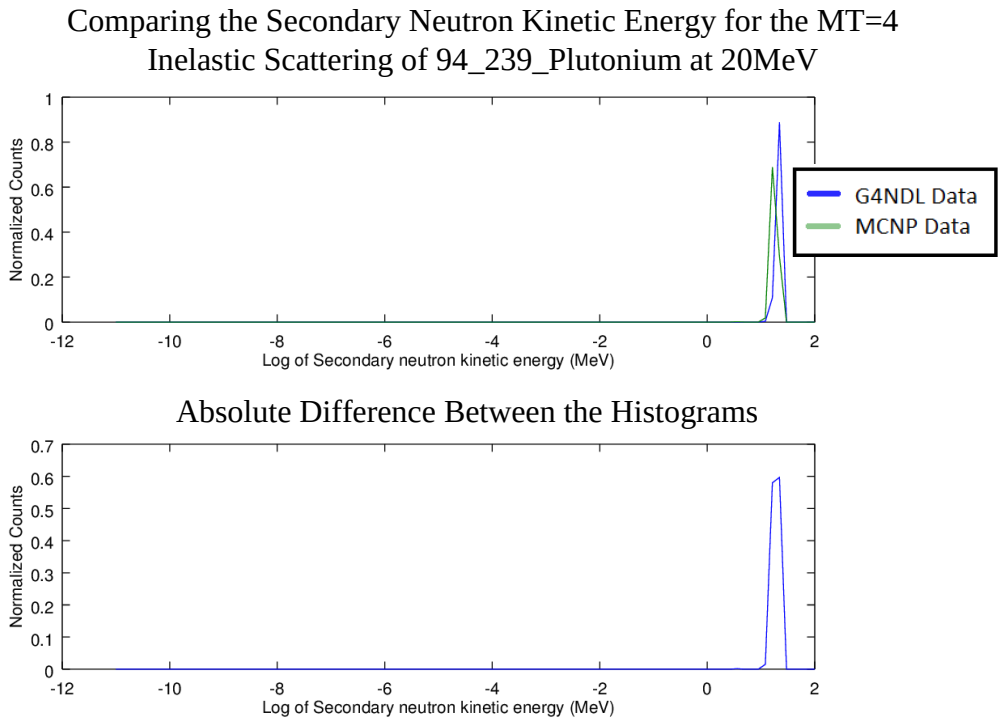


*Figure 6.30: The most discrepant outgoing neutron energy distributions for a fixed incoming neutron energy (column) of the inelastic MT=4 process of plutonium-239*

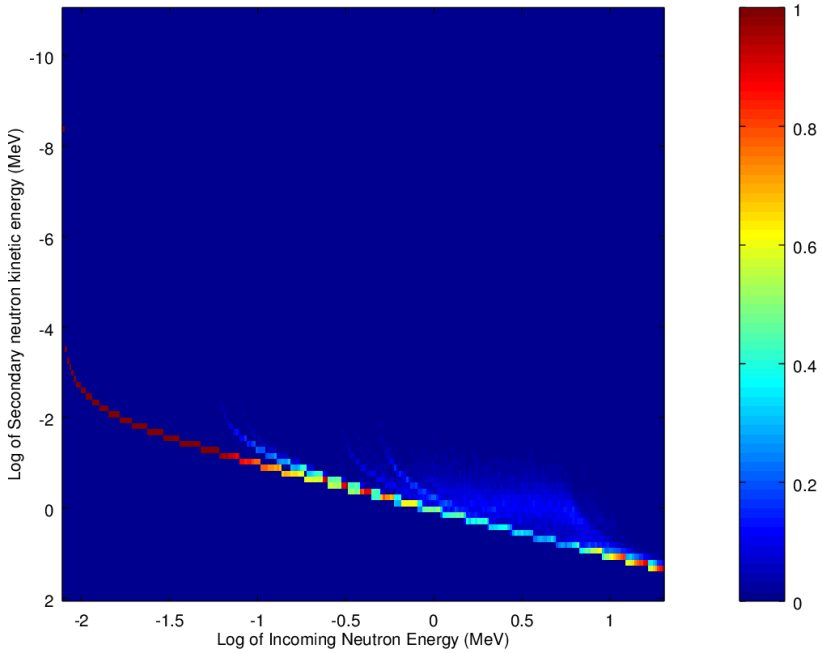


*Figure 6.31: The G4STORK outgoing neutron energy for inelastic MT=4 process of plutonium-239*

