ANGULAR IMPORTANCE SAMPLING FOR FORWARD AND ADJOINT MONTE CARLO RADIATION TRANSPORT

by

Philip S. Britt

A dissertation submitted in partial fulfillment of

the requirements for the degree of

Doctor of Philosophy

(Nuclear Engineering and Engineering Physics)

at the

UNIVERSITY OF WISCONSIN–MADISON

2021

Date of final oral examination: 2021-11-18

The dissertation is approved by the following members of the Final Oral Committee:
Paul P.H. Wilson, Professor, Engineering Physics
Douglass Henderson, Professor, Engineering Physics
Carl Sovinec, Professor, Engineering Physics
Bryan Bednarz, Associate Professor, Medical Physics
Qin Li, Associate Professor, Mathematics

Acknowledgments

I would like to personally thank all of my friends and family for the support and insight given when needed throughout this journey. Thank you to my advisor at the start, Douglass Henderson, who helped me navigate the beginning of this process and my advisor at the end, Paul Wilson, who helped me put the final pieces together. Thank you to my wife, Carly, who shared this journey with me.

This work was performed under appointment to the Nuclear Regulatory Commission Fellowship program at the University of Wisconsin - Madison Department of Engineering Physics and under grant funding from the Domestic Nuclear Detection office.

Table of Contents

List of Tables

List of Figures

Abstract

Variance reduction is an important tool to increase the rate of convergence in certain configurations of Monte Carlo problems. Methods such as CADIS are particularly useful to achieve this increased rate of convergence. However, CADIS does not include information for direction phase space, and an equivalent method has not been used for the adjoint Monte Carlo method. In this work, the benefits of including direction information in a weight window and weight target (a new type of importance sampling technique presented here) are analyzed and explored, along with a way to use importance sampling theory on the adjoint Monte Carlo method.

Chapter 1

Introduction

Monte Carlo methods are a popular choice for radiation transport problems - mostly due to their ability to model complex geometries relatively easily and implement continuous energy physics. However, utilizing Monte Carlo methods for some problems can lead to computationally expensive simulations that can take an inordinate amount of time to complete. A primary example of the kind of conditions that would lead to such a problem would be one where a large amount of attenuation occurs between the particle source and region of interest (such as a detector), where the set of particles that are able to produce results in that region of interest are a relatively small subset of all possible particle tracks for said problem. For this purpose, we seek methods to increase the rate at which these rare samples occur while also making sure that these methods do not alter the results in such a way to bias the statistical results. These methods fall within the scope of the term "variance reduction".

1.1 Variance and Efficiency

The Monte Carlo process of modeling radiation transport is one where the natural random processes of particles are modeled in order to generate statistical information from a subset of all possible particle tracks. This is done by representing these natural random processes (such as collision mechanics and optical distance) as PDFs (probability distribution functions) and sampling them appropriately. Once

this sampling procedure is constructed, the particle histories, where a single history includes a particle produced at its respective source and all of its progeny, transport through the medium and generate scores for estimators. The scores are numerical quantities generated via various events that the particles generate - whether they be collisions, transporting through space, energy deposition, and others. The estimators are simply mathematical rules that take these scores and convert them into relevant statistical data. A simple example of this is contained in equation 1.1 which presents the estimator one may use to calculate the mean $\hat{x}$ of some set of scores x_i. After N particle histories are completed, the distribution of scores generated via the Monte Carlo process is then used to determine the mean, variance, and other statistical quantities of interest that directly represent the solution to the transport equation and its precision and efficiency.

The term "variance reduction" is used because variance is a key statistical quantity used in determining the precision of a solution given by the Monte Carlo method. A mean is generated from the events that occur due to this subset of particle tracks, and that mean has an variance associated with it. For example, say we calculate the mean as follows:[1]

$$\hat{x} = \frac{1}{N} \sum_{i=1}^{N} x_i, \tag{1.1}$$

where $\hat{x}$ is the mean value of the set of scores x_i with a total of N histories. In a radiation transport sense, the scores x_i are the values produced by the various types of events mentioned earlier. The standard deviation S of the population (equivalent to the square root of the variance) is thus estimated as follows:[1]

$$S_x^2 \approx \hat{x^2} - \hat{x}^2, \tag{1.2}$$

where $\hat{x^2}$ is defined as follows:

