Analysis of the Super-history powering method in Monte Carlo neutron transport simulations

JARL ROBERT PEDERSEN
Analysis of the Super-history powering method in Monte Carlo neutron transport simulations

JARL ROBERT PEDERSEN

Master in Nuclear Energy Engineering
Date: February 7, 2019
Supervisor: Jan Dufek
Examiner: Henryk Anglart
School of Physical Sciences
Abstract

The Monte Carlo method to solve the neutron transport problem by stochastic processes has attained increasing popularity over the years due to the declined cost of computing. Unfortunately, due to the nature of the stochastic evaluations, the desired results from a Monte Carlo simulation require relatively large computational resources compared to deterministic methods. Specifically for loosely coupled transport problems with inherently large dominance ratios, the super-history method is one of the simplest approaches to effectively reduce the bias to the converged neutron source distribution. This is achieved by reducing the frequency of neutron source normalizations to only be performed periodically, after a selected number of super-generations. Since the source neutron distribution is undisturbed when traversing through super-generations, the method has a theoretical advantage to reduce the neutron source distribution bias closer to the optimum asymptotic rate.

The purpose of this work is to demonstrate the effects of the super-histories to the bias of the neutron source distribution. An implementation of the super-history method for the Serpent Monte Carlo code was performed by modifying its source code. The test calculations of the modified code demonstrated a decreasing bias for increasing numbers of super-generations for a simple slab reactor of ten meters. On the other hand, the calculations showed that increasing numbers of super-generations lead to increased memory consumption and increases the risk for computational instabilities. The optimal utilization of the method is yet to be derived, however, this work has provided several diagnostics of the effects of the super-histories for a simple neutron transport problem.

Future developments of Monte Carlo codes and calculation methods may benefit from the super-history method to converge towards correct source neutron distributions. The super-history method would, in general, improve the convergence for arbitrary transport problems compared to regular power iterations for a moderate number of super-generations. However, for neutronic systems with dominance ratios approaching unity, no method exists for Monte Carlo simulations to guarantee a correct convergence.
# Contents

1 Introduction ................................................. 1  
   1.1 Purpose of the thesis ..................................... 1  
   1.2 Structure of the thesis .................................. 3  

2 Fundamental neutron transport .......................... 5  
   2.1 Neutron transport ....................................... 5  
   2.2 Fission neutron source .................................. 9  
   2.3 Scattering interactions .................................. 11  
   2.4 Thermal scattering ...................................... 19  
   2.5 Fission interactions .................................... 22  

3 The Monte Carlo criticality calculation method ........ 27  
   3.1 Pseudorandom number generation ....................... 29  
   3.2 Sampling methods ...................................... 31  
   3.3 Transport and collision kernels ......................... 34  
   3.4 Neutron splitting and Russian roulette ................. 37  
   3.5 Tallying ................................................ 38  

4 The Serpent Monte Carlo code ............................ 40  
   4.1 Pseudo-random number kernel .......................... 41  
   4.2 Tallying management .................................... 41  

5 The super-history method for criticality calculations 44  
   5.1 Method description ..................................... 44  
   5.2 Previous studies ....................................... 46  

6 Numerical test setup ....................................... 51  
   6.1 Acceleration criteria .................................... 51  
   6.2 Super-history implementation in Serpent 2 ............. 53  
   6.3 Source convergence problem ............................. 55
<table>
<thead>
<tr>
<th>Section</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>7 Results</td>
<td>58</td>
</tr>
<tr>
<td>7.1 Batch size of 100 neutrons</td>
<td>61</td>
</tr>
<tr>
<td>7.2 Batch size of 20 neutrons</td>
<td>67</td>
</tr>
<tr>
<td>8 Discussion</td>
<td>73</td>
</tr>
<tr>
<td>9 Conclusion</td>
<td>75</td>
</tr>
<tr>
<td>Bibliography</td>
<td>77</td>
</tr>
</tbody>
</table>
Chapter 1

Introduction

1.1 Purpose of the thesis

For many decades, nuclear fission energy has provided large shares of the electrical power demand to several countries. The interest in determining the neutron transport and the spatial fission source is relevant when designing reactor cores. As well for the design of radiation shielding, fuel assemblies, control rods and selection of cooling materials.

The development of computer codes has made simulations of large and complex geometric configurations possible. The aim of the development of computer codes has been to provide general purpose applications to simulate the neutron density and multiplication factors for any materials and geometries. A recent trend for some codes (MCNP [50], Serpent [30], RMC[48], etc.) has been to include coupled physics simulations like neutron-photon coupling, fuel burn-up schemes, structural analysis as well as fluid dynamics and heat transfer simulations to name a few.

New functionalities can provide engineers and scientists with new tools for simulating reactors and evaluate the multiple behaviors of the reactor in a performance targeted study or for safety assessments. These tools have the goal to support such studies with accurate simulations, which can reduce the economic and societal consequences of avoiding failures and accidents or unexpected decreasing performance at nuclear reactor facilities. As these tools provide extensive calculations for reactor safety, the computer codes have a direct influence of the public perception on nuclear reactor safety and acceptance of nu-
clear power. Therefore, the development of the codes has to provide high-quality implementations of theoretical models to resemble experimental results.

Naturally, more functionalities will unavoidably increase the computational resources required for such simulations. Therefore, developing computer codes to improve the utilization of computational resources is justified. In particular, for Monte Carlo criticality calculations, the computational cost can become unreasonably large for a highly non-homogeneous material distribution of control rods, fuel rods, neutron moderators and poisons in a large spatial configuration. Therefore, the development of acceleration methods is of interest, which can provide accurate results within shorter computing times.

There are widely regarded three major approaches for solving the neutron transport in a given configuration, including solutions of the Fick’s diffusion approximation, numerical solutions to the Boltzmann’s integro-differential equation with a collision and fission reaction terms, and lastly Monte Carlo calculations. Most of these solutions evaluate the steady-state transport of the neutrons together with a multiplication eigenvalue.

For Monte Carlo calculations, several acceleration methods [6, 9, 14, 16, 21, 24, 25, 27, 52] for criticality simulations have been proposed. In this work, the method of interest is the super-history powering method, initially proposed by Brissenden and Garlick [6]. The purpose of this thesis is to analyze the effect of super-histories on the fission neutron source convergence.

The super-history powering method is a modification of the power iteration method. In power iteration, the neutron fission source distribution is sampled after a single generation of neutrons is simulated. The next generation of neutrons is generated by an appropriate batch size of the sampled distribution, and the process is repeated for an appropriate number of cycles. Unfortunately, to converge the source to the fundamental eigenmode, the power iteration is prone to decaying convergence rate due to limited statistics from the simulated neutrons. Brissenden and Garlick [6] argued that this leads to over-representation of statistically insignificant neutrons due to the sampled source neutrons from the previous cycle. This is especially true for loosely coupled systems and heterogeneous spatial distributions of fissile materials, arising to dominance ratios close to unity. This may also be thought as the sampling causes a distortion of ignoring
the ability of canceled neutrons to generate fission neutrons in their subsequent generations. This issue is usually resolved by increasing the batch size to provide better statistics, however, this increases the number of numerical operations and memory demand.

To overcome this deficiency, the super-history powering method is introduced where the batch size sampling of the neutron fission source is only conducted after a specified number of neutron generations has been simulated. For each generation, the simulated neutrons are produced by the previous generation fission source neutrons. Thus no sampling is conducted between the cycle generations and unlike the regular power iteration, the distortion effect from the canceled neutrons are reduced when sampling for the initial generation for the next cycle.

The disadvantage of the super-history powering method is that the successive generations may lead to an extinction of the neutrons, especially for a severely biased fission source or an overestimated eigenvalue. This leads to either a complete termination of the simulation or the necessity to resample a fission distribution from the previous cycle. However, this disadvantage is considered manageable for up to 10 super-generations per cycle.

The study conducted by She, Wang, and Yu [38] indicated that the super-history powering has not improved the computing efficiency towards fission source convergence. On the other hand, other studies have found that the method provided significant improvements and the method has even been implemented into the MONK7 code [40, 41], as well in the MCNP5 code [43] as a patch [7, 8]. Both implementations were found to be advantageous regarding the convergence of the source distribution, and the fundamental eigenvalue.

1.2 Structure of the thesis

This thesis is build up by two components; the first component aims to describe the fundamental techniques in the Monte Carlo calculations in Chapter 2 and 3. The second component describes the super-history method in detail and presents the implementations and results in Chapter 5-7.

Chapter 2 introduces the reader to the models used in neutron transport, and justification of necessary simplifications. The chapter
takes the reader from the wave mechanics in the Schrödinger’s equation to the Boltzmann transport theory where probability densities replace the wave mechanics. The interactions in the deduced transport problem are described in detail, providing a concise explanation of the mechanics and the reliance on experimental data.

Chapter 3 describes the essentials of the criticality Monte Carlo method and applies the probability functions that are discussed in Chapter 2. Some additional features are introduced in Chapter 4, describing the specific implementations for the Serpent code. The framework for Monte Carlo simulations in Chapter 2-4 provides an overview of the concepts relevant for Monte Carlo simulations.

Chapter 5 presents the mechanisms of the super-history method in detail, as well as the advantages and shortcomings. A literature study of previous assessments of the method is considered and evaluated, mainly focused on the study of source convergence for the RMC code [38].

Chapters 6 and 7 describe the modifications in the Serpent code and the results that the super-history method provided. Finally, the discussion and conclusion in Chapter 8 and 9, evaluates the implications of the method, results, as well to suggest improvements to the modifications and development of new methods.
Chapter 2

Fundamental neutron transport

2.1 Neutron transport

Deterministically, no particle transport can be described sufficiently for an arbitrary system to the actual transport occurring in nature. The reason for this is due to the uncertainty principle introduced by Heisenberg in 1927, describing that a particle exhibits uncertainties in predicted positions and momentum as:

$$\sigma_{\vec{r}} \sigma_{\vec{p}} \geq \frac{\hbar}{2}$$ (2.1)

where $\sigma_{\vec{r}}$ is the deviation of the particle position $\vec{r}$, and $\sigma_{\vec{p}}$ is the deviation of the particle momentum $\vec{p} = m\vec{v}$, where $m$ is the particle mass, and $\vec{v}$ is the velocity vector. $\hbar$ is the reduced Planck constant $^1$.

The randomness of particle properties can be applied to a probability density of particles. This would require that the wave function in the Schrödinger equation is solved (assuming non-relativistic particles), and the probability density function for the particle species $j$ is computed by:

$$p_j(\vec{r}_j, t) = \int_{\mathcal{R}\neq \vec{r}_j} d\mathcal{R} \psi(\mathcal{R}, t) \psi^*(\mathcal{R}, t)$$ (2.2)

where $\psi(\mathcal{R}, t)$ is the time-dependent wave function, and $\psi^*(\mathcal{R}, t)$ is its complex conjugate. $\mathcal{R} = \{\vec{r}_k; k = 1..n\}$ is a set of position vectors for a system of $n$ particles. The Schrödinger equation takes the time-dependent form:

$^1 \hbar = 6.582119514(\pm4 \times 10^{-8}) \times 10^{-16}$ eV s [35]
$i \frac{\partial}{\partial t} \psi(R, t) = \hat{H} \psi(R, t)$ \quad (2.3)

where $i = \sqrt{-1}$, $\partial/\partial t$ is the time derivative, and $\hat{H}$ is the Hamilton operator given as:

$$\hat{H} = -\sum_{k=1}^{n} \frac{\hbar^2}{2m_k} \nabla_k^2 + \sum_{j=1}^{n} \sum_{k=1}^{n} V(\vec{r}_j, \vec{r}_k) \quad (2.4)$$

where $m_k$ is the mass of the $k$th particle, $\nabla_k^2$ is the spatial Laplace operator, and $V(\vec{r}_j, \vec{r}_k)$ is the potential applied by particle $k$ to particle $j$. The potential formally consists of all the potentials between the particles, such as electric, gravitational and nuclear potentials. For neutrons, the electric and gravitational potentials may be neglected.

Applying the probability density approach is too cumbersome to be applied for large systems of particles. For instance, a macroscopic system consisting of an Avogadro’s constant ($N_A$)$^2$ number of particles would result in a $N_A$ number of Laplace operators and up to $N_A \cdot (N_A - 1) \simeq N_A^2$ number of particle potentials. A complete description of the strong nuclear force would require the interactions between the quarks, and a few models have been developed to approximate a solution with these effects such as Skyrme Hartree-Fock calculations [45]. Clearly, there would be only practically achievable solutions for a few particles by this approach and would not be suitable for a bulk neutron transport problem.

To perform particle transport for a macroscopic system, the bulk transport of the particles are of interest. The exact number of particles cannot be predicted, but a density of particles can be predicted. A sufficient transport theory will preserve the randomness of particles, thus there is a necessity to apply probability density functions for the possible interactions that take place.

The Boltzmann’s transport theory describes macroscopic thermodynamical statistical systems in a non-equilibrium state. The theory combines macroscopic bulk transport behavior of large bodies of thermodynamic systems together with statistical thermodynamic theory and probability density functions. Thus bridging the gap between macroscopic and microscopic thermodynamical systems. Applications

\[ N_A = 6.022140856(\pm 7.4 \times 10^{-8}) \times 10^{23} \text{ [35]} \]
for the Boltzmann’s transport theory is found in many problems dealing with non-equilibrium macroscopic transport where microscopic properties affect the bulk transport. Some examples are plasma transport, heat diffusive processes, molecular and particle transport in macroscopic scales. The macroscopic scales may range from laboratory scale chambers to astronomical scales where plasma transport are of interest.

The transport equation may be taken as [12]:

\[
\frac{\partial n(\vec{r}, \vec{v}, t)}{\partial t} + \vec{v} \cdot \nabla n(\vec{r}, \vec{v}, t) + \frac{\vec{F}}{m_p} \cdot \frac{\partial n(\vec{r}, \vec{v}, t)}{\partial \vec{v}} = \left( \frac{\partial n(\vec{r}, \vec{v}, t)}{\partial t} \right)_{\text{coll}} + q \tag{2.5}
\]

where \(n(\vec{r}, \vec{v}, t)\) is the angular particle density, \(\partial / \partial t\) is the time derivative, \(v = |\vec{v}|\) is the particle velocity, \(\nabla\) denotes the spatial gradient and \(m_p\) is the particle mass. The applied body forces \(\vec{F}\) may describe long-range forces applied to the particles, such as electrostatic or gravitational fields. Short range forces, such as nuclear forces are handled by the collision term, \(\left( \partial n(\vec{r}, \vec{v}) / \partial t \right)_{\text{coll}}\). The collision term is thus preserving the randomness of the particles as they collide, thus the randomness of the particles are only considered at the collision sites. \(q\) is an external particle source that may be present in the transport problem, either artificially or from physical processes.