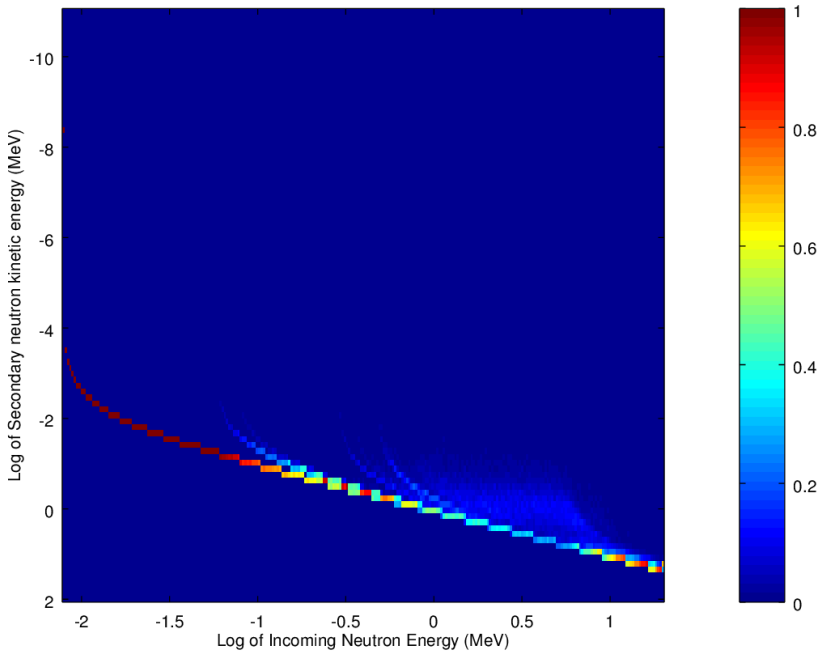


*Figure 6.32: The MCNP6 outgoing neutron energy for inelastic MT=4 process of plutonium-239*

### 6.2.4 Comparison of the Most Discrepant Data

The data that was found to differ the most for any process/isotope combination was the inelastic MT=22 data for europium-154. Images comparing the outgoing neutron energy distribution (We do not show the other properties since they matched perfectly) are shown below in Figure 6.33 to Figure 6.35. The data can be seen to differ significantly quantitatively when one compares the outgoing energy distributions at an incoming neutron energy where the two datasets were measured to differ the most (Figure 6.33). However, the data is still qualitatively quite similar, with peak values that are close enough to each other for this discrepancy to have an insignificant effect on the results of the SCWR simulation.

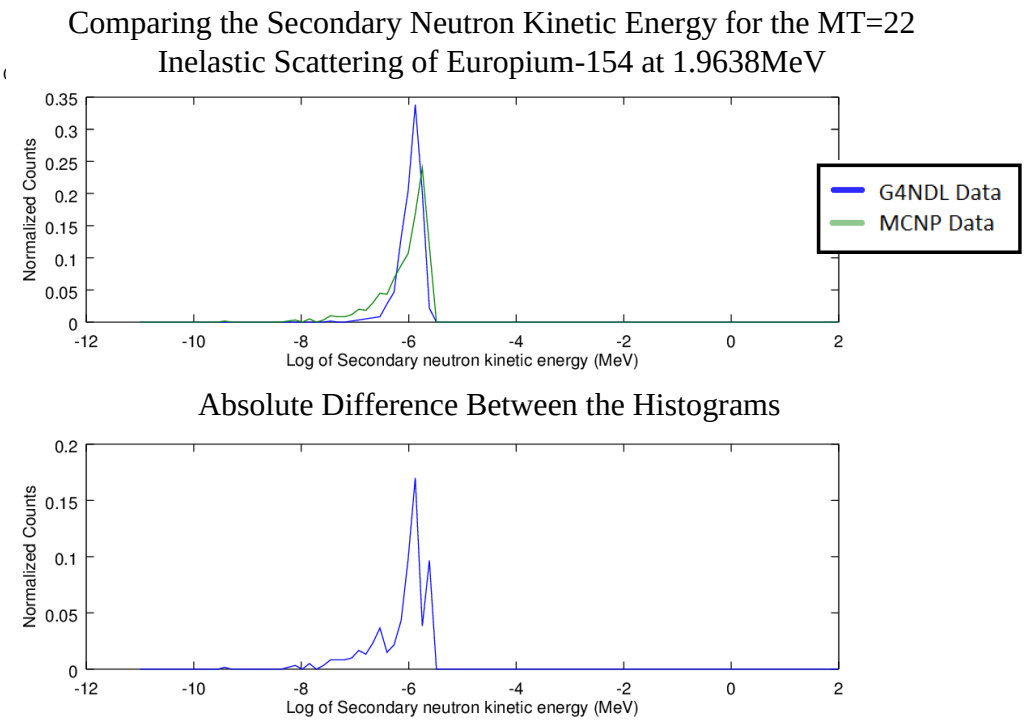


*Figure 6.33: The most discrepant outgoing neutron energy distributions for a fixed incoming neutron energy (column) of the inelastic MT=22 process of europium-154*