$$\hat{x^2} = \frac{1}{N}\sum_{i=1}^{N} x_i^2, \tag{1.3}$$

and the population standard deviation S_x is related to the standard deviation of the mean $S_{\hat{x}}$ as follows:[1]

$$S_{\hat{x}}^2 = \frac{S_x^2}{N}. \tag{1.4}$$

We immediately note from equation 1.4 that $S_{\hat{x}} \propto \frac{1}{\sqrt{N}}$, which will become important later. The standard deviation of the mean is then used to define the estimated relative error R of a given Monte Carlo answer as follows:[1]

$$R \equiv \frac{S_{\hat{x}}}{\hat{x}}. \tag{1.5}$$

The relative error is used to represent statistical precision of the mean $\hat{x}$. In essence, the lower R is, the more precise $\hat{x}$ is which means it is a better estimate of the true answer that is sought from the problem.

For a given number of histories generated, the relative error R is typically dependent on the configuration of the problem itself. For instance, if there is a large amount of attenuation present in the problem, it will likely take additional histories to generate a relative error value low enough such that confidence can be had in the answer. In order to measure this relationship between the number of histories used to generate a relative error, the figure of merit is introduced as follows:[1]

$$FOM \equiv \frac{1}{R^2 t}, \tag{1.6}$$

where t is the time it takes to simulate a problem. The higher the FOM is, the more efficient the simulation is. Based on earlier equations describing the variance $S_{\hat{x}}$ and

its relationship to the relative error R, one can express R as follows:

$$R = \frac{C_R}{\sqrt{N}}, \tag{1.7}$$

and one can assume the following is true for t:

$$t = C_t N, \tag{1.8}$$

since t should be linearly proportional to the number of histories performed. These equations simply bind the various properties of the problem (variance reduction included) together into coefficients C_R and C_t to express these variables' (R and t) functional dependence on N. Thus, we can re-express the FOM as follows:

$$FOM = \frac{1}{C_R^2 C_t}. \tag{1.9}$$

If all other properties of the problem aside from variance reduction remain the same, the FOM becomes a measure solely of the effects of variance reduction on the relative error and the time it takes to complete the problem. Thus, this quantity is essentially a measure of the efficiency of the simulation, i.e. a measure of the length of time it takes to get an answer for a given relative error while taking into account how each variable is proportional to the number of histories generated. In general, it is typically found that variance reduction methods end up increasing C_t while decreasing C_R. The hope is that C_R is decreased sufficiently in comparison to C_t in order to increase the FOM and, therefore, the overall efficiency of the simulation itself.

1.2 Motivation

As mentioned earlier, one can generate a problem where only a small number of particle tracks will ever generate samples for the mean $\hat{x}$. The score values x_i are dependent on the ability of particles to generate events in or near regions of interest. Variance reduction techniques are therefore used to encourage particles to travel towards these regions of interest with the hope that this results in additional scores and a combination of C_R and C_t that produces an increased figure of merit.

There are a variety of methods to accomplish this, but this work is primarily concerned with importance sampling techniques and their approximations. Described in detail in section 2.6, these techniques seek to modify the particle generation and transport PDFs such that each particle track is more likely to generate samples in a region of interest. The particle generation and transport PDFS include several different dimensions for the steady state case: position $\vec{r}$, energy E, and direction $\hat{\Omega}$ ($\hat{\cdot}$ used to denote a unit vector in this case).

However, most variance reduction techniques in application today modify PDFs for position and energy, but often either use approximations for direction space or do not treat direction space at all, as noted in section 2.2. It seems intuitive that a reasonable way to increase efficiency of a Monte Carlo simulation would be to encourage the particles to point in a direction that increases its likelihood of generating samples.

It is also of note that there are two separate methods of simulating the Monte Carlo method for particle transport: forward and adjoint. The forward is one where a history begins at the radioactive source and transports through the problem phase space using the properties of the given type of particle being simulated (photon, neutron, etc.). The events (collisions, transporting through phase space,

deposition of energy/charge, etc.) are representations of the particle's physical processes and are used to generate scores for the estimators. However, in the adjoint simulation, the history begins at the detector and uses "reversed" physics to transport through the geometry. An example of these reversed physics would be that adjoint particles typically increase in energy after a collision rather than decrease. The events generated by adjoint particles represent the sensitivity of the detector to a hypothetical forward particle existing at that point in phase space where the event occurs.