The Boltzmann collision term including scattering, fission, and absorption is relevant for neutron transport in multiplicative media and is taken as:

\[
\left( \frac{\partial n(\vec{r}, \vec{v}, t)}{\partial t} \right)_{\text{coll}} = \int_0^\infty dE' \int_{4\pi} d\hat{\Omega}' \Sigma_s(\vec{r}, \hat{\Omega}', E' \rightarrow \hat{\Omega}, E)n(\vec{r}, \hat{\Omega}', E', t)v(E')
\]

\[
+ \frac{\chi(\vec{r}, E)}{4\pi} \int_0^\infty dE' \int_{4\pi} d\hat{\Omega}' \nu(\vec{r}, \hat{\Omega}', E') \Sigma_f(\vec{r}, \hat{\Omega}', E')n(\vec{r}, \hat{\Omega}', E', t)v(E')
\]

\[
- \Sigma_t(\vec{r}, E)n(\vec{r}, \hat{\Omega}, E, t)v(E) \tag{2.6}
\]

where \(\Sigma_i\) is the macroscopic reaction cross section that may simply be represented by the nuclide densities \((N_j)\) contributing to the probability of the \(i\)th reaction as \(\sum_j N_j \cdot (\sigma_i)_j\). \(\Sigma_s\) represents the cross section for scattering, \(\Sigma_f\) for fission, and \(\Sigma_t = \sum_i \Sigma_i\) is the total reaction cross section. The absorption cross section describes all non-scattering processes as \(\Sigma_a = \Sigma_t - \Sigma_s\). Here, the collision term is also dependent on
the angular dependence of the neutron density \( n(\vec{r}, \hat{\Omega}, E, t) \). The angle integral \( \int_{4\pi} d\hat{\Omega}' \) is equivalent to \( \int_0^\pi \sin \theta' d\theta' \int_0^{2\pi} d\varphi' \), using the azimuthal and polar angles of the incident neutron. \( \chi(\vec{r}, E) \) and \( \bar{\nu}(\vec{r}, E') \) are the fission energy spectrum and the average number of neutrons emitted per fission respectively, depending on the fissionable nuclide residing at spatial location \( \vec{r} \).

The particle flux \( \psi \) is described as the product of the particle density and the particle velocity as:

\[
\psi(\vec{r}, \vec{v}, t) = n(\vec{r}, \vec{v}, t) \cdot v
\]

(2.7)

For convenience, \( n(\vec{r}, \vec{v}, t) \) in Eq. (2.5) is replaced with the right-hand side of Eq. (2.7). The advantage is that this allows the transport equation to change its domain from density change per time unit to change of flux of spatial direction and simplifies the mathematical problem. Thus, for critical neutron transport, the Boltzmann’s collision term including fission, scattering and leakage terms, without body forces imposed to the neutrons is:

\[
\frac{1}{v(E)} \frac{\partial \psi(\vec{r}, \hat{\Omega}, E, t)}{\partial t} + \hat{\Omega} \cdot \vec{\nabla} \psi(\vec{r}, \hat{\Omega}, E, t) + \Sigma_t(\vec{r}, E) \psi(\vec{r}, \hat{\Omega}, E, t)
\]

\[
= \int_0^\infty dE' \int_{4\pi} d\hat{\Omega}' \Sigma_s(\vec{r}, \hat{\Omega}', E' \rightarrow \hat{\Omega}, E) \psi(\vec{r}, \hat{\Omega}', E', t)
\]

\[
+ \frac{\chi(\vec{r}, E)}{4\pi} \int_0^\infty dE' \int_{4\pi} d\hat{\Omega}' \bar{\nu}(\vec{r}, E') \Sigma_f(\vec{r}, \hat{\Omega}', E') \psi(\vec{r}, \hat{\Omega}', E', t) + Q
\]

(2.8)

The source \( Q = q/v \) may be interpreted as an arbitrary source \( Q(\vec{r}, \hat{\Omega}, E) \) which may represent an external source or be utilized for adding delayed neutrons from decaying fission products.

Note that Eq. (2.8) assumes a completely decoupled energy distribution of the angular distribution, and assumes that \( \chi(\vec{r}, E) \) is independent of \( \hat{\Omega}' \). The fission term also assumes an immediate release of prompt neutrons, although the actual release time is about \( 10^{-13} \) seconds after an initiated fission event [11]. Despite this difference, the assumption is reasonable when we consider the neutron generation time in the order of about \( 10^{-5} \) seconds.

In criticality calculations, a pseudo-steady state of the reactor is assumed, therefore we also assume time independence of Eq. (2.8) by applying:

\[
\frac{\partial \psi(\vec{r}, \hat{\Omega}, E, t)}{\partial t} = 0
\]

(2.9)
and considering the time-independent angular flux as \( \psi = \psi(\vec{r}, \hat{\Omega}, E) \). To justify this steady state, a factor \( k^3 \) will be introduced in order to correct the fission multiplying term by \( \bar{\nu}(\vec{r}, E') \rightarrow \bar{\nu}(\vec{r}, E')/k \).

Furthermore, the source \( Q \) is often neglected, assuming no influence of external sources or delayed neutrons (alternatively, \( \bar{\nu} \) may be modified to \( \bar{\nu}/(1 - \beta) \) in order to accommodate those neutrons for accurate \( k \)-estimation in conjunction with an energy spectrum \( \chi(E) \) corrected with delayed neutron energies). With these considerations applied, Eq. (2.8) without delayed neutrons becomes:

\[
\hat{\Omega} \cdot \nabla \psi(\vec{r}, \hat{\Omega}, E) + \Sigma_t(\vec{r}, E)\psi(\vec{r}, \hat{\Omega}, E) = \int_0^\infty dE' \int_{4\pi} d\hat{\Omega}' \Sigma_s(\vec{r}; \hat{\Omega}', E' \rightarrow \hat{\Omega}, E')\psi(\vec{r}, \hat{\Omega}', E') + \frac{\chi(E)}{4\pi k} \int_0^\infty dE' \int_{4\pi} d\hat{\Omega}' \bar{\nu}(E')\Sigma_f(\vec{r}, E')\psi(\vec{r}, \hat{\Omega}', E')
\]

(2.10)

### 2.2 Fission neutron source

For Monte Carlo calculations, the computations are targeted towards the fission neutron density instead of a neutron density dependent on kinetic neutron energy, and angles. Deducing the critical neutron transport problem make use of transport operators in order to abstract the transport mechanisms. We may rewrite Eq. (2.10) as:

\[
B\psi = \frac{1}{k} M\psi
\]

(2.11)

where \( B \) is defined as the transport operator consisting of the leakage operator \( L \), the collision operator \( C \) and the scattering operator \( S \) as:

\[
B = L + C - S = \hat{\Omega} \cdot \nabla + \Sigma_t(\vec{r}, E) - \int_0^\infty dE' \int_{4\pi} d\hat{\Omega}' \Sigma_s(\vec{r}; \hat{\Omega}', E' \rightarrow \hat{\Omega}, E')
\]

(2.12)

and \( M \) is the fission operator:

\[
M = \frac{\chi(E)}{4\pi} \int_0^\infty dE' \int_{4\pi} d\hat{\Omega}' \bar{\nu}(E')\Sigma_f(\vec{r}, E')
\]

(2.13)

\( k \) is in the succeeding sections deduced to an eigenvalue corresponding for the \( i \)-th modes of the angular flux. For this section a constant is sufficiently describing the transport problem.
Multiplying Eq. (2.11) by $MB^{-1}$ from the left gives:

$$M\psi = \frac{1}{k}MB^{-1}M\psi$$

(2.14)

Introducing the fission source density rate $s = M\psi$ and the operator $F = MB^{-1}$, reduces Eq. (2.14) to:

$$ks(\vec{r}) = Fs(\vec{r})$$

(2.15)

The fission operator $F$ has the physical interpretation as the ratio of fission neutrons generation rate per neutron absorption and leakage rates. The operator now describes the eigenvalue problem in Eq. (2.15), subjected to the eigenvalue $k$ and eigenvector $s(\vec{r})$. Rearranging Eq. (2.15), we isolate the expression for the eigenvalue $k$ as:

$$k = \frac{Fs(\vec{r})}{s(\vec{r})}$$

(2.16)

$k$ is now describing the relative change of fission source density rate on the response of $F$.

Considering that Eq. (2.16) is collectively evaluating the eigenvalue for all the eigenvectors $s(\vec{r})$, we can consider that there exists $N$ sets of eigenvalues $k_N$ and eigenvectors $s_N(\vec{r})$, and require that there exist a globally positive fundamental solution $s_0(\vec{r})$. Then we have a physical solution of:

$$k_0 = \frac{Fs_0(\vec{r})}{s_0(\vec{r})}$$

(2.17)

where the fundamental eigenvalue $k_0$, is often taken as the effective eigenvalue, $k_{eff}$. As there may exist higher order solutions of $s_i(\vec{r})$, it’s not guaranteed that they are globally positive. The eigenvalues can therefore be classified as: $k_0 > |k_1| > |k_2| > ... > |k_i|$. Furthermore, $s(\vec{r})$ may be presented as a linear series of the eigenvectors as:

$$s(\vec{r}) = \sum_i \gamma_i Fs_i(\vec{r}) = \sum_i \gamma_i k_i s_i(\vec{r})$$

(2.18)

Finally, Eq. (2.18) may be divided by the fundamental eigenvalue $k_0$ to give:

$$\frac{s(\vec{r})}{k_0} = \sum_i \gamma_i \frac{k_i}{k_0} s_i(\vec{r})$$

(2.19)

From this relation, $s(\vec{r})$ will gradually converge to the fundamental mode $s_0(\vec{r})$, as the higher order terms decay by the order of $O(k_i/k_0)$. 
### 2.3 Scattering interactions

The scattering of neutrons from a nucleus is divided into two classes of interactions; elastic and inelastic scattering. The treatment of these interactions is performed by considering the center of mass coordinate reference system (CM), assuming the target nucleus is stationary as depicted in Fig. 2.1:

![Collision interface at the center of mass coordinate frame.](image)

As depicted in Fig. 2.1, the collision only considers the azimuthal cosine from the direction of $\hat{\Omega}_c$. Therefore, the polar angle $\phi$ is regarded as isotropic in the CM system.

For all neutron collisions, the energy and momentum in the collisions are conserved as:

$$E' + \frac{1}{2} MV'^2 = E + \frac{1}{2} MV^2 + E_m \quad (2.20)$$

where $E'$ is the neutron incident kinetic energy, $E$ is the neutron kinetic energy after the collision, $M$ is the target nuclide mass, $V'$ and $V$ are the velocities of the target nuclide before and after the collision respectively. $E_m$ is any energy that is transferred from kinetic energy to an excited atomic state of the target.

$$m \vec{v}' + M \vec{V}' = m \vec{v} + M \vec{V} \quad (2.21)$$

where $m$ is the neutron mass, $\vec{v}'$ and $\vec{V}'$ are the initial directional velocities of the neutron and the nuclide respectively. $\vec{v}$ and $\vec{V}$ are the velocities after the collision. The movement of the CM system is taking place in a static reference system, namely the Lab-reference. The CM system moves in the Lab-reference system such that the incident
particle angle $\theta$ in the CM system coincides with the angle in the Lab system. Consequently, the azimuthal angle $\varphi$, exists in the perpendicular plane of $\theta$, as depicted in Fig. 2.2:

![Diagram of particle angle $\theta$ and azimuthal angle $\varphi$ in the CM system](image)

Figure 2.2: The stationary reference frame and the moving CM reference frame.

The CM reference frame travels within the Lab reference frame in the velocity of:

$$\vec{V}_{CM} = m\vec{v'} + M\vec{V'}$$

where $A = M/m$. Assuming a stationary nucleus and taking the velocity of the nucleus to zero, we get:

$$\vec{V}_{CM} = \frac{\vec{V}'}{A + 1}$$ 

(2.23)

where $A = M/m$. Assuming a stationary nucleus and taking the velocity of the nucleus to zero, we get:

$$\vec{V}_{CM} = \frac{\vec{V}'}{A + 1}$$ 

(2.23)

4Treatments of nuclides in motion is handled by Doppler broadening of the reaction cross sections (scattering, fission, etc.) to accommodate nuclide movements relative to the collision site. Details of Doppler broadening are omitted in this work, as the computational results in this work are derived from pre-broadened cross sections with a uniform temperature profile of 300 K. It can, however, be noted that developments on broadening techniques, utilizing delta tracking (see Sect. 3.3) to achieve memory efficient calculations are implemented in many Monte Carlo codes, including Serpent [46].
The velocity of the incident neutron in the CM frame is, therefore:

\[ \vec{v}_c' = \vec{v}' - \vec{V}_{CM} \]  

(2.24)

Figure 2.3: Lab-CM system velocity triangle.

After the collision, we have that the exit velocity of the neutron is:

\[ \vec{v}_L = \vec{V}_{CM} + \vec{v}_c \]  

(2.25)

Note that the velocity of the CM frame is constant during the collision and that \(|\vec{v}_L| = |\vec{v}|\).

From Fig. 2.3, the trigonometric products are easily obtainable as:

\[ v_L \sin \theta_L = v_c \sin \theta_c \]  

(2.26)

\[ v_L \cos \theta_L = V_{CM} + v_c \cos \theta_c \]  

(2.27)

Consequently, a division of these products yields:

\[ \tan \theta_L = \frac{\sin \theta_c}{\cos \theta_c} \]  

(2.28)

Reusing the stationary nucleus assumption, i.e. \( V' = 0 \) and combining energy and momentum balances of Eqs. (2.20, 2.21) and scattering correlations of Eqs. (2.23, 2.28) we get:

\[ \tan \theta_L = \frac{\sin \theta_c}{g + \cos \theta_c} \]  

(2.29)

where \( g \) is:

\[ g = \frac{1}{\sqrt{A^2 - \frac{E_m}{E'} A(A + 1)}} \]  

(2.30)
The cosine of the Lab-reference angle can be shown to be:

\[ \cos \theta_L = \frac{g + \cos \theta_c}{\sqrt{g^2 + 2g \cos \theta_c + 1}} \quad (2.31) \]

The scattering polar angle of the Lab system will always be smaller than that in the CM reference frame, due to the movement of the CM frame.

Since the CM polar angle is in the perpendicular plane to the vector \( \vec{V}_{CM} \) as depicted in Fig. 2.2, there is no difference between that angle in the different reference systems, therefore we simply have that: \( \varphi_L = \varphi_c \).