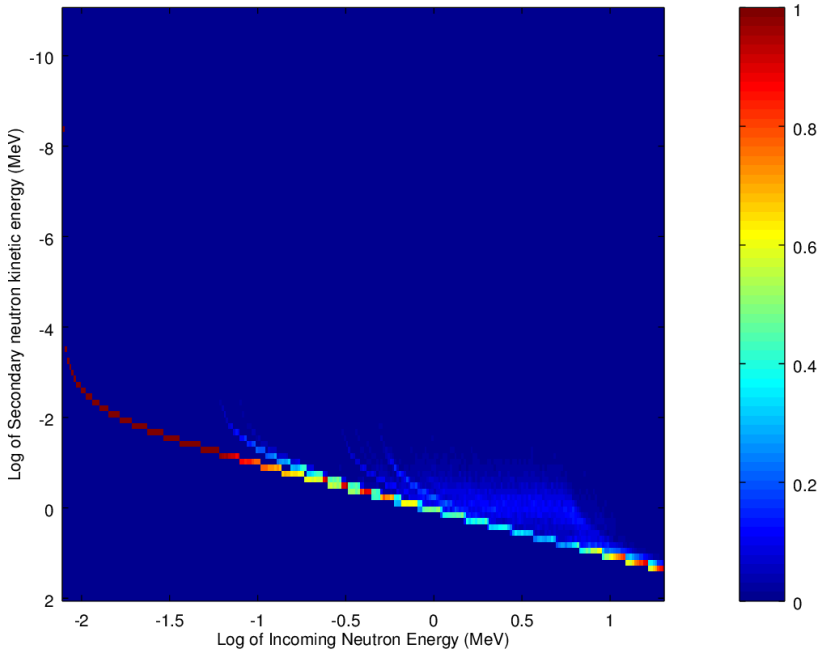


*Figure 6.34: The G4STORK outgoing neutron energy for inelastic MT=22 process of europium-154*

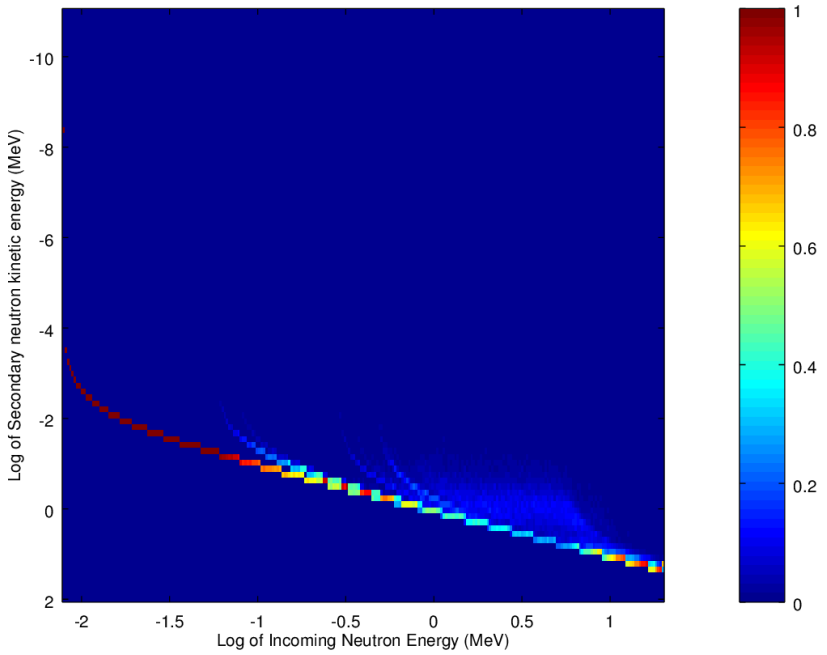


*Figure 6.35: The MCNP6 outgoing neutron energy for inelastic MT=22 process of europium-154*

## 6.3 The Results from Examining the Cause of the Remaining Discrepancy

In an attempt to determine where the remaining discrepancies between the G4STORK and MCNP were coming from, the source code of MCNP and G4STORK was further analyzed (described here, 3.4.4.2). During this process it was determined that the discrepancy was probably being caused by differing outgoing neutron energy distributions, differing cross-section data extraction algorithms, or differing transportation algorithms. The results collected from testing each of these possible causes, are analyzed in the following subsections.

### 6.3.1 The Results from Examining the Out-going Neutron Energy Distribution

By following the methodology described in section 3.4.4.3, the outgoing neutron energy distributions produced by MCNP and G4STORK were compared. The energy distributions that were measured to differ the most between codes are shown in Figure 6.36 to Figure 6.41 below. Note, if there is a significant (visible) difference between the MCNP and G4STORK histograms, a third image of the squared difference between the two will be shown. The difference between these energy distributions can clearly be seen to be insignificant. This means that differences in the outgoing neutron energy distributions are probably not the cause for the discrepancy between the results of G4STORK and MCNP.