Zero variance theory uses this relationship between the forward and adjoint Monte Carlo simulations in order to produce an importance sampling implementation that would theoretically only require a single history in order to achieve the result of interest for a problem, explained in detail in section 2.7. It is therefore often the case that various methods used today seek to produce approximations to the adjoint solution for use in importance sampling in order to increase the FOM for forward calculations since this should reduce the number of histories required to be performed in order to acquire a reasonable relative error. However, since the forward solution is adjoint to the adjoint simulation, it would also seem that one could use an approximation to the forward solution to similarly generate beneficial importance sampling parameters to increase the FOM for an adjoint simulation. This has not been attempted before.

This work therefore seeks to explore the benefits of both - applying importance sampling over direction phase space and for the adjoint Monte Carlo radiation transport simulation. The goal of this work is to demonstrate that including direction space in importance sampling techniques improves the rate of convergence, measured by the FOM, for a Monte Carlo radiation transport problem, and also to provide a method that performs variance reduction using the same importance

sampling techniques for continuous energy adjoint Monte Carlo radiation transport. In chapter 3, the efficacy of importance sampling utilizing direction information will be shown. In that chapter, a direction discretization utilizing a discrete ordinates angular quadrature will be provided and utilized to set up a direction discretization for estimators and variance reduction meshes. This information will then be used to show how including this information benefits the rate of convergence for a Monte Carlo simulation where the direction that a particle travels should intuitively greatly affect the rate of convergence. In chapter 4, a CADIS-equivalent method will be explored for the adjoint which uses the forward flux in the importance function for the adjoint Monte Carlo simulation. The efficacy of this will be determined and shown through improvements in the FOM measurement.

Chapter 2

Literature Review

In this chapter, a background for the Monte Carlo method in particle transport is presented with a more detailed exploration of importance sampling, zero variance solutions, and the state of the art importance sampling implementations currently used today. We begin with a basic review of Monte Carlo and variance reduction.

2.1 Monte Carlo

The Monte Carlo method is, at its most basic definition, a method in which random samples are taken from some existing population and statistical information is calculated from these samples. Take a simple example of a single 6-sided die. When the die is rolled, we can say that each side has an equal probability of occurring with a value $p = \frac{1}{6}$. Directly calculating the mean value of the enumerated sides that would be found if the die were to be rolled repeatedly is relatively simple - it's just $F_{die} = \sum_{l=1}^{6} p_l l = \sum_{l=1}^{6} \left(\frac{1}{6}\right) l = 3.5$.

If we were to use the Monte Carlo method to calculate this mean value though, we would simply roll the die repeatedly, keep a record of the summation of the resulting sides, and divide it by the number of rolls performed. This rule of calculating the mean value through statistical samples is a simple version of an estimator.[2] It can be mathematically expressed as follows:

$$\hat{F}_{die} = \frac{1}{N} \sum_{i=1}^{N} l_i, \tag{2.1}$$

Where $\hat{F}_{die}$ is an estimate of F_{die}, N is the number of rolls, and l_i is the numerical representation of the side that results from a roll (a value between 1 and 6). We also expect this estimator to be unbiased, which has a formal definition as follows:

$$b_\Lambda(P) = \langle \Lambda(X) \rangle - \theta = 0, \tag{2.2}$$

where b_Λ is the bias of some estimator Λ with a space of samples X from population P and real-valued statistical parameter θ of that population, and $\langle \cdot \rangle$ is the expectation value operator. The estimator Λ is measuring an estimate of the true statistical parameter θ. In the example above, $P = [1, 2, 3, 4, 5, 6]$, X would be the samples taken from P through repeated rolls and used in the estimator described in equation 2.1, and $\theta = 3.5$, the true mean of the population. If the estimator is unbiased, then $b_\Lambda(P) = 0$, and as the estimator in equation 2.1 approaches infinite samples ($N \to \infty$), $\hat{F}_{die} = F_{die}$.