The resulting scattering direction of the particle in the laboratory space is transformed from the incident angle, \( \hat{\Omega}' \) to the new angle \( \hat{\Omega} \), where the coordinates in the CM system are:

\[ \hat{z}_c = \hat{\Omega}' \quad (2.32) \]
\[ \hat{y}_c = \frac{\hat{\Omega}' \times \hat{z}_L}{|\hat{\Omega}' \times \hat{z}_L|} \quad (2.33) \]
\[ \hat{x}_c = \hat{y}_c \times \hat{z}_c \quad (2.34) \]

Suppose that the incident particle has the direction in the Cartesian coordinates of:

\[ \hat{\Omega}' = \begin{bmatrix} \sin \theta' \cdot \cos \varphi' \\ \sin \theta' \cdot \sin \varphi' \\ \cos \theta' \end{bmatrix} = \begin{bmatrix} u \\ v \\ w \end{bmatrix} \quad (2.35) \]

Then the scattering direction from the collision site can be shown to become:

\[ \hat{\Omega} = \begin{bmatrix} \frac{u - uv \cos \varphi_c}{\sqrt{1 - w^2 \sin \theta_L \cos \varphi_c + w \cos \theta_L}} \\ \frac{u + uv \cos \varphi_c}{\sqrt{1 - w^2 \sin \theta_L \cos \varphi_c + w \cos \theta_L}} \\ \sqrt{1 - w^2 \sin \theta_L \cos \varphi_c + w \cos \theta_L} \end{bmatrix} \quad (2.36) \]

The new angles after the collision are easily obtainable from the direction vector as:

\[ \cos \theta = \hat{\Omega} \cdot \hat{z} \quad (2.37) \]
\[ \cos \varphi = \hat{\Omega} \cdot \hat{x}/\sin \theta \quad (2.38) \]

The elastic scattering processes undergo a reaction, where the only interaction is the transfer of kinetic energy to the target nuclide. Thus
there is only an exchange of kinetic energy between the particles. The reaction takes on the form:

\[ n + A \rightarrow n + A \]  

(2.39)

where the momentum and kinetic energy are conserved, and no excitation takes place, i.e. \( E_m = 0 \). The kinetic energy of the scattered neutron can be shown to become [32]:

\[ E = \frac{E'}{(A + 1)^2} \left[ \cos \theta_c + \sqrt{\cos^2 \theta_c + A^2 - 1} \right]^2 \]  

(2.40)

The energy restored in the particle after the collision is always lower than the initial energy assuming \( \theta_c > 0 \), and the energy decreases by increasing azimuthal angle \( \theta_c \) and decreasing nuclide mass ratio \( A \). The kinetic energy from Eq. (2.40) is only suitable for sufficiently fast neutrons, where target nuclei can be considered stationary. The implication is that the neutron can be slowed down by a sufficient number of collisions to reach below the thermal energies of the nuclides in its vicinity. If the target nuclide is a proton (i.e. a hydrogen nucleus) or another neutron where \( A = 1 \), then for angles \( \theta_c > \pi/2 \), the kinetic energy is completely transferred to the target.

The cross sections for the scattering process is varying for all materials and are typically found experimentally. For many cases, the probability distribution for \( \cos \theta_c \) can be treated as isotropic by taking the distribution uniformly for the interval \([-1, 1]\), but most cross section libraries have experimentally determined probability distributions. Actually, in the JEFF 3.3 library [23], the probability distribution for hydrogen is exactly isotropic, however, for larger nuclides, the deviation is minuscule (for \(^{238}\text{U}\), the deviation is smaller than \(10^{-14}\)%). The microscopic cross sections of a few important nuclides for nuclear reactors is depicted in Fig. 2.4:
As depicted in Fig. 2.4, the large nuclides (uranium and plutonium) have distinct resonances at the epithermal energy ranges from $4 \times 10^4$ eV to $3 \times 10^4$ eV, while small nuclides like hydrogen and deuterium have smooth cross section spectrums.

Neutrons carrying sufficient energy may excite the target nucleus by reconfiguring the nucleus configurations, transferring kinetic such that the nucleus transitions to an excited state. In the same analogy as Eq. (2.39), the reaction is now:

$$n + A \rightarrow n + A^* \quad (2.41)$$

As a consequence, some of the kinetic energy is transferred to excite the nucleus structure, thus the energy of the scattered neutron is now:

$$E = \frac{1}{(A+1)^2} \left[ \cos \theta_c \sqrt{E'} \pm \sqrt{E' \cdot (\cos^2 \theta_c + A^2 - 1) - A(A+1)E_m} \right]^2 \quad (2.42)$$
To avoid complex solutions in Eq. (2.42), we require that:

$$E' \geq \frac{A(A + 1)E_m}{\cos^2 \theta_c + A^2 - 1}$$  \hspace{1cm} (2.43)

If the condition of Eq. (2.43) is not met, then the interaction is unphysical and not applicable. To ensure this condition, the polar scattering angle $\theta_c$ may be restricted to:

$$\cos^2 \theta_c \geq \frac{E_m}{E'} A(A + 1) - A^2 + 1$$  \hspace{1cm} (2.44)

Since $\cos^2 \theta_c \leq 1$, the minimum requirement for the inelastic collision is, therefore:

$$\frac{E'}{E_m} \geq 1 + A^{-1}$$  \hspace{1cm} (2.45)

Here, we have the minimal condition of the incident energy on the condition for minimal scattering polar angle. Thus to have any physical solution to the inelastic scattering collision, the incident energy needs to be at least a factor of $1 + A^{-1}$ larger than the excitation energy.

The cross sections for inelastic scattering are typically only apparent for high energy neutrons, thus influences the neutron transport mainly for fission neutrons and in some degree epithermal neutrons. Typically, the influence of the cross sections of increasing excitation states replaces the cross sections of lower excitation states. The cross sections for the 1st and 2nd excited states of a few important nuclides in nuclear reactors is depicted in Fig. 2.5:
Figure 2.5: Inelastic cross section for selected nuclides from the JEFF 3.3 library [23] at 300 Kelvin. Note that solid lines are for the 1st state excitations, and the dashed lines are for 2nd state excitations.

As depicted in Fig. 2.5, the cross sections generally influence the transport for the increasing size of the nuclide. Actually, the cross sections are extended for decreasing neutron energies when the excitation energy $E_m$ decreases. For the nuclides in Fig. 2.5, the 1st excitation state energy increases from 77 eV for $^{235}\text{U}$, to 4.439 MeV for $\text{nat C}$.

The angular probability densities are more anisotropic than that compared to elastic collisions. For the 1st excitation cross sections in Fig. 2.5, the respective scattering angular probability densities are depicted in Fig. 2.6:
Comparing the cross sections of elastic and inelastic collisions from Fig. 2.4 and 2.5, in conjunction with the anisotropic behaviors depicted in Fig. 2.6, the fast and epithermal neutrons would experience slightly less random (or isotropic) transport. Furthermore, the fast neutrons would experience a proportionally larger transfer of their energy to nuclides per collision, considering the energy transfer to excitation states.

### 2.4 Thermal scattering

For neutrons that have been significantly slowed down by previous collisions, the kinetic energy becomes comparable to the kinetic energies of atoms in the vicinity. Thus there is also a probability that the neutron may absorb kinetic energy from striking nuclides. This event typically occurs for neutron energies less than 1 eV, matching the local thermal energies of the atoms. At such conditions, the neutron is
considered to be thermalized and the velocity distribution may be approximated by the Maxwell-Boltzmann distribution:

$$f_{MB}(E) = \frac{2}{\sqrt{\pi} \left(k_B T\right)^{3/2}} \sqrt{E} e^{-\frac{E}{k_B T}}$$

(2.46)

where $k_B$ is the Boltzmann constant, $T$ is the local temperature at the collision site and $E$ is the kinetic energy of the neutron. The probability density function $f(E)$ is normalized such that $\int_0^\infty f(E) dE = 1$.

Since the Boltzmann is accurate for a free gas model (excluding energy stored in molecular bound flexing and rotation), “thermal scattering laws” has been developed in order to provide angular scattering cross sections for materials to accommodate the inter-molecular interactions that are not comprehended by the Maxwell-Boltzmann distribution.

The thermal scattering laws are designated into three classes of scattering interactions[20]:

- Coherent elastic scattering
- Incoherent elastic and inelastic scatterings

These scattering regimes, differ fundamentally, where coherent scattering involves scattering processes without transfer of kinetic energy to the nuclide, however, the scattering angle is changed. Incoherent scattering, on the other hand, alters the kinetic energy of the neutron, with the possibility of absorbing vibrational energy from the nuclide. The coherent scattering process is important in crystal lattices where atoms are strongly bounded, such as UO$_2$ ceramics. Incoherent scattering is an important process in unstructured atomic compositions, like liquids or gases where structures are randomized.

The cross section for incoherent scattering processes is guarded by the convenient scattering law, using scattering profiles of $\hat{S}(\alpha, \beta)$ on the form:

$$\sigma_{inc}(E', E, \mu_L) = \frac{\sigma_b}{2k_B T} \sqrt{\frac{E}{E'}} \hat{S}(\alpha, \beta)$$

(2.47)

where $\sigma_b$ is the characteristic bound scattering cross section with respective contributions from a coherent and incoherent part. $\mu_L = \cos \theta_L$ is the cosine of the azimuthal angle in the Lab coordinate.

$^5k_B = 8.6173303(\pm5 \times 10^{-6}) \times 10^{-5} eV/K$, [35]
system. The momentum transfer ($\alpha$) and energy transfer ($\beta$) terms are given as:

$$\alpha = \frac{E + E' - \sqrt{EE'} \cos \theta_L}{Ak_BT}$$

$$\beta = \frac{E - E'}{k_BT}$$

The bound cross section is linear with the free cross section for a sufficiently large incident energy as:

$$\sigma_b = \left( \frac{A + 1}{A} \right)^2 \sigma_s$$

where $\sigma_s$ denotes the free scattering cross section without the influence of inter-atomic fields. The formal convention of the ENDF format is to apply the symmetrical profile of $S(\alpha, \beta) = S(\alpha, -\beta)$ about $\beta$ as:

$$\hat{S}(\alpha, \beta) = S(\alpha, \beta)e^{-\beta/2}$$

In cross section libraries (such as the JEFF 3.3 library [23]), $S(\alpha, \beta)$ values are commonly tabulated. The scattering law obeys the two conservative moment theorems:

$$\int \hat{S}(\alpha, \beta)d\beta = 1$$

$$\int \hat{S}(\alpha, \beta)\beta d\beta = -\alpha$$

For a free gas model without molecular structures which alters the scattering mechanisms, the differential profile takes the following form:

$$S_{fg}(\alpha, \beta) = e^{-\frac{\alpha^2 + \beta^2}{4\alpha}}$$

This allows scattering processes where there is a possibility that kinetic energy may increase for the scattering neutron, instead of the continual reduction of subsequent collision energy transfers imposed by the stationary nuclide assumption in Sect. 2.3.

For further investigations of thermal scattering, several sources [20, 28, 33] provide further descriptions of coherent and incoherent scattering reactions. In this section, however, the descriptions are kept at a minimum to provide only the essentials for thermal scattering. Descriptions of short collision time approximations and profile models for crystals lattices, diffusive motions, etc. may be assessed in the cited work.
2.5 Fission interactions

The angular dependence of $\Sigma_f$ in Eq. (2.12) (eg. $\hat{\Omega}' \rightarrow \hat{\Omega}$) is commonly neglected as fission neutrons are often considered isotropic. This assumption is not exactly valid considering that the prompt neutrons are emitted after about $10^{-18}$ seconds after the neutron impact on the fissionable nucleus by the rapid decay of fission fragments. Since the fission fragments are separated at about $10^{-20}$ seconds, the Coulombs repulsion allows for an acceleration of the fragments before neutrons are emitted [47]. This, along with the initial fission energy leads to neutrons distributed along the travel path of the fission fragments. This is supported by experimental observations for spontaneous fission of $^{252}$Cf [4, 5] and by fission of $^{235}$U with slow neutrons [39]. However, the studies have reported that the neutron distribution is isotropic (80-90\%) from the fission fragments after the fission separation. The angular probability densities were deduced from these studies and Fig. 2.7 depicts the distributions for a Lab reference system:
In most nuclear libraries, including JEFF 3.3 [23], the fission neutrons are emitted isotropically. The reason for this is due to very few experimental assessments on the angular and energy dependencies. The studies based in Fig. 2.7, studied fission events with either slow neutrons or by spontaneous fission. This may not be applicable for fission event for increasing incident neutron energy, as the fission scattering angle and energy distribution functions may be dependable on the incident neutron energy.

Furthermore, a study on the angular distribution of light fission fragments from a colliding neutron [26] is found to be concentrated at angles between $18^\circ$ and $78^\circ$ from the incident neutron beam direction composed of both thermal and fast neutrons. This results in a threefolded dependency on the incident neutron, light fission fraction, and prompt neutron angular distributions. Thus, the experimental basis is,
unfortunately, lacking assessments on the combined energy and angular dependencies. For experimental calculations, the angular distributions may be considered, assuming that the angular dependencies are naively independent of the incident and prompt neutron energies. Nevertheless, this complexity is commonly ignored for transport problems and the results in this thesis are assuming the isotropic fission neutron emittance.

The fission cross sections for a few important nuclides is given in Fig. 2.8:

![Fission Cross Sections Graph](image)

Figure 2.8: Fission cross sections for $^{235}\text{U}$, $^{238}\text{U}$, and $^{239}\text{Pu}$ from the JEFF 3.3 library [23].

Similarly to the scattering cross sections in Fig. 2.4, there are resonances for the epithermal neutrons. Note that the fission cross section of $^{238}\text{U}$ is about $10^7$ smaller than that of the $^{235}\text{U}$ and $^{239}\text{Pu}$ nuclides at incident neutron energies lower than $10^4$ eV. The capture cross section for $^{238}\text{U}$ is about $10^5$ larger, thus the fission contribution from this particle is negligible. However, the capture leads to decay which generates $^{239}\text{Pu}$ that have a much larger fission cross section. Thus we may
consider $^{238}\text{U}$ as a fertile fission material.

The energy-dependent probability distribution for the neutrons leaving the fission event $\chi(E)$ for $^{238}\text{U}$, and $^{239}\text{Pu}$ is given in Fig. 2.9:

![Graph showing the fission neutron energy probability distribution for $^{238}\text{U}$, and $^{239}\text{Pu}$ taken from the JEFF 3.3 library [23].]

Figure 2.9: Fission neutron energy probability distribution for $^{238}\text{U}$, and $^{239}\text{Pu}$ taken from the JEFF 3.3 library [23].

Additionally to fission events, other multiplicative reactions contribute to the criticality of a given neutron transport problem. Such reactions "knock out" neutrons from the nuclide, and may be represented as $(n,xn)$ reactions, where $x - 1$ neutrons are sputtered from the nuclide. Similarly, there are reactions emitting protons, alpha particles, etc. These reactions typically require neutron energies over 4 MeV to liberate the nucleon from the strong nuclear force in the nucleus. Similarly to fission reactions, the angular probability distribution functions are assumed isotropic. These reactions are usually not treated in direct numerical solutions to Eq. (2.9), but are commonly applied in Monte Carlo simulations.