#### 6.3.1.1 Elastic Scattering

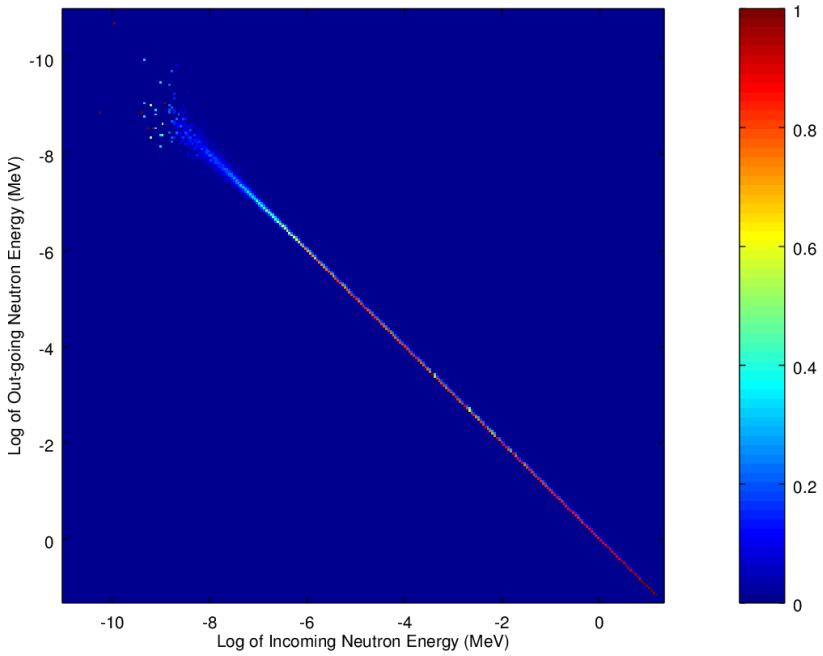


*Figure 6.36: The G4STORK outgoing neutron energy for the elastic scattering of zirconium-100*

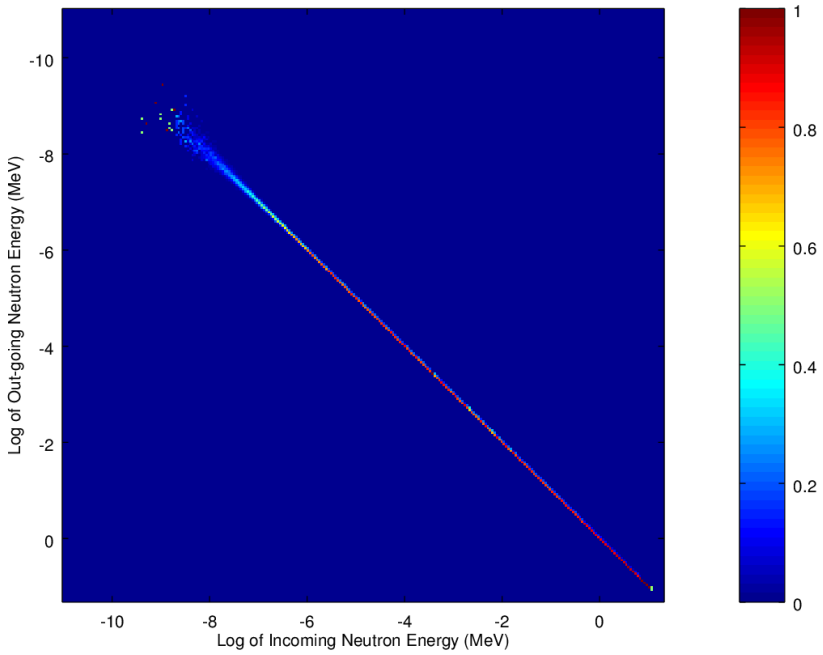


*Figure 6.37: The MCNP6 outgoing neutron energy for the elastic scattering of zirconium-100*