2.2 Variance Reduction

Variance reduction is a diverse field with many different techniques, all primarily concerned with reducing the variance of an estimator and producing a more accurate answer given a number of samples. This particular work will be more concerned with importance sampling and its related techniques. Using importance sampling in the previous example would mean altering the probabilities of occurrence of each side of the die. To continue with the previous example of using a 6-sided die from section 2.1, we first must express an estimator equivalent to that in equation 2.1 using weighted samples l_i as follows:

$$\hat{F}_{die} = \frac{1}{\sum_{i=1}^{N} w_{l,i}} \sum_{i=1}^{N} w_{l,i} l_i, \tag{2.3}$$

where $w_{l,i}$ is the weight of sample i. In equation 2.1, these weights are just equal to 1. These weights are necessary to avoid biasing if we are to implement importance sampling. The altered probabilities are noted as $\overline{p}_l$, where l is again the numeric value representing a side. In order to avoid bias, the weights are defined as follows:

$$w_l = \frac{p_l}{\overline{p}_l}. \tag{2.4}$$

It's not hard to prove that this is not biased. We can easily calculate the average directly once again using these weights and their respective probabilities:

$$F_{die} = \sum_{l=1}^{6} \overline{p}_l w_l l = \sum_{l=1}^{6} \overline{p}_l \frac{p_l}{\overline{p}_l} l = \sum_{l=1}^{6} p_l l = 3.5. \tag{2.5}$$

For an example of a possible practical way to use this method in this context, one could set the $\overline{p}_3$ and $\overline{p}_4$ extremely high relative to $\overline{p}_1$, $\overline{p}_2$, $\overline{p}_5$, and $\overline{p}_6$. Presumably, this would cause the result of the estimator expressed in equation 2.3 to approach the mean of 3.5 much faster. While it's trivial in this example, particle transport problems typically involve complicated physics and geometry that need to be accounted for, and thus this technique can be used in Monte Carlo for particle transport to increase the rate of statistical convergence for a problem. This will be further explored in sections 2.6 and 2.7.

For variance reduction methods commonly used in particle transport other than importance sampling, the MCNP5 manual includes various examples such as energy cutoff, time cutoff, splitting and stochastic termination (rouletting), weight cutoff, weight windows, exponential transform, DXTRAN, and so forth.[1]. These methods

all alter the sampling procedure in various ways and simultaneously alter the weight of the individual samples to avoid biasing any estimators, just as in the previous example.

2.3 Forward Monte Carlo Particle Transport

Analog Forward Transport

The particle transport equation can be expressed as Fredholm integral equations of the second type. A full examination of how these equations are derived from the integro-differential particle transport Boltzmann equation and solved can be found in the thesis of Alex Robinson.[3] However, for the purposes of this work, this is not necessary since the integral equations are the ones used to define the Monte Carlo simulation. We begin with two equations that are relevant - the steady state equation for emission and collision density in a particle transport problem:

$$\chi(\vec{r}, E, \hat{\Omega}) = S(\vec{r}, E, \hat{\Omega}) + \int\int\int C(\vec{r}, E' \to E, \hat{\Omega}' \to \hat{\Omega}) T(\vec{r}' \to \vec{r}, \hat{\Omega}', E') \chi(\vec{r}', \hat{\Omega}', E') d\vec{r}' dE' d\Omega', \tag{2.6}$$

$$\psi(\vec{r}, E, \hat{\Omega}) = Q(\vec{r}, E, \hat{\Omega}) + \int\int\int T(\vec{r}' \to \vec{r}, E, \hat{\Omega}) C(\vec{r}', E' \to E, \hat{\Omega}' \to \hat{\Omega}) \psi(\vec{r}', \hat{\Omega}', E') d\vec{r}' dE' d\Omega', \tag{2.7}$$

where χ is the emission density, S is the source density, ψ is the collision density, and Q is the source's first collided collision density over phase space which can also be expressed as $\int T(\vec{r}' \to \vec{r}, E, \hat{\Omega}) S(\vec{r}', E, \hat{\Omega}) d\vec{r}'$. The phase space is described in $\vec{r}, E, \hat{\Omega}$ which are the position, energy, and direction respectively. T and C are the transport and collision kernels respectively.

Emission and collision density have relatively intuitive definitions. The emission density χ is the density of particles being emitted from some point in phase space, whether from an external source or exiting a collision at a phase space point. The collision density ψ is the density of particles entering a collision at some phase space point.