As discussed at the beginning of this chapter, the delayed neutrons
are not considered in steady-state simulations, but Monte Carlo simulations may accommodate these neutrons as well, however releasing the neutron immediately or for a certain simulation frame in a dynamic calculation.
Chapter 3

The Monte Carlo criticality calculation method

In Monte Carlo criticality transport calculations, the approach is to solve Eq. (2.14) by power iteration, introducing consecutive solutions and iterate new corrected solutions as:

\[
s_0^{(n)}(\vec{r}) = \frac{F \cdot s_0^{(n-1)}(\vec{r})}{k_{eff}} + O \left( \left( \frac{k_1}{k_0} \right)^n \right)
\]  

(3.1)

Here, \( s_0(\vec{r}) \) is replaced by \( s_{mc}(\vec{r}) \) to represent an approximation of the fission source. Note that the Monte Carlo method only calculates the fundamental eigenmode, however as deduced in Sect. 2.2, \( s(\vec{r}) \) eventually converges to \( s_0(\vec{r}) \) in the order of \( k_1/k_0 \). Thus the Monte Carlo estimate, \( s_{mc}(\vec{r}) \) contains an inherent bias \( O((k_1/k_0)^n) \) from Eq. (3.1) decaying for each cycle as \( k_1/k_0 < 1 \).

In standard Monte Carlo criticality calculations, an approximation is performed for \( s_0^{(n)}(\vec{r}) \), with an initial guessed source of \( s_0^{(0)}(\vec{r}) \) (usually taken uniformly). The approximation makes use of a sampling procedure of the fission source neutrons, in order to collect a batch sized amount of neutrons. This process is called source normalization and is performed for each simulation cycle \( n \), such that the computational effort is approximately equal for each cycle.

\[
s_0^{(n)}(\vec{r}) \text{ is approximated as: } s_{mc}^{(n)}(\vec{r}) = \frac{F \cdot s_{mc}^{(n-1)}(\vec{r})}{k_{mc}^{(n-1)}}
\]

(3.2)

where \( m \) denotes the sampling of \( m \) neutrons from the distribution of last calculation result \( F \cdot s_{mc}^{(n)}(\vec{r}) \). When introducing the sampling, some
bias is introduced since the sampling may favor statistically insignificant neutrons. As found analytically by Brissenden and Garlick [6], the source sampling introduces a systematic error that of a Poisson distribution such that:

\[ s^{(n)}_{mc}(\vec{r}) = C^{(n)} \frac{F_{s^{(n-1)}_{mc}}(\vec{r})}{k^{(n-1)}_{mc}} + \mathcal{O}(1/\sqrt{m}) \]  

(3.3)

where \( C^{(n)} \) is a constant arising from the normalization, simply as the relative number of the source neutrons compared to the fission neutrons as:

\[ C^{(n)} = \frac{\int_{R} s^{(n)}_{mc}(\vec{r}) d^3r}{\int_{R} s^{(n-1)}_{mc}(\vec{r}) d^3r} \]  

(3.4)

where \( R \) is the spatial domain of the imposed criticality problem. Here, \( \int_{R} s^{(n)}_{mc}(\vec{r}) d^3r = m \), hence the normalization to the batch size number \( m \).

The process of sampling source neutrons is discussed further in detail in Sect. 3.2.

The difference between \( s_0(\vec{r}) \) and the iterated solution of Eq. (3.2) is analytically found [6] in the order of \( \mathcal{O}(1/\sqrt{m}) \). Hence, the unbiased solution is obtained for a sufficiently large batch size. Due to the statistical uncertainty, the Poisson distribution applies to \( s^{(n)}_{mc}(\vec{r}) \) for a bias in the order \( \mathcal{O}(1/\sqrt{m}) \) for each cycle \( n \).

Additionally, to the approximation \( s_0^{(n)}(\vec{r}) \), an approximation for \( k_{eff} \) is necessary, as this quantity is reducing the number of fission neutrons produced per fission event. The approximation of \( k_{eff} \) is also performed with an initial guess \( k_{mc}^{(0)} \), with the consecutive estimates are evaluated as:

\[ k_{mc}^{(n)} = k_{mc}^{(n-1)} \cdot \frac{\int_{R} F_{s^{(n-1)}_{mc}}(\vec{r}) d^3r}{\int_{R} s^{(n-1)}_{mc}(\vec{r}) d^3r} \]  

(3.5)

Consequently, the error of the initial guessed source, \( \mathcal{O}((k_1/k_0)^n) \) is persistent in the subsequent cycles but decays for an increasing number of cycles. The sampling is known to introduce an error in the order of \( \mathcal{O}(1/\sqrt{m}) \), which decays for increasing batch size. This error is unfortunately introduced for every cycle, resulting in a perturbation about \( s_0(\vec{r}) \). As discussed in Ch. 1, the utilization of the super-histories will avoid this bias for \( l \) super-generations in replacement of cycles. Description of the super-history method is given in Ch. 5.
For convenience, we may represent the source distribution for the MC approximation to be a vector representing the normalized distribution for a unit volume in the spatial domain \( V_i \in \mathbb{R}^3 \) with \( s^{(n)}_i \in \mathbb{R}^3 \):

\[
\begin{align*}
    s^{(n)}_i &= \frac{\int_{V_i} s^{(n)}_{mc}(\vec{r})d^3r}{\int_{R} s^{(n)}_{mc}(\vec{r})d^3r} \\
\end{align*}
\]

(3.6)

### 3.1 Pseudorandom number generation

Pseudorandom numbers are frequently applied when stochastic processes are simulated. These numbers allow for discrete sampling of these processes when simulating a single stochastic event at a time. After a sufficient number of samples, the stochastic process can produce a discrete number of outcomes that represent a continuous probability density function. This allows for discrete treatment of successive neutron interactions such as fission and scattering, instead of iterative probability densities that are tedious to compute.

There exist many methods in order to produce random numbers, ranging from recursive and multiple recursive methods to simply use prepared tabulated numbers. They typically require a seed provided by either the user or by an algorithm (created from a timestamp etc.). The linear congruential generator was originally proposed by Lehmer [29] as a sequence, using the last obtained remainder value as:

\[
x_{i+1} = (a \cdot x_i + b) \mod M
\]

(3.7)

where \( a, b \) and \( M \) are constants set by the constrained computational system or the user. It is required that \( M \geq x_i \) since \( M \) is set as the constraining number for computing the sequence, often taken as the largest integer number in a data structure. For example, an unsigned 32-bit integer number has its maximum at \( 2^{32} - 1 \), and the minimum at 0. For this case, \( M \) may be taken as \( 2^{32} \) such that the maximum value of \( x_i \) is kept in the integer range. The generation of random numbers \( \zeta_i \in [0, 1] \) is performed by the division:

\[
\zeta_i = \frac{x_i}{M}
\]

(3.8)

Note that the numbers computed in this fashion are not strictly random as the sequence is deterministic, making it possible to predict future remainders. Other random sequences may be applied, such that
the random numbers appear less deterministic, like applying a timestamp as a seed to the generator. Another method to compute the remainders is to provide the entire sequence of previous remainders as [17]:

\[ x_{i+1} = \left( \sum_{j=0}^{i} a_j \cdot x_{i-j} \right) + b \mod M \]  

(3.9)

Many other methods to generate random numbers exist, including polynomials, etc. however, only the simplest methods are presented here. A disadvantage with increasingly complex methods is the increasing computing time and memory requirements.

To ensure sufficient randomness, the uniformity may be compared to an expected probability. For example for a given set of generated random numbers the computed \( p\left[\left( i+1\right)/a < \zeta \leq i/a \right] \) may be compared to the analytical value \( 1/a \) with \( i \in [0, 1, ..., a-1] \). Small deviations of the computed probabilities from the expected probabilities ensure a sufficient uniformity of the number generator.

Furthermore, to ensure serially independence of the consecutive random remainders, a serial correlation test may be performed by calculating covariances for a given set of generated numbers with a lag, \( k \) as:

\[ \rho_k = \frac{\sum_{i=1}^{N-k}(\zeta_i - E[\zeta])(\zeta_{i+k} - E[\zeta])}{\sum_{i=1}^{N-k}(\zeta_i - E[\zeta])^2} \]  

(3.10)

where \( E[\zeta] \) is the expected value of the random number set, i.e. \( 1/2 \). If the number generator provides serially independent numbers, the calculated coefficients \( \rho_k; k \in [1, 2, ...] \) forms a normal distribution with an average close to zero and a standard deviation below \( N^{-1/2} \) for sufficiently large \( N \). If any coefficient \( |\rho_k| = 1 \), the lagged series is completely correlated and the random number generator has a period of \( k \).

A consequence of the deterministic generators is that the sequence of number is deterministic and only depends on the seed and the parameters and thus predictable. On the other hand, simulations using the deterministic methods has the ability to reproduce their results since the same random numbers are used.
3.2 Sampling methods

The selection of certain events to occur in the Monte Carlo calculations is determined by evaluating the cumulative distribution function of either a discrete or continuous distribution. This formulation is known as the fundamental formulation of Monte Carlo (FFMC) and constitutes the core of the Monte Carlo calculations. For the case of neutron transport, the events are the physical interactions that may take place in the neutron histories, however, FFMC is not limited to particle transport and may be applied to virtually any application dealing with probability distributions. The cumulative distribution for a continuous probability density is simply evaluated with an integral as:

\[ P(x) = \int_0^x p(x')dx' \]  

(3.11)

For a discrete distribution, a similar analogy is utilized, however, replacing the continuous integral with a series on the form:

\[ P_k = \sum_{i=1}^k p_i \]  

(3.12)

where \( k \in \mathbb{N}; k \leq K \) and the spectrum of events \( (i \in [1, 2, ..., K]) \) constitutes a set of possible events with corresponding probabilities \( p_i \).

The FFMC decides the selection of the event from \( x \) in continuous distributions and \( i \) in discrete distributions by a random number, \( \zeta \in [0, 1] \). When the random number is distributed uniformly, the FFMC ensures isotropic selection of the events. Therefore a cumulative set of probabilities \( \vec{P} \) is selected by the random number as:

\[ \vec{P} = \vec{\zeta} \]  

(3.13)

Note that for a continuous uniformity of \( \zeta_i, i \in \mathbb{N} \) then each random number is unique, hence \( \zeta_i \neq \zeta_j, j \neq i \).

From Eq. (3.2), the notation \( m \) essentially means the transfer of the fission neutrons created by \( s_{mc}(\vec{r}) \) to \( s_{mc}^{(n+1)}(\vec{r}) \). Consider the fission store \( S^{(n)} \) where information for all fission neutrons from \( s_{mc}^{(n)}(\vec{r}) \) is recorded. This set consists of spatial locations (\( \vec{r} \)), path directions (\( \hat{\Omega} \)), particle energies (\( E \)) and statistical weights (\( \omega \)) as:

\[ S^{(n)} = \{\vec{r}_i^{(n)}, \hat{\Omega}_i^{(n)}, \omega_i^{(n)}, E_i^{(n)}\}; \quad i = 1, 2, ..., q^{(n)} \]  

(3.14)
where \( q^{(n)} \) is the number of fission neutrons generated from \( s_{\text{mc}}^{(n)}(\vec{r}) \).

The sampling of source neutrons can be performed by primarily two methods; weight reduction or by creating a selection of the neutrons to the source. The weight reduction is the easiest approach and conserves the total weight equal to the batch number. For a certain weight threshold (less than unity), the neutrons will undergo a Russian roulette scheme where the set of neutrons are reduced when \( q^{(n)} \) is larger than \( m \). The weights for the source distribution may simply be adjusted as:

\[
\omega_j^{(n+1)} = \frac{m}{q^{(n)}} \omega_j^{(n)}
\]

(3.15)

The sum for the set of weights is obviously \( m \), hence the total weight is conserved.

A different approach to sample source neutrons is to select a set of neutrons from a fission store, the weights are assigned to unity\(^1\), however, the number of selected neutrons is kept at the batch size. One convenient procedure is to duplicate neutrons from a set where \( q^{(n)} < m \) and delete neutrons when \( q^{(n)} > m \). A simple scheme to perform this task is presented in Algorithm 3.1, where \( d_i^{(n+1)} \subset s^{(n+1)} \), and \( \vec{D}_i \subset S^{(n)} \) are the attributes of the \( i \)-th neutron in the source distribution and the fission store respectively.

\(^1\)This is only applicable to transport problems without importance regions, where variance reduction is not considered.
Algorithm 3.1 Source sampling by random duplication or deletion.

**Input:** $S^{(n)}$, $m$, $\zeta$  

**Output:** $s^{(n+1)}$  

$s^{(n+1)} \leftarrow S^{(n)}$

for $k = 1..|m - q^{(n)}|$ do

if $q^{(n)} < m$ then

$i = \text{INT}(q^{(n)}\zeta_k)$

$d_{q^{(n)+k}} \leftarrow D_i$

$q^{(n+1)} \leftarrow q^{(n+1)} + 1$

else if $q^{(n)} > m$ then

$i = \text{INT}(q^{(n+1)}\zeta_k)$

Remove $d_i$ from $s^{(n+1)}$

$q^{(n+1)} \leftarrow q^{(n+1)} - 1$

end if

end for

In Algorithm 3.1, the number of neutrons in the source distribution $s^{(n+1)}$ is simply the same as the number of fission neutrons produced by the simulation cycle, but the population is adjusted to the desired batch size. This method requires $|m - q^{(n)}|$ random numbers to be generated to a series $\zeta$.

A different approach is to create a selection of source neutrons by isotropically sample the particle indexes and duplicate the fission neutrons to the source distribution. This method is very simple and uses the random number generator extensively as shown in Algorithm 3.2:

Algorithm 3.2 Source sampling by isotropic selection

**Input:** $S^{(n)}$, $m$, $\zeta$  

**Output:** $s^{(n+1)}$  

for $k = 1..m$ do

$i = \text{INT}(q^{(n)}\zeta_k)$

$d_k \leftarrow D_i$

end for

Algorithm 3.2 may also be modified to only accept source neutrons that have not been transferred, such that $D_i^{(n)}$ is either removed from the fission store $S^{(n)}$ or that the fission neutron is prevented from transferring a second time.
Computationally, the sampling methods are comparable, however, Algorithm 3.1 is naturally more efficient over Algorithm 3.2 for large batch sizes. The reduction of neutron weights would require splitting or Russian roulette performed on the neutrons prior to the simulation of the histories. Thus, the methods differ primarily by the selection of the fission source neutron.

### 3.3 Transport and collision kernels

The probability density function for a neutron to interact with the nuclides in a given material is given as:

\[
p(r) = \Sigma_i e^{-\Sigma_i r}
\]  

(3.16)

where the probability for no collisions for a free path distance \(r\) is \(\exp(-\Sigma_i r)\), and the probability of a collision within \(dr\) is \(\Sigma_i dr\). Conveniently, the average of Eq. (3.16); \(\Sigma^{-1}_i\) is called mean-free-path (mfp) indicating how freely the neutrons move within a medium. The FFMC is simply deduced as the cumulative probability over \(dr'\) as:

\[
\int_0^r p(r')dr' = (1 - e^{-\Sigma_i r}) = \zeta
\]

(3.17)

The path length sampled by \(\zeta\) can simply be obtained by:

\[
r = -\frac{\ln (1 - \zeta)}{\Sigma_i}
\]

(3.18)

Note that the term \((1 - \zeta)\) can be substituted by \(\zeta\) since \(\zeta \in [0, 1]\) and the random number generator has an isotropic distribution.