#### 6.3.1.2 Fission

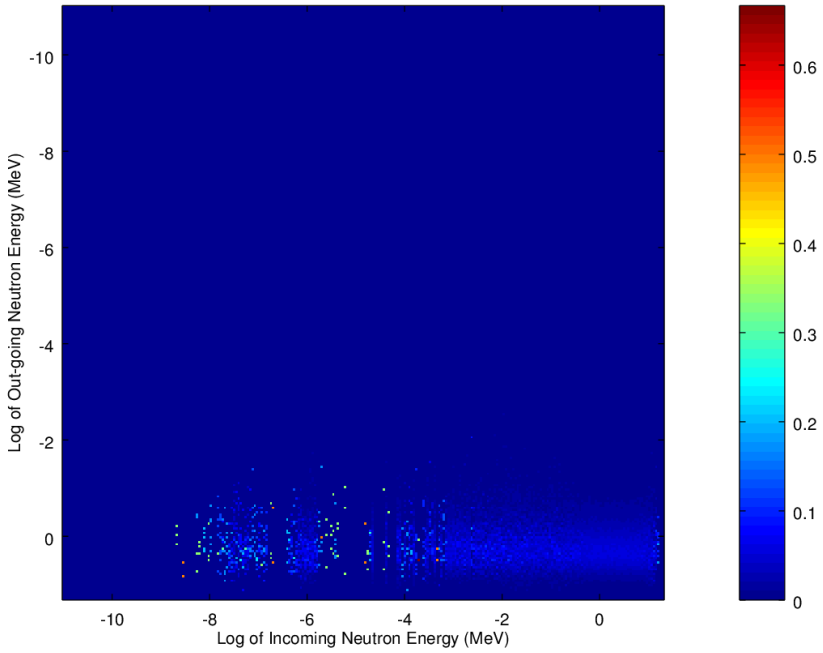


*Figure 6.38: The G4STORK outgoing neutron energy for the fission of plutonium-240*

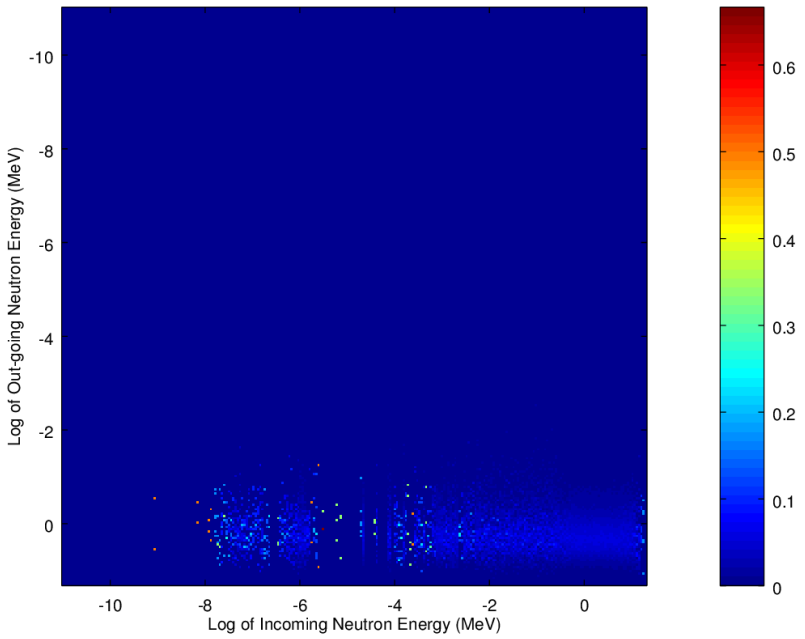


*Figure 6.39: The MCNP outgoing neutron energy for the fission of plutonium-240*

#### 6.3.1.3 Inelastic Scattering

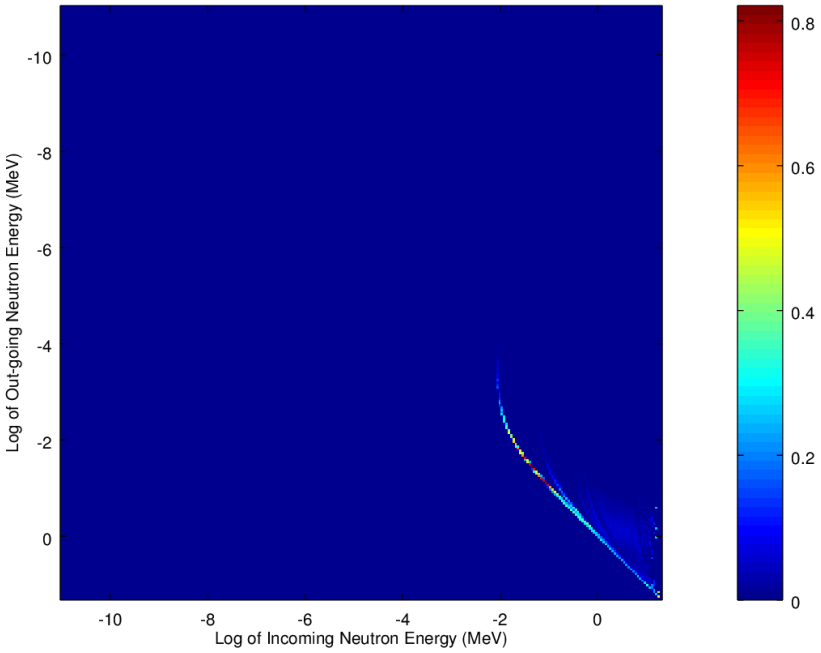


*Figure 6.40: The G4STORK outgoing neutron energy for the inelastic scattering of zirconium-100*