The transport kernel is the kernel that represents the transport of the particle from one spatial position to another*. The collision kernel is what represents the collision mechanics that a given particle experiences, transitioning it into new energy and direction coordinates (and possibly removes or generates additional particles, depending on the reaction type). The emission and collision density functions are also related via the following relations [4][5]:

$$\psi(\vec{r}, E, \hat{\Omega}) = \int T(\vec{r}' \rightarrow \vec{r}, E, \hat{\Omega}) \chi(\vec{r}', E, \hat{\Omega}) d\vec{r}', \tag{2.8}$$

$$\chi(\vec{r}, E, \hat{\Omega}) = \int C(\vec{r}, \hat{\Omega}' \rightarrow E, \hat{\Omega}' \rightarrow E) \psi(\vec{r}, \hat{\Omega}', E') dE' d\Omega'. \tag{2.9}$$

In a Monte Carlo particle transport simulation, the source function S is the first PDF that is sampled. For brevity, in the rest of this document where feasible, the phase space vector variable $\boldsymbol{Y}$ will be used in place of the individual phase space coordinates. Thus the PDF for the source is expressed as follows:

$$p_S(\boldsymbol{Y}) = \frac{S(\boldsymbol{Y})}{\int S(\boldsymbol{Y}) d\boldsymbol{Y}}. \tag{2.10}$$

From this PDF, the relevant CDF is formed:

*If this problem were not steady state, time would also be included as a variable of transition in this kernel

$$P_S(\mathbf{Y}) = \int_{\mathbf{Y}_0}^{\mathbf{Y}} p_S(\mathbf{Y}). \tag{2.11}$$

The transport kernel is described next. In the steady state, this kernel is used to describe how a particle is transported from one spatial coordinate to another. It can be expressed as follows:[5]

$$T(\vec{r}' \to \vec{r}, E, \hat{\Omega}) f(\vec{r}', E, \hat{\Omega}) = \Sigma_t(\vec{r}, E) e^{-\beta(\vec{r}, L, E, \hat{\Omega})} f(\vec{r}', E, \hat{\Omega}), \tag{2.12}$$

where $\beta(\vec{r}, L, \Omega, E) = \int_0^L \Sigma_t(\vec{r} - L' \cdot \hat{\Omega}, E) dL'$. The variable L is related to $\vec{r}$ as follows:

$$L = \|\vec{r} - \vec{r}'\|_2, \tag{2.13}$$

where $\|\cdot\|_2$ is the 2-norm (or vector magnitude) defined in equation 3.6.

We use T to define a PDF for a single particle that assumes that the direction and energy coordinates of said particle have already been determined (either through the sampling of the source or the collision kernel). In order to do this, we first note that $\Sigma_t(\vec{r}, E)$ is typically assumed to be constant inside a given material, meaning that the material's temperature and density are assumed constant throughout. Using this assumption means β can be simplified as follows:

$$\beta(L, E) = \sum_m \Sigma_t(E)_m L_m, \tag{2.14}$$

where we can immediately note that β is actually just the number of mean free paths a particle will travel through a series of materials. The dependence of Σ_t on position is removed since it is effectively a constant of the material m due to the above assumptions and since the energy E has already been selected and is known. This therefore allows us to generate a PDF based on the transport kernel using the

mean free path β as a variable as follows:

$$p_T(\beta) = e^{-\beta}, \tag{2.15}$$

where β is defined over the domain $[0, \infty]$. The Σ_t in front of the exponential from equation 2.12 disappears due to normalization of the PDF. Again, one needs to integrate this as follows in order to get the CDF to sample from to get a new position:

$$P_T(\beta) = \int_0^{\beta} p_T(\beta')d\beta' = 1 - e^{-\beta}.$$

Once β is appropriately sampled, the individual L_m values can be calculated noting the definition of L in equation 2.13 and doing some appropriate vector math to determine material boundary intersection points.

It should be noted that Σ, while commonly used for neutrons, could just as well be used for photons in this abstracted exploration of the transport equations. Lastly, the collision kernel C is described. For the integral equations 2.8 and 2.9, the kernel takes the following form:[5]

$$C(\vec{r}, E' \to E, \hat{\Omega}' \to \hat{\Omega}) f(\vec{r}, E', \hat{\Omega}') = \frac{\Sigma_s(\vec{r}, E', \hat{\Omega}') g(E' \to E, \hat{\Omega}' \to \hat{\Omega}) + \Sigma_a(\vec{r}, E')}{\Sigma_t(\vec{r}, E')} f(\vec{r}, E', \hat{\Omega}'). \tag{2.16}$$

An analog simulation is considered first for an more in-depth explanation of the collision kernel. This means that the physical characteristics of the simulation are unaltered and both absorption and scattering are possible. First, a simple set of 2 probabilities are sampled to determine whether the particle is scattered or absorbed:

$$p_C = \begin{cases} \frac{\Sigma_a}{\Sigma_t} \\ \\ \frac{\Sigma_s}{\Sigma_t} \end{cases}, \tag{2.17}$$