The selection of nuclide interaction is contrary to the FFMC path length estimator, a discrete distribution. The probability density selecting interactions of \(i \in [1, 2, ..., I]\) is \(\Sigma_i / \Sigma_t\). The FFMC for selecting the \(I\)-th interaction is, therefore:

\[
P_i = \sum_{i=1}^{I} \frac{\Sigma_i}{\Sigma_t}
\]

(3.19)

Thus, the collision interaction \(n\) that is chosen for the specific collision is:

\[
\sum_{i=0}^{n} \Sigma_i > \zeta \Sigma_t
\]

(3.20)
Computationally, the interaction that is selected would be selected isotropically for a sufficient number of collision trials corresponding to the properties of the transport problem. Thus the interactions with the largest macroscopic cross section $\Sigma_i$ at a certain neutron energy are selected most frequent. Similarly, the nuclide selected for the collision in the composite material is:

$$\sum_{j=0}^{n} N_j (\sigma_i)_j > \zeta \sum_j (\Sigma_i)_j$$  \hspace{1cm} (3.21)

The same methods may be applied to sample other collision properties. For the instance for fission reactions; the number of neutrons, energies, and directions are selected from their respective probability densities, using the random numbers. However, for the number of fission neutrons, it is commonly used to control the population of neutrons for each cycle. This allows very sub-critical systems to be simulated where fission events are relatively infrequent with the same statistical accuracy as critical and super-critical systems. The selection of a sampled fission neutron population from a single fission event with a reduction of $\bar{\nu}$ is conducted as:

$$\nu = \text{INT} \left( \frac{\bar{\nu}}{k_{\text{eff}}} \right) + \begin{cases} 
  1 & \frac{\bar{\nu}}{k_{\text{eff}}} - \text{INT} \left( \frac{\bar{\nu}}{k_{\text{eff}}} \right) \leq \zeta \\
  0 & \frac{\bar{\nu}}{k_{\text{eff}}} - \text{INT} \left( \frac{\bar{\nu}}{k_{\text{eff}}} \right) > \zeta 
\end{cases}$$  \hspace{1cm} (3.22)

The reduction of $\bar{\nu}$ is the most common practice, but an alternative method may apply neutron weights to the sampled neutrons from the incident neutron $\omega'$ by $\omega' / k_{\text{eff}}$. For fairly low eigenvalue estimates, the particle splitting would, however, produce a biased population of fission neutrons opposed to the method in Eq. (3.22) since the splitting results to many neutrons with identical properties.

To move the particles in a spatial domain of heterogeneous distribution of compositions. It is necessary to track the particles such that the track estimates are kept dependent on the material regions. When a neutron crosses a boundary, (i.e. an interface of fuel material and cladding material) the path length estimator may be valued such that the neutron path crosses the interface surface, and a new path length estimate is performed for the particle placed on the path intersection of the surface. By this convention, the path length estimates are corrected for the material properties in the spatial domain. To evaluate whether a neutron track has passed a boundary,
we may evaluate the signed geometric function $\text{sgn} f(\hat{x}\vec{r}, \hat{y}\vec{r}, \hat{z}\vec{r})$, where $f(x, y, z)$ is the boundary of the material interface and $\vec{r}$ is the position vector of the neutron. If the sign changes over the track length estimate, then the neutron should be placed at the boundary such that $f(\hat{x}(\vec{r}^e + \hat{\Omega}\vec{r}_e), \hat{y}(\vec{r}^e + \hat{\Omega}\vec{r}_e), \hat{z}(\vec{r}^e + \hat{\Omega}\vec{r}_e)) = 0$ where a line search determines the neutron positioning in $0 < r_e < r$. Then, a new track length estimate with the material properties in the new interface is performed. Codes like Serpent [30], MCNP [43] etc. provides constructive solid geometry models which produce the different material boundaries where $f(x, y, z)$ may be produced by additive operations, that defines the boundaries by a series of surfaces.

Such evaluations of multiple path length estimates may require many computations until a collision is found, especially if there are many thin interfaces of different materials. Therefore, as an alternative to this method, virtual collisions may be introduced in order to reduce the number of computations required to find a collision site. Virtual collisions introduce no interactions, where the collision conserve the energy and direction from the incident neutron. The cross section for virtual collisions is added such that the total cross section is uniform for a whole domain. Thus, the path length estimate is correct for the entire spatial domain.

The virtual cross sections, $\Sigma_{\text{vir}}$ is therefore applied in the $i$-th region as:

$$(\Sigma_{\text{vir}})_i = \max_k (\Sigma_t)_k - (\Sigma_t)_i$$

Applying this for a distribution of neutron energies, the virtual cross sections from a profile for the spatial regions with different material compositions. The collision interaction would then be rejected by the probability $1 - \Sigma_t/(\Sigma_t + \Sigma_{\text{vir}})$.

The disadvantage with the delta tracking method is that for inhomogeneous regions with large variations of $\Sigma_t$, large amount computational efforts of rejecting collision interactions may dominate the total calculation time. For such cases, using the surface tracking method is more efficient. Alternatively, using a combined approach may be beneficial when both large variations of $\Sigma_t$ and many thin material interfaces are present in a given neutron transport problem.
3.4 Neutron splitting and Russian roulette

The treatment of neutron weights produced by neutron source sampling or by other means is conducted by neutron splitting or so-called Russian roulette. For variance reduction techniques, the simulation of neutron weights smaller than unity is typically applied in highly absorbing media and weakly multiplicative media. This is useful for calculating the shielding of nuclear reactors such that the mitigation is as low as reasonably achievable (ALARA).

For a region where the desired threshold neutron weight is \( \omega_{th} \), the splitting of neutrons and termination by Russian roulette is designated for two kernels in a combined algorithm:

\begin{algorithm}
\textbf{Algorithm 3.3 Russian roulette & neutron splitting}
\begin{align*}
\text{Input: } & \omega, \omega_{th}, d \\
\text{Output: } & \vec{\omega} \\
\text{if } & \omega < \omega_{th} \text{ then} \\
& \text{Perform Russian roulette:} \\
& \text{if } \zeta > d^{-1} \text{ then} \\
& \quad \vec{\omega} \leftarrow \{0\} \\
& \text{else if } \zeta \leq d^{-1} \text{ then} \\
& \quad \vec{\omega} \leftarrow \{\omega \cdot d\} \\
& \text{end if} \\
\text{end if} \\
\text{if } & \omega > \omega_{th} \text{ then} \\
& \text{Perform neutron splitting:} \\
& U = \text{INT}(\omega/\omega_{th}) \\
& \text{if } \omega/\omega_{th} - U \leq \zeta \text{ then} \\
& \quad \vec{\omega} \leftarrow \{\omega_i; \quad i = [1, ..., U + 1]\}; \text{ where } \omega_i = \omega_{th}. \\
& \text{else} \\
& \quad \vec{\omega} \leftarrow \{\omega_i; \quad i = [1, ..., U]\}; \text{ where } \omega_i = \omega_{th}. \\
& \text{end if} \\
\text{end if}
\end{align*}
\end{algorithm}

The roulette parameter, \( d \) in algorithm 3.3 may be selected to a fixed value or as a proportion of \( \omega \), and \( \omega_{th} \) etc. If it’s desirable to only perform the Russian roulette without splitting the neutron by its excessive weight after the multiplication of \( d \), then this parameter may be
adjusted each time to $\omega/\omega_{th}$. For all neutrons with zero weights is to be deleted, and all results with $\bar{\omega} = \{\omega_i; \ i > 1\}$, the neutron is to be duplicated $i - 1$ times and stored in the source neutron structure.

### 3.5 Tallying

Tallying (or counting) the source neutrons, collisions, etc. is constructing the results of a Monte Carlo simulation. Effectively, the tallying produces averages and standard deviations from a series of obtained population sizes in the desired phase spaces. There are four different techniques that are used to select the populations for this purpose [18]:

- Collision estimator
- Path-length estimator
- Surface-crossing estimator
- Analytical estimator

The collision estimator bases its population size on the neutron weight collected at the collision sites. The path-length estimator, on the other hand, collects the neutron weights at the line segment between the collisions. Conceptually, the collision and path-length estimators are the same, but the path-length estimator collects partial weights depending if the particle crosses material interfaces. Thus, the path length estimator is advantageous in conjunction with the Woodcock’s delta tracking method.

The surface-crossing estimators takes into account the angle at material surfaces that the neutron pass through. This allows estimates for neutron currents and the estimators are dependent on the crossing area rather than weights based in volumes as the collision and path-length estimators.

The analytical estimator is independent of volume or surface sites, and introduces probability sampling for composite neutron weights. The estimator includes the probability of two independent probability densities, namely the scattering angle-energy probabilities of a collision, and the collision path length. The estimator attempts to collect the weight as representative cell volumes rather than points in the spatial domain by collecting weights from these probabilities. This estimator is not discussed further in this section as the method itself is
utilizing considerable computing efforts. A more detailed description is found in Ref. [18].

For the collision estimator, the normalized collision flux in a certain phase space about \( \{ \Delta V_i, \Delta \hat{\Omega}_i, \Delta E_i \} \) for a given number of \( h \) simulated neutrons may be computed as:

\[
\psi(\vec{r}_i, \hat{\Omega}_i, E_i) = \frac{1}{h \Delta V_i \Delta \hat{\Omega}_i \Delta E_i} \sum_{r_j \in \Delta V_i \atop \Omega_j \in \Delta \hat{\Omega}_i \atop E_j \in \Delta E_i} \frac{\omega_j}{\sum_k(E_i)} \tag{3.24}
\]

where \( \omega_j \) are the weights of the neutrons collected at the collision sites during the simulation of their histories. For reaction rates of type \( k \), the estimator may expand the weight in Eq. (3.24) by: \( \omega_j \rightarrow \omega_j \Sigma_k(E_i) \) to give \( R_k(\vec{r}_i, \hat{\Omega}_i, E_i) = \Sigma_k(E_i) \psi(\vec{r}_i, \hat{\Omega}_i, E_i) \). Finding the total reaction rate involves the sum of the discrete estimates. The path-length estimator is essentially identical but collects partial weights for the line segments, \( \Delta S_i \) as \( \omega_j \Delta S_i / \sum \Delta S_i \), thus the tallied population size is larger if the crossing of material interfaces occurs frequently.

The surface-crossing estimator collects its weights, reduced by the surface normal vector, \( \hat{n} \) as \( \omega_j \times \hat{n} \cdot \hat{\Omega}' \). In the estimates of angular flux, etc., the estimates replace \( \Delta V_i \) by the interface surface \( \Delta A_i \) in Eq. (3.24), the result is now however directional angular flux where incoming flux is negative and vice versa.

After a series of independent (or overlapping) estimates for the desired reactions rates, neutron flux or neutron density, the tallying procedure produces the mean and standard deviations for the estimates. Tallying different merits independent of weights is also useful in Monte Carlo simulations. Estimations of the average \( k_{eff} \) and neutron lifetime \( l_n \) etc. can be performed, as well as their standard deviations. The cycle-wise estimates of \( k_{eff} \) are obligatory for source normalization, but quantities like \( l_n \) can be calculated on the fly with path length estimates and velocities.
Chapter 4

The Serpent Monte Carlo code

The Serpent 2 code is written entirely in the C99 standard [22] of the C programming language. The code is capable to be compiled with options for OpenMP [10] and MPI [15] for communications to multiprocessor threads and node units respectively for efficient parallel processing of computational intensive parts of the code. First, the program distributes equal partitions of the load to the available nodes, then the processes are divided into parallel computing threads in the OpenMP code to be processed multi-core processors. This allows processing to be distributed on many computers in parallel computing cores.

The MPI nodes calculate identical replicas of the MC calculations independently from the other nodes with different random number seeds. The MPI nodes then perform simulations with designated OpenMP threads. Thus the MPI nodes are independent, but the nodes orchestrate the simulation cycles with the available processing threads. Further details of the implications and actual speedup are beyond the scope of this thesis, but it’s reasonable to assume 4 parallel OpenMP threads on a single MPI node is minimally affecting the speedup, as most of the computational load can be performed in parallel [49].

For sampling the fission source as described by Eq. (3.2), the code adjusts the statistical weight of the neutrons obtained by fission from the previous cycle such that the cumulative weight is equal to the batch size. This also allows the Serpent code to be used for shielding applications and other variance reduction methods.

Other implementations are discussed in the following sections of this chapter.
4.1 Pseudo-random number kernel

The random numbers in Serpent are calculated similarly to Eq. (3.7) by Algorithm 4.1:

\begin{algorithm}
\textbf{Algorithm 4.1} Serpent pseudo-random number generation algorithm \\
\textbf{Input:} $x_i$ \\
\textbf{Output:} $x_{i+1}, \zeta_{i+1}$ \\
1: $x_{i+1} \leftarrow (x_i \cdot 2862933555777941757 + 12345) \mod 2^{64}$ \\
2: $\zeta_{i+1} \leftarrow (x_{i+1})_2 \gg 12$ \\
3: $\zeta_{i+1} \leftarrow \frac{\zeta_{i+1}}{2^{52}}$
\end{algorithm}

In line 2 of Algorithm 4.1, the integer $x_{i+1}$ is shifted 12 binary spaces to the right by the operator $\gg$, effectively reducing $x_{i+1}$ by 12 digits in the binary (base-2) number system. Creating an equivalent operation in the regular base-10 system is not possible, so conversions are necessary. The maximum value of $\zeta_{i+1}$ before the division in line 3, is then obviously $2^{64-12}$ resulting in a random number of 52-bit resolution. The operation is performed due to the double precision float number used for storage of $\zeta_{i+1}$ assigns 52 bits for decimal numbers, 11 bits for the exponent and one bit for the sign [22]. The structure of Alg. (4.1) resembles many of the characteristics of Eqs. (3.7, 3.8), but differs by the operation of shifted binary spaces and reduction to 52-bit resolution.

Computing $10^5$ random numbers, using Algorithm 4.1 with a seed value of 1517476871, produced satisfactory randomness and serial independence. The calculations of $p[(i + 1)/a < \zeta \leq i/a]$ using $a = 100$, resulted to a standard deviation of $3.41 \times 10^{-4}$ about the analytical average, 0.01. The serial correlation test, using Eq. (3.10) resulted in a normal distribution for the correlation coefficients with the average at $2.12 \times 10^{-5}$ and standard deviation at $3.23 \times 10^{-3}$ with the lag $k = [1, 2, \ldots, 10^3]$.

4.2 Tallying management

Serpent 2 allows a specified batching interval for the tallying procedure. This allows for more computing time for the particle histories, that otherwise would be occupied by tallying reactions and the cumu-
relative source distribution as the result collection may be ineffective. For an increasing number of mesh nodes to tally the responses, the computation effort to search for tallied weights increases. Thus, collecting results from several Monte Carlo cycles can reduce the computational time consumed by tallying, especially for small normalization batch sizes.