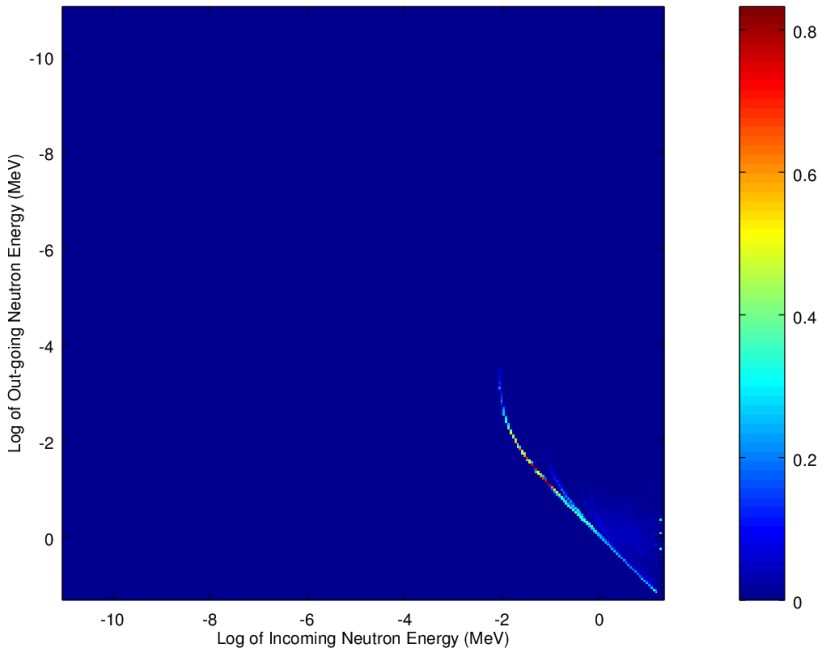


*Figure 6.41: The MCNP outgoing neutron energy for the inelastic scattering of zirconium-100*

### 6.3.2 The Results from Examining the Extracted Cross Section Data

By following the methodology described in section 3.4.4.4, the cross-section data modified by the extraction routine of G4STORK, and the cross-section data modified by the extraction routine of MCNP were compared to the original cross-section data. The cross-section data modified by G4STORK was found to line up perfectly with the original data. This wasn't a surprising result since GEANT4 only applies a 2% precision cutoff on the cross-section data. The cross-section data modified by MCNP was found to differ slightly from the original data. Some of the cross-section data modified by MCNP that were measured to differ the most, are shown in Figure 6.42 to Figure 6.45 below. Significant differences in the cross-section data can be seen to only occur in energy regimes not commonly reached by neutrons in a reactor. Thus the differing cross-section extraction routines are probably not the cause of the discrepancy between the results of G4STORK and MCNP.

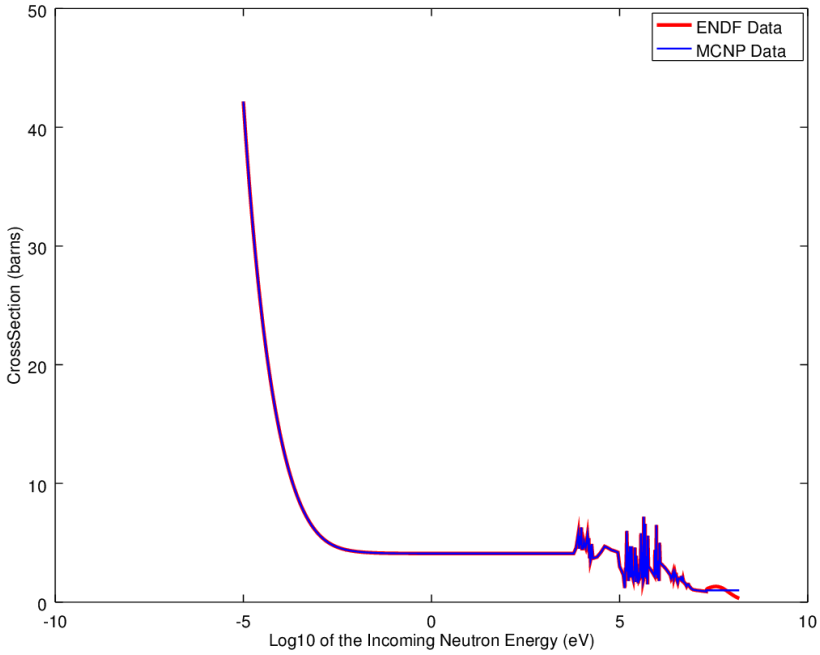


*Figure 6.42: compares the cross-section data of the original ENDF data and the extracted MCNP data for the elastic scattering process of phosphorus-31.*