If the particle is absorbed, then that history is halted and another begins. Otherwise, if it is scattered, a PDF that describes the probability of all the different scattering reactions is sampled to determine what reaction it undergoes:

$$p_s = \begin{cases} \frac{\Sigma_{s_1}}{\Sigma_s} \\ \\ \frac{\Sigma_{s_2}}{\Sigma_s} \\ \\ \dots \\ \\ \frac{\Sigma_{s_N}}{\Sigma_s} \end{cases}. \tag{2.18}$$

After the reaction type is determined, the specific outgoing energy and direction (or energies and directions in the case of a reaction that produces multiple particles) are selected through a PDF that is formed from the physics of that particular type of reaction.

If a particle is absorbed outside of the detector region, it is not able to contribute to a detector response afterwards. This is especially consequential in deep-shielding problems, where the medium is one that does not easily permit particles to pass through it. Due to this, one might consider looking for a way to omit the absorption process from the simulation. This is the premise behind implicit capture, described in section 2.6.

2.4 Estimation of Forward Results

The description of how a particle transports through phase space has been made in the previous section. However, a method of gaining results from this simulation is also required. The mathematical rules that perform this task are called "estimators" and are of the same concept as that presented in the introduction *.

The objective of an estimator for the forward Monte Carlo simulation is to compute an inner product of a response function and a given particle transport function. If that particle transport function were the flux, then this can be expressed as follows:

$$F = \int \eta_\phi(\mathbf{Y})\phi(\mathbf{Y})d\mathbf{Y}, \tag{2.19}$$

where $\phi(\mathbf{Y})$ is the particle flux and $\eta_\phi(\mathbf{Y})$ is some function that describes a response to the flux $\phi(\mathbf{Y})$. If one simply wants the flux itself, then $\eta_\phi(\mathbf{Y}) = 1$. In addition, the response function can be a total cross section to find collision density, some function that converts the flux to a dose rate, etc.

One need not rely on flux information to find results. The collision and emission densities can also be used to produce equivalent results. Take the following response to the collision density $\psi(\mathbf{Y})$:

$$F = \int \eta_\psi(\mathbf{Y})\psi(\mathbf{Y})d\mathbf{Y}. \tag{2.20}$$

For the responses in equations 2.19 and 2.20 to be equal, the following transformation must take place:

*In MCNP, they are called "tallies".

$$\eta_\psi(\mathbf{Y}) = \frac{\eta_\phi(\mathbf{Y})}{\Sigma_t(\mathbf{Y})},$$

which is due to the transformation from flux to collision density being $\psi(\mathbf{Y}) = \Sigma_t(\mathbf{Y})\phi(\mathbf{Y})$.

There are many different types of estimators for particle transport. In this work, two in particular will be mathematically described.

The Last Event Estimator

The first to be analyzed is the last event estimator. This estimator only takes estimates when the particle experiences absorption. It can be described as follows:[6]

$$F = \sum_{deaths} \frac{w_i \eta_\psi(\mathbf{Y})}{P_a(\mathbf{Y})}, \tag{2.21}$$

where w_i is the weight calculated for a given particle and $P_a(\mathbf{Y})$ is the probability that no particles survive after the collision takes place. $\eta_\psi(\mathbf{Y})$ is some arbitrary response function to collision density chosen by the user for the specific problem being solved.

This estimator is important because it is for this estimator that using the adjoint function as an importance function provides a zero variance solution,[7][6] which is often used in literature as a basis for a type of importance sampling technique.[8][9][10] This will be discussed more in depth in section 2.7.

The Collision Estimator

The second type of estimator is a collision estimator. It simply tallies when a collision occurs:

$$F = \sum_{\text{collisions}} w_i \eta_\psi(\mathbf{Y}), \tag{2.22}$$

where w_i is the weight and $\eta_\psi(\mathbf{Y})$ is, again, some arbitrary response function given by the user. If $\eta_\psi(\mathbf{Y}) = 1$, then the collision density is being estimated. With this in mind, it becomes important to take this into account when using this response function to transform the result into other quantities of interest. Noting this, if one were to desire flux from this estimator, then $\eta_\psi(\mathbf{Y}) = \frac{1}{\Sigma_t(\mathbf{Y})}$.