The population size for a tallying procedure is:

\[
n = \frac{[\text{Active cycles}]}{[\text{Batching interval}]} = \frac{C_a}{B} \tag{4.1}
\]

The effects of tally batch intervals would affect the resulting statistics of the source distribution as opposed to performing the tally after every cycle in the following way:

- A larger collection to compute the source distribution tally results to smaller deviations within the collection, as the relative deviation in the mesh nodes are \(1/\sqrt{N_i}\) for \(N_i\) collected source neutrons within the mesh node \(i\) boundaries. Increasing the batch interval would, therefore, decrease the inherent bias for the tallies.

- Fewer sampled source distributions to compute the standard deviation of the cumulative source as \(\sqrt{\frac{1}{C_a/B} \sum_{j=1}^{C_a/B} (S_j - \bar{S})^2}; B \geq 1; S_j \in \{\frac{1}{B} \sum_{n=(j-1)B}^{jB} s^{(n)}; j = [1, 2, ..., C_a/B]\}\)

Some considerations have to be taken into account when selecting a batching interval. The batching interval may introduce larger deviations if the sampled source distributions considerably differ. On the other hand, increasing the batching interval will reduce the bias in the tallies.

There exist some heuristic rules to select appropriate batching intervals [1]. Among the classes of methods, includes:

- Fixed batching interval
- Corrected batching intervals
- Overlapping batching intervals

The fixed batch interval remains constant during simulation cycles, while for corrected batch intervals, so-called auto-correction is applied.
Additionally, overlapping batch intervals includes estimation of source distributions merits over preceding and successive batch intervals. The overlapping scheme estimate averages and standard deviations for a larger population of histories than simulated. Selecting a certain strategy to obtain a reduced standard deviation, also have implications for memory usage. If a corrected scheme is utilized, the memory usage may significantly vary and usage of overlapping batching intervals increases the amount of sampled data on the memory.

The implications of selecting a batching interval are beyond this work, however, a constant value is to be selected such that the memory usage is somewhat constant for the given interval. The trade-off by selecting a fixed batching interval is that a limited amount of memory is allocated to store fission neutrons.
Chapter 5

The super-history method for criticality calculations

5.1 Method description

The super-history powering method briefly explained in Chapter 1, modifies the power iteration of Eq. (3.2) by performing normalization for the last super-generation. Taking the cycles as \( n \in [1, 2, \ldots, N] \) and super-generations as \( l \in [1, 2, \ldots, L] \), we may describe the source normalization and transfer of source neutrons for \( s^{(n,l)} \) as:

\[
\frac{s^{(n,l)}_{mc}(\vec{r})}{s_{mc}^{(n,l-1)}(\vec{r})} = k_{mc}^{(n)} \tag{5.1}
\]

Note that the eigenvalue is only calculated cycle-wise such that all the super-generations within a cycle use the same eigenvalue when sampling the source neutrons. Hence there is a necessity to evaluate a representative eigenvalue for the collection of super-generations within the cycle. An average of intermediate eigenvalues computed in super-generations may represent the eigenvalue for the entire cycle as:

\[
k_{mc}^{(n)} = \frac{k_{mc}^{(n-1)}}{L} \sum_{t=1}^{L} \frac{\int_{R} s_{mc}^{(n-1,l)} d\vec{r}^{3}}{\int_{R} s_{mc}^{(n-1,l)} d\vec{r}^{3}} \tag{5.2}
\]

Other evaluations of the cycle eigenvalue are possible, for example, a \( L \)-th root of the cumulative product or using a median, etc.

For the source normalization for the last super-generation, \( l = L \), the source normalization returns to the sampling methods to readjust
the total number of source neutrons to the batch number. Otherwise, the eigenvalue is calculated for all cycles and super-generations and generally makes no distinction between the evaluations.

The theoretical advantage of performing super-generations is that the systematic sampling bias is taken into consideration between the super-generations. The error introduced by sampling was as shown in Eq. (3.3) that of the Poisson distribution, that is circumvented by the super-generations. Hence for a decreasing batch number, the bias is increasing for cycle-wise source neutron transfers. Therefore, the super-generation method has an increasing advantage compared to conventional power iteration for a decreasing batch number.

Increasing the batch number will limit the advantage of the super-generations, but the super-histories introduce virtually no additional computations that may increase the computational efforts. Thus, no theoretical disadvantages are introduced by this method.

On the other hand, a disadvantage is that too many super-generations in a cycle may alter the number of source neutrons in the last super-generation significantly compared to the initial generation. The concern is that if the last generation produces a significantly lower population than that of the initial generation, the bias introduced when normalizing the source for the next cycle is of a higher order of magnitude compared to the conventional cycle routines. Since the eigenvalue perpetuates around the exact value, it’s likely that the number of last super-generation neutrons are periodically larger and smaller than the batch size.

For the event of a considerably smaller number of last generation neutrons compared to the first generation, the normalization sampling of source neutrons will introduce a larger bias compared to the opposite case. Since there are fewer neutrons to sample from, the source neutron distribution will consist of many neutrons with the same properties. For example, after a sufficient number of super-generations, the source distribution may only produce a single fission neutron. Thus there will be a sampled batch size number of neutrons at the same location with the same directions and energies. This source distribution is not resembling a physical neutron distribution and would generate a significant deviation.

This disadvantage may be limited to some degree by increasing the number of super-generations such that the applied eigenvalue in the normalization is relaxed. However, increasing the number of genera-
tions also leads to a greater risk to reduce the subsequent number of neutrons if the eigenvalues were overestimated by insufficient sampling. Thus the number of neutrons may still avalanche to a considerably smaller size than that of the sampling batch size.

5.2 Previous studies

As discussed briefly in Ch. 1, the super-history powering method has already been tested and implemented in production versions of MONK [40]. An independent implementation in MCNP is available as a modification for the purpose of variance reduction [7, 8]. Additionally, an asymptotic super-history method has been implemented in the RMC code [38].

The earliest work on the super-history powering method was focused on chaotic fission source convergence problems, where the eigenvalue was not guaranteed to converge to the correct value. The problems of interest were spatially coupled reactivity systems separated by void (or air) where the collective eigenvalue is expected larger than the largest eigenvalue of the independent systems. Whitesides [51] proposed the convergence problem of a coupled system of plutonium spheres in a 9x9x9 Cartesian lattice configuration, reflected by a layer of water outside the lattice. The center sphere was of critical mass surrounded by a sub-critical lattice of smaller spheres. Thus the interest was to converge the eigenvalue for a complex spatial geometry where the anticipated value is larger than unity.

Initially, the super-history powering method was tested for a configuration of a supercritical Plutonium sphere of radius 4 inches placed in a storage room, where the walls had cylindrical holes filled with fissile materials [6]. The sphere and the storage room was found to have eigenvalues at 1.0171 ± 0.0056 and 0.8258 ± 0.0062 respectively after a few thousands of neutron histories. Simulating the combined system with conventional powering iteration for a batch size of 200 neutrons over 137 cycles resulted in an eigenvalue of just 0.8261 ± 0.0195. Bris Kendall and Garlick[6] found that the normalization process had maintained unrepresentative fission neutrons which resulted in the failing result. They further investigated that simulating for 5 and 10 supergenerations using the MONK code [31] increased the eigenvalue as well the uncertainty despite the total number of simulated neutron his-
tories increased together with the number of super-generations used for each cycle. The result became acceptable for 10 super-generations with a resulting eigenvalue of \(1.0366 \pm 0.0111\) but increased the number of simulated histories from 1900 to 45033 neutrons compared to regular power iteration.

Blomquist [3] found that the super-history powering method implemented in MONK was similarly advantageous when converging the fission source for the Whitesides \(k_{\text{eff}}\) of the world problem [51]. Intuitively, the eigenvalue for the system is greater than unity since there are spheres surrounding the critical sphere, providing the center sphere additional neutrons. However, it was found that Monte Carlo codes with a limited batch size do not reflect this intuition due to bias introduced by sampling source neutrons and returns sub-critical results. Stratified sampling [16] was also examined for the problem [34] and were found to only require 75\% of the batch sizes compared to conventional sampling to obtain the same eigenvalue within a 95\% confidence interval. However, the computational effort is increased when utilizing stratified sampling. It was reported that the computational effort for stratified sampling leads to twice the computational time as conventional sampling [3]. The super-history powering method, on the other hand, skips sampling altogether except when transitioning between simulation cycles. The only deficiency is that the subsequent super-generations may have significant deviations in population sizes compared to an initial cycle, especially if the eigenvalue is severely overestimated or underestimated, leading to smaller or larger populations respectively.

Blomquist [3] used the MONK code with 10 super-generations per cycle, 6 neutrons initially distributed for each sphere, arising to a batch size of 4374 neutrons. 150 generations were used for collecting statistics for the eigenvalue and 30 generations were used for converging. The experiment was repeated 20 times with different random number seeds for comparing the super-history and stratified sampling methods respectively. Blomquist found that the super-histories significantly improved the eigenvalue convergence. The correct eigenvalue for the Whitesides problem is in the range of 1.01-1.02. By the average of the 20 simulations, the super-history powering method produced an eigenvalue of 0.9931 which was 0.0049 larger than that of regular power iterations for the same configuration. Although the stratified sampling produced a better result of 0.9955, the computational time was dou-
bled compared to conventional power iterations and the super-history method.

In a later study by Blomquist and Gelbard [2], the super-history method was compared to the conventional power iteration method on homogeneous and inhomogeneous one-dimensional slabs to investigate the effect on absorption rate and eigenvalue convergence. For the selected problems, the super-history method had comparable efficiency as conventional power iterations. The study included a simplified simulation kernel, assuming single speed neutrons with determined reaction cross sections. For the homogeneous test case, the width was selected to 15.5 cm with reflective boundary conditions using 3 equal sized volume cells to record absorption rates in the respective cells was performed. The simulation was replicated 1000 times with different seeds to the random number generator for each simulation using 100 generations. An identical inhomogeneous slab with different fission cross sections for the volume cells facing the reflective boundaries was selected smaller than that of the central volume cell while keeping $\Sigma_t$ uniform for the slab.

For both test cases, no significant improvement was achieved likely due to the coarse spatial mesh and that the small dimension of the slabs leads to a highly coupled neutronic system allowing for rapid decaying initial source bias $i$. The distinction between the homogeneous and inhomogeneous test cases is that the neutron source is skewed to the center of the slab, but the neutronic coupling is comparable. Thus the super-history method is equally efficient as conventional power iterations for highly coupled neutronic systems with small dominance ratios.

She, Wang, and Yu [38] developed the asymptotic Wielandt’s and super-history methods respectively to study the neutron source convergence. For the super-history method, the cycle batch sizes were split into an adjusted initial batch size for the first super-generation such that a single cycle expects to simulate approximately the same number of neutron histories. i.e.:

$$m_{\text{adj}} = \frac{m}{L} \quad (5.3)$$

$\Sigma_t = 1cm^{-1}$ and $\Sigma_a = 0.1cm^{-1}$. By utilizing the diffusion approximation for vacuum boundary conditions on the 15.5 cm homogeneous slab and assuming a uranium composited slab, Eq. (6.12) returns a dominance ratio of about 0.76. This indicates a strongly coupled neutron transport problem.
She, Wang, and Yu argued that using a large $L$ leads to a faster fission source convergence, but involves large fluctuations in the eigenvalue estimations and may not guarantee a correct source distribution. Therefore, they argued that super-histories are useful for achieving rapid convergence, but the number of super-generations has to be reduced in order to achieve correct convergence. They introduced the asymptotic super-generation cycles, where $L$ was gradually reduced for each incrementing cycle number $p_i$ as:

$$p_i = \begin{cases} \frac{n_{in}}{a^c} & i = 1 \\ 10^{-1} & 1 < i \leq c \end{cases}$$

(5.4)

$$L_i = a^{c-i}$$

(5.5)

where $a$ and $c$ are the asymptotic rate and order respectively, and $n_{in}$ is the number of inactive cycles necessary to converge with conventional power iterations.

She, Wang, and Yu concluded that this method would converge the fission source faster than regular power iteration (reducing the computational time by $\sim 80\%$) and that the conventional super-history method would not improve the acceleration rate due to severe fluctuations. This conclusion is contradictory to the results found by Blomquist [3], where the super-histories was advantageous when converging to a final state (since $k_{eff}$ convergence in their very chaotic system indicates fission source convergence to a certain extent).

The disadvantage of the approach used by She, Wang, and Yu is that the adjustment of batch sizes in Eq. (5.3) leads to larger normalization biases, however less frequent during the simulations of super-generations. Thus, compared with cycles with no super-histories applied, the systematic bias is much smaller, but more frequently applied. Furthermore, the authors have not specified the neutron source sampling routines, but it’s assumed that they used the default method from the RMC Monte Carlo code. Unfortunately, no description of this is given in their in their publications [38, 48]. Presumably, the sampling routine is similar to the Serpent code by adjusting weights and perform particle splitting or Russian roulettes on the neutrons.

The reported acceleration of source convergence is partially contributed by the growing number of batch neutrons that is well documented [14]. Thus the benefits of the super-histories themselves were not proven in the studies conducted by She, Wang, and Yu.
Other studies have acquired benefits by using the super-history method, including variance reduction [7, 8] and computing eigenvalue sensitivity coefficients for cross-sectional data [37].
Chapter 6

Numerical test setup

6.1 Acceleration criteria

The measure of efficiency for Monte Carlo codes is formally the figure of merit (FOM) composed of the total computational time $T$ and the variance as:

$$FOM \equiv \frac{1}{\epsilon^2 \cdot T}$$

(6.1)

where $\epsilon$ is the standard deviation of the targeted quantity. Conventionally, the inverse of FOM is a merit for the ineffectiveness of the code and it’s desirable to minimize both the time and the variance.

Most of the computational time is utilized for computing neutron paths, tracking neutrons within surfaces, collisions, and random numbers. Additionally, a significant amount of time is also used for storing and computing statistical merits as mean and standard deviations for all the interactions as well to compute the fission matrix. Other factors that affect the computational time is memory and multi-core processor management. These factors may vary between computers and the efficiency can be significantly affected by the algorithms used for memory access as well for sampling and generation of random numbers. To provide reproducibility, a convention is to consider that the computational time is correlated to the total number of neutron histories simulated, i.e. $T \propto h$. This leads the neutron history based FOM as:

$$FOM_h = \frac{1}{\epsilon^2 \cdot h}$$

(6.2)

From this relation, $FOM_h$ is indirectly determined by the batch size, the number of cycles and the number of neutron generations. Thus,
FOM$_h$ is evaluated independently from the computing speed. This is useful for comparing the different acceleration methods across different computer systems without the bias from the computing efficiency and speed.