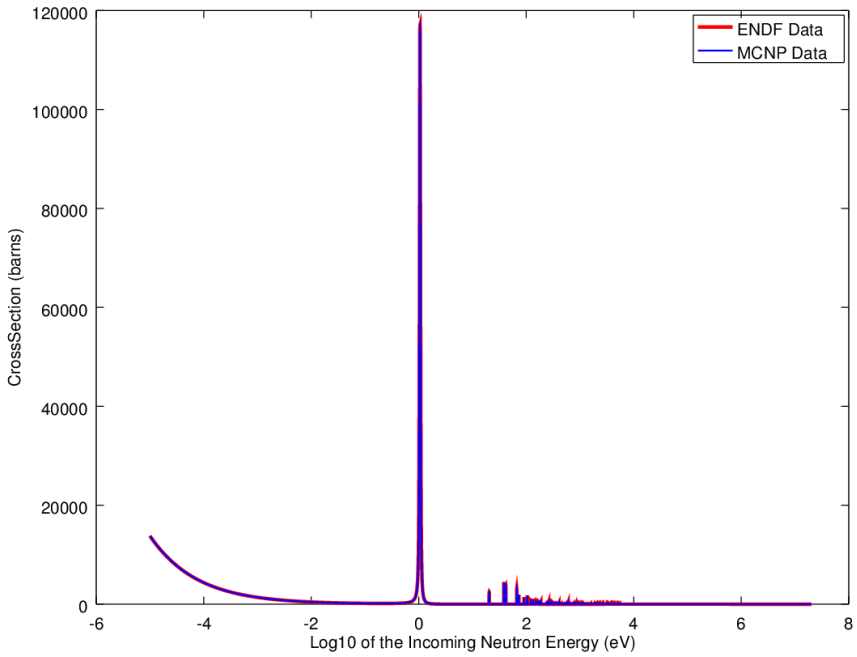


*Figure 6.43: compares the cross-section data of the original ENDF data and the extracted MCNP data for the capture process of plutonium-240.*

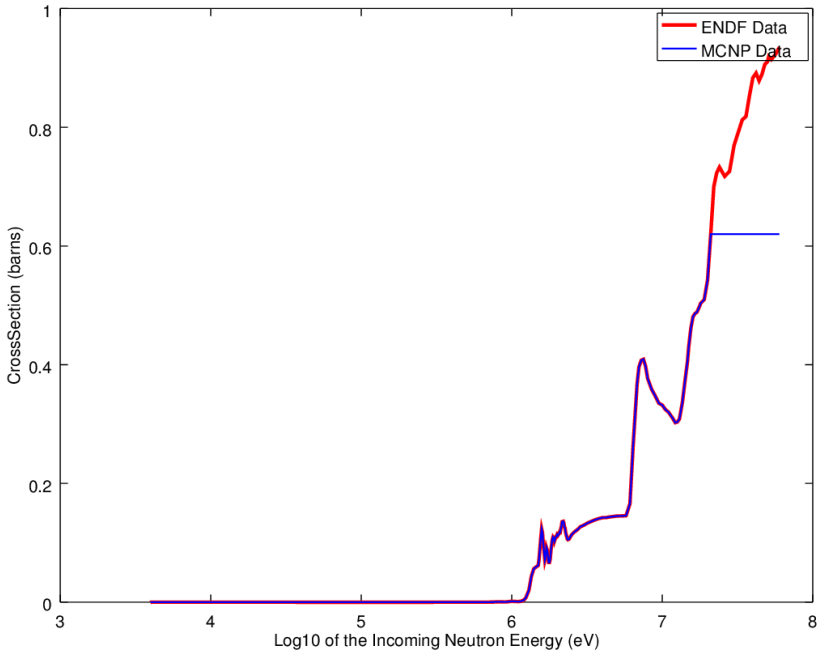


*Figure 6.44: compares the cross-section data of the original ENDF data and the extracted MCNP data for the combined fission process of thorium-232.*

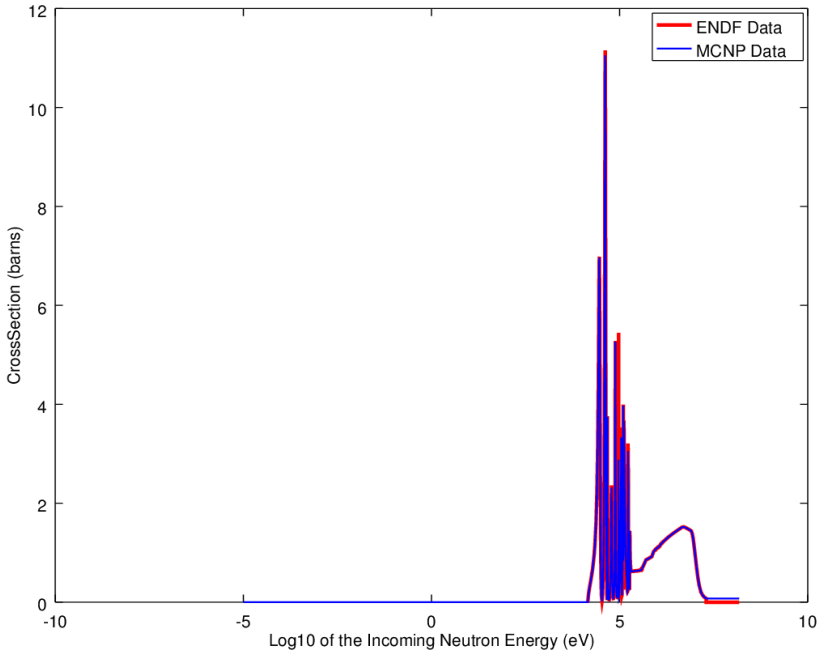


*Figure 6.45: compares the cross-section data of the original ENDF data and the extracted MCNP data for the inelastic scattering MT=4 process of iron-57*

# 7 References

[1] L. Russell, Simulation of Time-Dependent Neutron Populations for Reactor Physics Applications using the Geant4 Monte Carlo Toolkit, Hamilton, Ontario: McMaster University, 2012.