Another estimator that is important to mention due to its widespread use in the field of Monte Carlo particle transport is the track length flux estimator. It is likely the most common estimator used in all of the current prevalent Monte Carlo codes. However, the zero variance importance sampling scheme for this estimator is relatively complicated,[11] and so it is not explored here for the purposes of zero variance.

2.5 Adjoint Monte Carlo Particle Transport

Adjoint Monte Carlo is somewhat more difficult than the forward. It's very important to keep the terminology distinct. For example, the adjoint to the forward emission density is a very different quantity from the emission density of the actual adjoint pseudoparticles being simulated in a Monte Carlo simulation. Let us start with the simple equation for the adjoint *to the forward emission density*. The following starting equations, discussion, and derivation are taken from the work of Hoogenboom:[4]

$$\chi^\dagger(\mathbf{Y}) = \eta_\chi(\mathbf{Y}) + \int K^\dagger(\mathbf{Y}' \to \mathbf{Y}) \chi^\dagger(\mathbf{Y}') d\mathbf{Y}', \tag{2.23}$$

where $K^{\dagger}$ is a combination of the adjoint transport and collision kernel, and the following relation holds with respect to the forward kernel for emission density (which is also a combination of the forward transport and collision kernel):

$$K^{\dagger}(\mathbf{Y}' \rightarrow \mathbf{Y}) = K(\mathbf{Y} \rightarrow \mathbf{Y}'). \tag{2.24}$$

A great deal of confusion arises directly from equation 2.23. While this function is adjoint to the forward emission density, it is not an emission density itself. Physically, it is impossible for a particle to be emitted or collide in a vacuum for the model we are using. However, the adjoint to the forward emission density itself is a flux-like quantity that is actually non-zero in a vacuum.[3][4] This function is interpreted as the response of the region of interest where the estimator exists to a forward emission density at that region in phase space $\mathbf{Y}$. The reason that it is non-zero in a vacuum can therefore be intuitively deduced that even though it is physically impossible for a particle to be emitted from vacuum, if a particle were hypothetically to be emitted from that vacuum then the response it is able to produce at the detector is not zero. The adjoint to the forward collision density and flux are similar in terms of their flux-like nature. [3]

Also, $\eta_{\chi}(\mathbf{Y})$ is the response function to the forward emission density in phase space $\mathbf{Y}$. The reason that the equation for the adjoint to the forward emission density is often the quantity used for Monte Carlo purposes is that if one references equations 2.6 and 2.7, the actual source profile (which is more easily known than the density of first collisions used as the source for the collision density equation) is utilized in the equation for emission density. The following inner products hold by definition of the adjoint:

$$F = \int \eta_\chi(\mathbf{Y})\chi(\mathbf{Y})d\mathbf{Y} = \int S(\mathbf{Y})\chi^\dagger(\mathbf{Y})\mathbf{Y}. \tag{2.25}$$

This therefore leads us to the conclusion that the adjoint to the forward emission density, $\chi^\dagger$, will be easier to produce results of interest from as opposed to the adjoint to the forward collision density, $\psi^\dagger$. This is because the adjoint to the forward emission density uses the physical source expression S in its integral to determine F, where S is typically already known. This is opposed to the first collision density Q from equation 2.7, which is not typically known beforehand. The kernel for adjoint transport is typically transformed in order to utilize the Monte Carlo method as follows:[4]

$$\tilde{\chi}^\dagger(\mathbf{Y}) = \Sigma_t(\mathbf{Y})\chi^\dagger(\mathbf{Y}), \tag{2.26}$$

$$L^\dagger(\mathbf{Y}' \to \mathbf{Y}) = \frac{\Sigma_t(\mathbf{Y})}{\Sigma_t(\mathbf{Y}')}K^\dagger(\mathbf{Y}' \to \mathbf{Y}). \tag{2.27}$$

Thus equation 2.26 is transformed as follows:

$$\tilde{\chi}^\dagger(\mathbf{Y}) = \Sigma_t(\mathbf{Y})\eta_\chi(\mathbf{Y}) + \int L^\dagger(\mathbf{Y}' \to \mathbf{Y})\tilde{\chi}^\dagger(\mathbf{Y}')d\mathbf{Y}'. \tag{2.28}$$

As mentioned in 2.4, the appropriate response function must also be transformed to keep the response the same:[3]

$$F = \int \frac{S(\mathbf{Y})}{\Sigma_t(\mathbf{Y})}\tilde{\chi}^\dagger(\mathbf{Y})d\mathbf{Y}. \tag{2.29}$$

Now, in order to define a transport sampling process which is analogous to the forward Monte Carlo simulation, a transport and collision kernel must be identified.