Tuttelberg and Dufek\cite{44} deduced the optimal batch size, requiring the initial relative error and the dominance ratio:

$$m_{opt} = \sqrt{\frac{h}{\epsilon_0} \left( 1 - \frac{k_1}{k_0} \right)}$$  \hspace{2cm} (6.3)

where an estimation of the initial error $\epsilon_0$ can be taken as:

$$\epsilon_0 = \frac{\| \bar{s}^{(0)} - \bar{s}_c \|}{2}$$  \hspace{2cm} (6.4)

where $\bar{s}_c$ and $\bar{s}^{(0)}$ is the cumulative (or converged) and the initial neutron source distributions, respectively. The vector form of the source distributions may be evaluated element-wise as the normalized distribution by:

$$s_i^{(n)} = \frac{\int_{V_i} s^{(n)}(\vec{r}) d^3r}{\int_{R} s^{(n)}(\vec{r}) d^3r}$$  \hspace{2cm} (6.5)

Following that $\bar{s}^{(n)} = \{s_i^{(n)}; i \in \mathbb{N}\}$, represents the volume fraction of the total neutron source distribution. The vector thus contains the node solution to the transport problem at cycle $n$, while $\bar{s}_c$ contains the average of all the source vector, thus representing the cumulative fission neutron source.

The evaluation of an optimal batch size also requires an estimation of the dominance ratio, which may be performed by either the noise propagation \cite{42}, coarse mesh projection \cite{36} or coarse mesh finite difference \cite{19} methods. Evaluations of the dominance ratio\footnote{A diffusion approximation for the source convergence problem in Sect. 6.3 is performed, however, this approach is analytical and unaffected from statistical bias introduced in the numerical methods.} and the optimal batch size is beyond the scope of this work. Since the comparison of conventional power iterations and super-generations is to determine whether the super-histories contributes to an acceleration of FOM, we may expect similar performance with optimal batch sizes.
6.2 Super-history implementation in Serpent 2

The Serpent 2.1.30 source code was used as a basis for testing the super-history method. There are some shortcomings this code produces. Mainly, there is a chance that no fission neutrons may be produced in a cycle or generation, in which the execution terminates with an error message. Another deficiency is that the code does not consider the fact that a considerably small number of fission neutrons introduces a bias larger than that of the systematic sampling bias. The Serpent code is therefore reliant that the software users select reasonably large batch sizes, such that the chance of zero fission neutrons is negligible, and that the introduced bias of sampling fewer fission neutrons are diminishing.

The super-history method was implemented such that the super-generations are essentially treated in the same manner as regular simulation cycles except that the normalization is skipped until the $L$-th generation is completed. The results are therefore collected from all the super-generations and not just for the $L$-th super-generations. The $k_{eff}$ values for each super-generation are stored until the computation of the average estimation with non-zero contributions is included. This leads to larger tally populations, but with a larger influence of the bias introduced by the source neutron sampling. This approach may differ from that conducted in the previous studies presented in Sect. 5.2, as they have not described the applied methods. Additionally, the computational trade-off by simulating super-histories diminishes as more neutron histories have to be simulated to compute the standard deviation of the spatial source distribution to the same accuracy. Although the result introduces larger biases, the resulting bias of the source neutron distribution is decaying from the initial source, and at a converged state, the bias is only periodically introduced. This results to a source distribution minimally disturbed the neutron source sampling. This approach saves computational time since the result from all the neutron histories are collected, compared to only collecting the last super-generation histories. For convenience, cycles are now referred to a collection of $L$ super-generations for the rest of this thesis.

The Serpent code does not have any guards to the chance of termination of neutrons due to no generation of fission neutrons (die out).
Thus, an implementation of reusing the previous neutron source distribution was conducted. This also allows the code to be used for arbitrary small batch sizes and large super generation numbers without halting the simulation.

Practically, this implementation was performed by creating duplicate neutrons in a saved source structure. When the saved structure is not needed to restore for zero fission neutrons, the neutrons are simply transferred to the source distribution with infinitesimal weights of $10^{-37}$ such that the neutrons are eliminated by the Russian roulette scheme. Then the new source is duplicated to the saved source structure. When the fission store is empty after a cycle or a super-generation, the neutrons in the saved source structure is simply duplicated to the source distribution such that the saved source structure may be reused if die out occurs consecutively. It’s preferred that the redundant neutrons are directly removed from memory, instead of occupying computing resources. However, this implementation was found to be the easiest to perform.

In the events of neutron die-outs, the average eigenvalue is contributed by a value of 0. This effectively reduces the effective eigenvalue calculation, but the value is not used for reducing $\bar{\nu}$. The zero value is however included in the calculation of average and standard deviations. If all super-generations within a cycle returns eigenvalues of 0, then the average eigenvalue from the previous cycle is assigned.

To avoid large sampling errors as discussed in Sect. 5.1, its preferable to save the previous source of which the number of fission neutrons in the last super-generation is larger than the batch size. Using this source for normalization introduces an error smaller in order of magnitude than the last obtain source distribution. Therefore we may introduce the previous distribution (backup source) under the following condition:

$$\frac{\mathcal{O}(\frac{1}{\sqrt{m_n}})}{\mathcal{O}(\frac{1}{\sqrt{m_B}})} \geq \tau$$

(6.6)

where $m_n$ is the number of fission neutrons from the $L$-th super-generation for a cycle, $n$. $\tau$ describes the maximum tolerated sampling bias ratio of the systematic batch sampling of $m_B$ neutrons. Assuming a constant bias term: $\mathcal{O}(\frac{1}{\sqrt{m}}) \sim C \frac{1}{\sqrt{m^2}}$, we may approximate the condition to

$^2$The formal definition of the truncation error is $|\mathcal{O}(f(x))| \leq C f(x)$. Thus the error perturbs about the continuous function $f(x)$ with a maximum amplitude of $C$. 

which the previous source from a previous cycle is to be introduced:

\[ m_n \leq \frac{m_B}{\tau^2} \]  \hspace{1cm} (6.7)

Thus, if the requirement is to not double the truncation order of the sampling bias, it’s sufficient to require only 25% fission neutrons of that in the batch size.

The multiplication eigenvalue, \( k_{\text{eff}}^{(n)} \) is produced for each cycle as an average of the estimate during the super-generations. A wide range of possible alternatives to produce a representative eigenvalue may be used, including median, weighting rules, and product rules. For this work, only a simple averaging without weights is used. To further improve the estimate, excluding contributions where the eigenvalue estimate is zero due to the die-out of neutrons is beneficial as the estimate is not physically possible in neither critical or sub-critical systems where an eigenvalue is expected larger than zero.

To investigate how the simulation is dependent on the neutron source backups, a utilization factor is computed for the simulations to:

\[ BU = \frac{\text{[Number of backups used]}}{\text{[Number of backups used]} + \text{[Number of backups created]}} \]  \hspace{1cm} (6.8)

The backup utilization factor \( BU \) provides useful diagnostic values as well as the neutron source population size statistics for the super-generations.

An additional component to the code was added, allowing the user to provide a reduction parameter, \( p_r \) to reduce the number of neutrons produced in fission in Eq. (3.22) as: \( \bar{\nu}/k_{\text{eff}} \rightarrow \bar{\nu}/(k_{\text{eff}} \cdot p_r) \). This allows for some degree of population avalanche control. For \( p_r < 1 \), the number of simulated histories is expected to increase for each super-generation on average. Its advisable to utilize the reduction parameter sparingly, as the total number of simulated histories will significantly deviate from the intended number.

### 6.3 Source convergence problem

The desired problem to test the super-history method is for a one-dimensional slab, of which there is a homogeneous distribution of fissionable and absorbing materials. The motivation for this is that the
homogeneous problem is weakly coupled for a large dominance ratio. Compared to fissile material separated by a strong neutron absorbing region, the neutronic coupling is only modified by the slab thickness. Furthermore, the separation of fissile materials may also lead to a nearly complete separation of the neutronic coupling, where the converge of the source neutron distribution becomes difficult to converge depending on the initial distribution.

From single speed diffusion theory, the eigenvalues of a 1D-slab can be approximated by:

\[ k_n = \frac{\bar{\nu} \Sigma_t}{\Sigma_a + DB_n^2}; \quad n \in \mathbb{N} \quad (6.9) \]

where the buckling factor, \( B_n \), is:

\[ B_n = \frac{(n + 1)\pi}{\tilde{L}} \quad (6.10) \]

where the extrapolated length \( \tilde{L} = L + 2 \cdot 0.7104\lambda_{tr} \) consist of the slab length \( L \) and two times the extrapolation length. \( \lambda_{tr} = \Sigma_t^{-1} \) is the mean-free-path length in the slab. The diffusion coefficient \( D \) is taken as:

\[ D = \frac{1}{3[\Sigma_a + \Sigma_s(1 - \frac{x}{3A})]} \quad (6.11) \]

where \( A \) is the atomic number of the target nuclide.

Combining Eqs. (6.9, 6.10), we may obtain the dominance ratio of:

\[ \frac{k_1}{k_0} = \frac{D\pi^2 + \tilde{L}^2\Sigma_a}{4D\pi^2 + \tilde{L}^2\Sigma_a} \quad (6.12) \]

For a sufficiently large slab, where \( L >> \pi \sqrt{D/\Sigma_a} \), the dominance ratio is approaching unity. Oppositely, the dominance ratio approaches 1/4 when the slab becomes infinitely thin.

The problem in interest is taken from Dufek[13], where the sampling bias has a strong influence on the converged neutron source distribution. The problem consists of a slab with a length of 10 meters and vacuum boundary conditions are applied at the slab interfaces, such that the escaping neutrons are not contained. The material used in the slab is a mixture of \(^{235}U\), \(^{238}U\), and carbon with compositions as given in Table 6.1:
Table 6.1: Mass densities of nuclides in the test slab with a length of 10 meters and vacuum boundary conditions.

<table>
<thead>
<tr>
<th>Nuclide</th>
<th>Mass density (g/cm³)</th>
</tr>
</thead>
<tbody>
<tr>
<td>$^{235}$U</td>
<td>0.14</td>
</tr>
<tr>
<td>$^{235}$U</td>
<td>1.0</td>
</tr>
<tr>
<td>natC</td>
<td>1.0</td>
</tr>
</tbody>
</table>

The dominance ratio of the slab may be calculated for the slab using cross sections from the JEFF 3.3 library [23] at 300 K and assuming thermal neutrons at 0.026 eV. This indicates a dominance ratio of 0.9996 by using Eq. (6.12), which suggest that the bias from an initial guessed neutron source distribution decays relatively slowly for each cycle. The length of the slab is large compared to the mean free path (1.9 cm for thermal neutrons), which means that the movements of neutrons are very restricted considering the large spatial domain. This allows simulations with a certain batch size to produce a theoretical persistent bias in the converged source as the bias slowly decays to a bias proportional to the combined order of source sampling and dominance ratio decay.

The guessed initial source distribution for the slab problem is taken as a uniform distribution through the axial direction of the slab. The temperature profile is taken uniformly at 300 K, such that Doppler broadening is only necessary for preparing cross section tables. Since the slab problem consists of a single material region, no delta tracking is necessary. The neutron weight threshold is set to unity, such that particle splitting and Russian roulette schemes ensure that the population size is equal to the batch size. Thus the problem allows for efficient simulations of neutron histories with a minimum of necessary sampling processes.
Chapter 7

Results

The slab experiment in Sect. 6.3 was tested for two batch sizes in interest, namely 20 and 100 neutrons. These two batch sizes would pose different characteristics of the super-history powering method as there is a much larger systematic sampling bias for 20 neutrons, but lower resolutions for statistical results, such as the eigenvalue and neutron source sites. On the other hand, the batch size of 100 neutrons provides better statistical results concerning the cycle eigenvalue averaging and hence better control towards population avalanche.

The maximum theoretical error allowed by the normalizations were set to 1.05 of the systematic bias. This would provide consistent sampling bias such that if the last generation fission neutrons are sufficiently lower than that of the normalization batch size, the bias remains consistent. The reuse of a previous source distribution on the other hand results to a prolonged bias originating from the initial neutron source distribution, and possibly over-represent certain source distributions from the last super-generations.

The mesh configuration was selected to 500 nodes along the axial direction of the slab, such that the pitch between the nodes is 2 mm. The meshing is assumed to sufficiently capture the tallies such that the tally standard deviation and distribution profile is acceptable. The seed for Algorithm 4.1 was selected to 1517476871 for all the simulations.

The number of active cycles for the test calculations were selected such that the total number of simulated histories in these cycles was approximately $10^9$. This would result in a sufficient population with to produce acceptable tally standard deviations. The batching interval
for the test calculations is taken for $10^5/m$ cycles. Thus for the batch size of 100 neutrons, $10^6$ active cycles were calculated, using a batching interval of 1000 cycles.

A reference calculation was performed, using a batch size of $5 \times 10^5$ neutrons and simulating $10^{10}$ histories. The reference batching interval was selected at every 50th cycle, however, the batching interval is utilized primarily to reduce the bias for the tallies rather than computing speed. Furthermore, to converge the source distribution, the number of inactive cycles was taken as 10% of the active cycles where tallied neutrons were disregarded.

All calculations were conducted on 4 allocated cores on an Intel Core i9-7940X CPU chip at 3.10GHz on an Ubuntu 17.10 operating system. The modified Serpent code was compiled by the GNU Compiler Collection (GCC), version 7.2.0 compiler with the default optimization options in the makefile distributed with the Serpent source code. The memory (random access memory) was manually allocated to the experiments ranging from a few hundred megabytes to about 20 gigabytes, allowing a few simulations to be executed simultaneously. The total amount of available memory on the test computer was, however, 62.6 gigabytes.

The converged neutron source distributions without the use of super-histories are depicted in Fig. 7.1. The reference calculation with a batch size of $5 \times 10^5$, have a sufficiently large batch size that the normalization bias is nearly negligible, and its source distribution is considered the correct distribution. Arguably, using super-histories for the reference calculation may improve the distribution, however, the calculated distribution is sufficient to compare the method for smaller batch sizes.
In Fig. 7.1, the neutron source distributions are normalized such that the total number of neutrons in the slab is at unity. Thus the neutron density is naturally smallest at the vacuum boundaries, where multiplicity is low due to leaking neutrons. For a physical slab, the total neutron population of the slab may be multiplied by the distribution to find the actual neutron density. The total population in the slab is unknown, but it’s not required for the source convergence since the problem is not an initial value problem.

As depicted in Fig. 7.1, the decrease of batch size results in a flattening of the neutron source. This is due to insufficient statistics that the low population provides, as the multiplicity is uniform throughout the slab when leakage is not taken into account. The increase of batch size leads to a distribution where leakage is taken into account. In the simple case of a batch size of a single neutron, the distribution would exhibit a uniform distribution since the single neutron and its successors would travel in a random motion.

The batch sizes of 20 and 100 neutrons are devoted to individual sections, as the effects of the super-generations are very different for these experiments. The batch size of 20 neutrons would have a theo-
retical systematic sampling bias 2.24 larger than that of 100 neutrons. The batch size of 20 also poses different problems regarding the stabil-
ity of the super-history estimates.