[2] D. Wang and S. Wang, "A Preliminary Cathena Thermal Hydraulic Model of the Canadian SCWR for Safety Analysis," Atomic Energy of Canada Limited, Chalk River, 2014.

[3] L. L. D. B. Y. O. K. Y. Y. B. a. G. W. T. Schulenberga, "Supercritical Water-Cooled Reactor (SCWR) Development through GIF Collaboration - IAEA-CN-164-5S06," [Online]. Available: http://www-pub.iaea.org/MTCD/publications/PDF/P1500_CD_Web/htm/pdf/topic5/5S06_H.%20Khartabil.pdf. [Accessed 01 07 2016].

[4] OECD, "Uranium 2014: Resources, Production and Demand," 2014. [Online]. Available: http://www.oecd-nea.org/ndd/pubs/2014/7209-uranium-2014.pdf.

[Accessed 01 07 2016].

[5] J. Sharpe, F. Salaun, D. Hummel, A. Moghrabi1, M. Nowak, J. Pencer, D. Novog and A. Buijs, "A Benchmark Comparison of the Canadian Supercritical Water-cooled Reactor (SCWR) 64-element Fuel Lattice Cell Parameters using Various Computer Codes," in *35th Annual Conference of the Canadian Nuclear Society*, Saint John, 2015.

[6] E. G. o. U. Analysis, "OECD Benchmark for Uncertainty Analysis in Best-Estimate Modelling (UAM) for Design, Operation and Safety Analysis of LWRs (OECD LWR UAM Benchmark)," Nuclear Energy Agency, State College, Pennsylvania, 2009.

[7] A. M. Jacobs, Basic principles of nuclear science and reactors, New Jersey: Princeton, 1960.

[8] Wikimedia, [Online]. Available: https://upload.wikimedia.org/wikipedia/commons/5/53/Binding_energy_curve_-_common_isotopes.svg. [Accessed April 2016].

[9] "Nuclear Power," Non Profit Nuclear Engineers, [Online]. Available: http://www.nuclear-power.net/. [Accessed April 2016].

[10] C. Shannon, "A mathematical theory of communication," *Bell System Technical Journal*, vol. 27, pp. 379–423, 623–656, 1948.

[11] D. Charles, "A CSEWG Retrospective," Brookhaven National Laboratory, Upton, New York, 2002.

[12] M. Mattes and J. Keinert, "Thermal Neutron Scattering Data for the Moderator Materials H2O, D2O and ZrHx in ENDF-6 Format and as ACE Library for MCNP(X) Codes," April 2005. [Online]. Available: http://www.iaea.org/inis/collection/nclcollectionstore/_public/37/088/37088511.pdf. [Accessed April 2016].

[13] Los Alamos National Laboratory, "MCNP6 USER'S MANUAL," May 2013. [Online]. [Accessed April 2016].

[14] J. M. Blatt and V. F. Weisskopf, Theoretical Nuclear Physics, Springer, 2012.

[15] M. Aziza, M. O. Shakerb, A. Aboanberb, E. Massouda and M. Slamaa, "SPECTRUM, IMPROVED FORMULA FOR PROMPT FISSION NEUTRON," *Journal of Nuclear and Radiation Physics,* vol. 6, pp. 31-41, 2009.

[16] W. Ford, Decryption of G4NDL, Hamilton, Ontario: McMaster University, 2015.

[17] GEANT4, "Physics Reference Manual," Dec 2015. [Online]. Available: http://geant4.web.cern.ch/geant4/UserDocumentation/UsersGuides/PhysicsReferenceManual/fo/PhysicsReferenceManual.pdf. [Accessed May 2016].

[18] University of Cambridge, "Scattering Theory," [Online]. Available: http://www.tcm.phy.cam.ac.uk/~bds10/aqp/lec20-21_compressed.pdf. [Accessed May 2016].

[19] K. Tatsumi, "Hadron Physics II - Geant4," Jan 2011. [Online]. Available: geant4.slac.stanford.edu/TAMU2011/HadPhysics2. [Accessed May 2016].

[20] Los Alamos National Laboratory, MCNP — A General Monte Carlo N-Particle Transport Code, Version 5, vol. III, Los Alamos, New Mexico: Los Alamos Controlled Publication, 2003.