The reference error was calculated from the reference calculation as:

\[ RE = \int_{x=-5m}^{5m} |s_{m,L}(x) - s_{ref}(x)| \, dx \]  

(7.1)

where \( s_{m,L}(x) \) is the source neutron distribution for \( m \) batch neutrons and \( L \) super-generations. Computationally, the integral in Eq. (7.1) was evaluated numerically for each mesh node by the simple mid-
point rectangle rule. The relative improvements in the experiments was calculated as:

\[ RI = \frac{|RE_{m,L} - RE_{m,1}|}{RE_{m,1}} \]  

(7.2)

The relative improvement was also calculated for the figure of merits, in the same fashion with \( RE = FOM_{m,L} \). To assess the uncertainty of the obtained average values, additive treatment was conducted for the variances in order to produce standard deviations.

### 7.1 Batch size of 100 neutrons

For the batch size of 100 neutrons, a series of experiments with up to 50 super-generations were tested. The neutron source distributions up to 40 super-generations were successful and provided a linear de-
crease of the bias to the converged source distribution. Furthermore, two tests were conducted with 50 super-generations to investigate the convergence, by modifying the \( k_{eff} \) used in sampling the fission neu-
trons.

The source distribution convergence effectively reduces bias as the systematic normalization error is only introduced at the interval of the number of super-generations used in the experiments. In Fig. 7.2, the increasing number of super-generations leads to the distribution to peak in the center of the slab. As discussed, the correct distribution tends towards the regions where the leakage effect is the smallest (and regions of high enrichment). Thus, the theoretical framework of the super-histories succeeds to reduce bias to the distribution in experi-
mental means.
As observed in Fig. 7.2, the neutron distribution is slightly skewed to the left-hand side. This may be contributed by the large dominance ratio of the slab. The utilization of backups is depicted in Fig. 7.4, however, the usage of backups is periodic, such that the fission source stored in the backup is periodically changed.

The normalization bias was shown to be reduced by the super-histories, using the source deviation from the reference calculation. The relative improvement was found to increase linearly from 5 super-generations, as depicted in Fig. 7.3:
As observed in Fig. 7.3, the source convergence bias is linearly decaying for the increasing number of super-generations. It’s anticipated that this decay would be limited since the improvement from the super-histories is reduced by the periodic normalization bias. Finding such a limit for this linear decay is beyond the scope for this work, but as the increase of super-generations is introduced, the normalization sampling is reintroducing the bias that is included in the tallying procedure to the first super-generation fission neutrons. Thus, the bias persists to some degree, despite the number of super-generations that are applied.

The performance of the super-histories is presented in Fig. 7.4, and shows that the implementation of the super-histories is decreasing the computational time, but increasing the total number of histories simulated. Similarly, the utilization of the backup source distribution is increased by the influence of the super-histories.
Figure 7.4: Simulated neutrons histories, CPU hours and backup utilization factor (calculated by Eq. (6.8)) for a batch size of 100 neutrons.

The computational time in Fig. 7.4 is dependent on three effects. Firstly, when the number of total simulated neutron histories increases, naturally the computational time increases. Secondly, the elimination of infinitesimal neutron weights when the backup source is updated are occurring less frequent as the number of super-generations increases. Lastly, the backup utilization is increasing for an increasing number of super-generations, thus the reuse of backup sources leads to a slightly smaller computational effort since the elimination of neutrons in the last super-generation is less computationally intensive than the larger backup source.

To evaluate the performance of the super-histories, the figure of merits may indicate the computational improvement of the super-history method, taking into account computing time and the source distribution error. The calculations of figure of merits are depicted in Fig. 7.5, together with their respective improvements calculated by Eq. (7.2):
Figure 7.5: Figure of merits for a batch size of 100 neutrons for different numbers of super-generations, with standard deviation bars.

The improvements of $FOM_h$ is consistently smaller than that of $FOM$, due to the effects discussed previously in this section. Thus, $FOM_h$ may provide a conservative estimate of the improvement compared to $FOM$, since the computational time was found to decrease in some extent for increasing number of super-generations. As observed in Fig. 7.5, the improvement of 40 super-generations is just above 200% for the $FOM_h$ estimate and about 360 % for the $FOM$ estimate. Other implementations of the neutron storing management for the backup and saved neutron sources may decrease the $FOM$ improvement, hence the $FOM_h$ does not take implementation efficiencies into account. Thus future studies may observe $FOM$ tend towards $FOM_h$ in the improvement evaluations.

Additionally to the utilization of backup sources, the degree of the avalanche of the neutrons has been found to increase for an increasing number of super-generations. As depicted in Fig. 7.6, the neutron population is increasingly varying by increasing super-generations:
Figure 7.6: Neutron source population size statistics for a batch size of 100 neutrons for different numbers of super-generations. Here, $\langle m \rangle$ is the average source neutron population for all super-generations, and $\sigma_m$ is the population standard deviation.

The disadvantage of the implemented eigenvalue averaging procedure became apparent at 50 super-generations due to the avalanche towards zero neutrons. The avalanche leads to very high estimates since non-zero contributions are used for the average. Thus when source populations of 1-2 neutrons produce fission events, the contribution to the average eigenvalue is larger than expected from a source population size of 100, since the contribution is statistically insignificant. This disadvantage may have been reduced by including zero contributions when no fission neutrons are produced. However, that would introduce another disadvantage that may avalanche the neutron populations to severely large numbers in the next cycle if there are relatively few generations producing fission neutrons in the same scenario.

Two cases were evaluated for 50 super-generations where the parameter $p_r$ was set to 1.0 and 0.98 for the respective test calculations. The results of the experiments are given in Table 7.1:
As the results suggest, the super-history method is limited when a high number of super-generations are utilized. Without controlling the criticality, the eigenvalue eventually becomes over-evaluated, due to the die-outs and re-usage of the source distribution of the previous super-generation. This becomes apparent as the backup utilization is nearly 100% as well that only ~6% of the intended number of neutrons were simulated. Trying to skew the avalanche away from die-outs, does not improve the source distribution error compared with using 40 super-generations as observed from Fig. 7.3. Furthermore, the skewing leads to almost a doubled number of the intended neutron histories to be simulated. The maximum recorded population size was also large, by about 3 times the simulations with 20-40 super-generations. This skewing approach has not produced better results and uses considerably more computational resources.

### 7.2 Batch size of 20 neutrons

The batch size of 20 neutrons was shown to provide little stability to the super-histories. Compared to the batch size of 100 neutrons, the expected number of fission events per generation is only 20%. Thus, there is a larger likelihood for the avalanche towards zero fission neutrons for the same number of super-generations.

Only two experiments, using 2 and 3 super-generations respectively was successful and achieved completion. Experiments using 4 or larger number of super-generations failed to complete the simulations due to segmentation faults triggered by a large neutron source population. This occurs when the eigenvalue is underestimated, typically when only a few neutrons are created from a reasonably large initial population and when the subsequent super-generations provide zero contributions to the estimate which is excluded. For the experiment of 4 super-generations, the segmentation fault occurred when 91861 neutrons were created in the last super-generation. Similarly for

<table>
<thead>
<tr>
<th>$p_r$</th>
<th>$RE$ [%]</th>
<th>$h$</th>
<th>CPU hours</th>
<th>$BU$ [%]</th>
<th>$&lt;m&gt;$</th>
<th>$maxm$</th>
</tr>
</thead>
<tbody>
<tr>
<td>1.0</td>
<td>87.88</td>
<td>$5.85 \times 10^7$</td>
<td>442.6</td>
<td>99.99</td>
<td>5.846</td>
<td>1290</td>
</tr>
<tr>
<td>0.98</td>
<td>8.072</td>
<td>$1.85 \times 10^9$</td>
<td>9294</td>
<td>26.75</td>
<td>185.24</td>
<td>32574</td>
</tr>
</tbody>
</table>

Table 7.1: Results for 50 super-generations for two $k_{eff}$ reduction parameters.
5 super-generations, with 324146 neutrons in the last generation. Despite the difficulties of obtaining complete simulations, the super-histories showed some improvement to the source convergence. The cumulative distribution of the active cycles is depicted in Fig. 7.7:

![Figure 7.7: Neutron source distribution for a batch size of 20 neutrons for different numbers of super-generations.](image)

Similarly, as the experiments for the batch size of 100 neutrons, the distribution is slightly skewed to the left-hand side of the slab for 3 super-generations. Again, this may be due to the large dominance ratio. The deviation of these source distributions compared with the reference calculation is given in Fig. 7.8:
As depicted in Fig. 7.8, the super-history improves the converged source by a few percents. Unlike for the simulations of 100 batch neutrons, the number of total neutron histories increases more rapidly for an increasing number of super-generations as depicted in Fig. 7.9:
Figure 7.9: Computational resources and source backup utilization for a batch size of 20 neutrons for different numbers of super-generations.

Furthermore, the neutron source backup utilization is also increasing more rapidly compared to the simulations with the batch size of 100 neutrons. Furthermore, the maximum population size in the simulation increased rapidly as depicted in Fig. 7.10:
Figure 7.10: Neutron source population size statistics for a batch size of 20 neutrons for different numbers of super-generations, where \( <m> \) is the average source neutron population for all super-generations, and \( \sigma_m \) is the population standard deviation.

The standard deviation of the population sizes reaches half of the mean value at only 3 super-generations. This is due to the few expected fission events within a generation.

The avalanche in Fig. 7.10 was as a contrast to the results for the 100 batch size experiments more skewed to very large population sizes, reaching \( \max m \sim 10^4 \) for only 3 super-generations compared to 10 super-generations as depicted in Fig. 7.6. The stability of the simulations may have been improved by applying some control for the neutron population avalanche. If the parameter \( p_r \) is selected smaller than unity, then the population numbers should skew away from population sizes near zero neutrons. This approach is somewhat similar for the case of the test of 50 super-generations for the 100 batch size. They, however, differ as the low batch size is prone to generate segmentation faults, and adjusting to slightly larger populations may improve the eigenvalue estimate by avoiding underevaluations.

The testing of avalanche control was not conducted, as the shortcomings are likely to be similar to those explored in Sect. 7.1. The re-
Results may improve significantly for a larger number of super-generations, however, it’s expected that the usage of computational resources will increase.
Chapter 8

Discussion

From the results, the super-history method was shown to effectively reduce the sampling bias from the normalization scheme. Some of the features of this method are however unexplored as only two batch sizes were simulated. As deduced from the results in Chapter 7, the stability of increasing super-generations is dependent on the number of batch neutrons. Attempting to skew the neutron population avalanche does not produce improved results although the stability is improved.

The problem used in this work is a simple single dimension problem with a homogeneous material throughout the spatial domain. The super-histories have proved to improve convergence for this high dominance ratio problem. For other applications, this method may improve source convergence for reactor problems with high dominance ratios arising from the weak neutronic coupling. For instance, a reactor with fuel assemblies separated by control rods will in some degree separate the neutron transport between fuel, thus increasing the dominance ratio due to the very restricted mobility of the neutrons.

Future implementations may attempt different eigenvalue averaging schemes to achieve estimations that provide better stability. For instance, neglecting the contributions created by super-generations after a die-out may significantly reduce the overestimations. A linear combination of previous eigenvalue estimates may also provide interesting results for the stability of the super-history method. Including previous results may delay the control of the neutron population avalanche, but can reduce the influence of occasional over and underestimated calculations.
As discussed in Sect. 2.5, the Serpent code are using isotropic fission neutron distribution, however, this is also assumed in the Boltzmann transport theory in Eq. (2.10). The theoretical framework for Monte Carlo simulations is not reliant on this assumption, as the fission operations are conducted in separative kernels, such that future implementations are related to the directions of fission fragments and neutrons are sampled from their respective angular and energy probability density functions. This is however based on a few experimental assessments, and the studies are only conducted for a few nuclides. The effects of non-isotropic fission neutron sources are however unlikely to provide a significant difference for simulations except for relatively few neutron histories.

The implementation of neutron source reuse in Sect. 6.2 is inferior compared to the direct removal of redundant neutrons. Unfortunately, no satisfactory method was found to perform this, but the resulting improvements to the figure of merits provide a confidence range, of which the actual performance reside. For future implementations, this issue may be addressed to avoid unnecessary computations.

The shortcomings of the super-history method became apparent for a certain number of super-generations for the fixed batch size. The results suggest that the increasing batch size allows an increase of super-generations without deteriorating the neutron population avalanche and computation stability. This finding is however only supported by experiments of two batch sizes. Furthermore, alternative eigenvalue estimation techniques may be useful to potentially find higher thresholds for the super-generation such that the increased accuracy of the eigenvalues leads to the improved control of the neutron population avalanche.

For future developments of derived super-history methods, growing the number of batch neutrons during inactive cycles in conjunction of increasing the number of super-generations may be a feasible candidate to improve figure of merits and acceleration of the neutron source. This method would require assessments on acceptable backup utilization factors, or utilize a conservative number of super-generations. An alternative is to alternate the increase of super-generations and neutron batch sizes until some threshold for the backup utilization is reached. Complementing the super-history with stratified sampling etc. can also provide improved convergence, however, these approaches require additional computing time.
Chapter 9

Conclusion

An implementation of the super-history powering method was proposed for the Serpent 2.1.30 code, including reuse of previous neutron source distributions whenever the source neutrons failed to provide fission neutrons and for controlling the sampling bias in the normalization scheme. The effective eigenvalue is kept fixed during the super-generations and is updated when the normalization scheme takes place, using an average of the eigenvalue estimates produced by each super-generations, excluding zero valued contributions.

The aim of this work was to demonstrate the capabilities of the super-history powering method regarding neutron source convergence. The super-history method proved to reduce the normalization sampling bias about linearly for the single dimensional slab problem for up to 40 super-generations for the batch size of 100 neutrons. The error of the converged source neutron distribution was reduced from 13.5% to 7.7% in the applied range of super-generations. The consequence of population avalanche became apparent when simulating 10 or more super-generations as the largest recorded population size was about 100 times larger than that of the batch size, and that the backup utilization factor increased from 36.7% to 65.5%. However, the computational time never increased, and the total number of simulated neutron histories did not exceed the intended number by more than 8%.

For the smaller batch size of 20, the result is not conclusive, due to the neutron population instability. However, from the results, the reference deviation of the source distribution suggest slight improvements for the small numbers of super-generations that were simulated.
The source distribution for 3 super-generations is strongly skewed to the left boundary, that is likely caused by the dominance ratio. The source distributions produced by the simulations of up to 3 super-generations were heavily biased compared to the 100 neutron batch size simulations.

Controlling the neutron population avalanche in the case of over and underestimations of the eigenvalue were not found satisfactory for the performance but provided improved stability of the super-histories. Furthermore, some improvements for stability is proposed to maintain the stability, while preserving performance. The proposed suggestion is to apply previous cycle estimates in a linear combination or to leave out estimates created in super-generations, where die-out of neutrons has occurred in the previous generation.
Bibliography