The mathematics for a suitable adjoint transport kernel are taken from the work by Hoogenboom, and are as follows:[4]

$$T^{\dagger}(\vec{r}' \rightarrow \vec{r}, E, \hat{\Omega}) = T(\vec{r}' \rightarrow \vec{r}, E, -\hat{\Omega}). \tag{2.30}$$

However, note from equation 2.24 that what is actually desired is a transport kernel that contains $\vec{r} \rightarrow \vec{r}'$ in order to have this kernel in a familiar form that can then be substituted into the kernel transform for equation 2.27. Fortunately, we may note the negative in front of the direction that gives us a relatively easy derivation. The following is therefore true:[4]

$$\begin{aligned} \frac{T(\vec{r} \rightarrow \vec{r}', E, \hat{\Omega})}{\Sigma_t(\vec{r}', E)} &= e^{-\int_0^L \Sigma_t(\vec{r}' - L' \cdot \hat{\Omega}) dL'} \\ &= e^{-\int_0^L \Sigma_t(\vec{r} + L' \cdot -\hat{\Omega}) dL'} \\ &= \frac{T(\vec{r}' \rightarrow \vec{r}, E, -\hat{\Omega})}{\Sigma_t(\vec{r}, E)}. \end{aligned} \tag{2.31}$$

Using equation 2.30, the adjoint transport kernel is as follows:

$$T^{\dagger}(\vec{r}' \rightarrow \vec{r}, E, \hat{\Omega}) = \frac{\Sigma_t(\vec{r}, E)}{\Sigma_t(\vec{r}', E)} T(\vec{r} \rightarrow \vec{r}', E, \hat{\Omega}). \tag{2.32}$$

The collision kernel is more complex. It is normalized to the adjoint scattering cross section *:

$$\Sigma^{\dagger}(\vec{r}, E') = \iint \Sigma_t(\vec{r}, E) C(\vec{r}, E \rightarrow E', \hat{\Omega} \rightarrow \hat{\Omega}') dE d\hat{\Omega}, \tag{2.33}$$

*This quantity does not have a subscript of "s" to denote that it is solely a scattering cross section because there is no such thing as absorption in the adjoint process

where

$$C^{\dagger}(\vec{r}, E' \to E, \hat{\Omega}' \to \hat{\Omega}) = \frac{\Sigma_t(\vec{r}, E) C(\vec{r}, E \to E', \hat{\Omega} \to \hat{\Omega}')}{\Sigma^{\dagger}(\vec{r}, E')}. \tag{2.34}$$

These kernels are substituted back into equation 2.26, and the following equation is obtained:

$$L^{\dagger}(\boldsymbol{Y}' \to \boldsymbol{Y}) = P^{\dagger}(\vec{r}, E') C^{\dagger}(\vec{r}', E' \to E, \hat{\Omega} \to \hat{\Omega}') T^{\dagger}(\vec{r}' \to \vec{r}, E, \hat{\Omega}), \tag{2.35}$$

where $P^{\dagger}(\vec{r}, E) = \frac{\Sigma^{\dagger}(\vec{r}, E)}{\Sigma_t(\vec{r}, E)}$. This is the adjoint weight factor. It is analogous in how it is formed to the forward non-absorption probability used in implicit capture, described in section 2.6. However, it is not a probability since its value is bounded by $[0, \infty)$.

Next, the response functions can be analyzed. If we assume that there is some response F that we are looking for, we can express this result F in terms of multiple different response functions which correspond to a given function that describes particle population - whether it be the emission density, collision density, or flux itself:[4]

$$F = \int \eta_{\chi}(\boldsymbol{Y}) \chi(\boldsymbol{Y}) d\boldsymbol{Y} = \int \eta_{\psi}(\boldsymbol{Y}) \psi(\boldsymbol{Y}) d\boldsymbol{Y}, \tag{2.36}$$

where one may recall that ψ is the forward collision density. Referring to equation 2.8 and 2.9, it can be shown that these response functions may also be transformed between one another. Continuing from equation 2.36,

$$\begin{aligned} F &= \int\int\int \eta_{\psi}(\vec{r}, E, \hat{\Omega}) T(\vec{r}' \to \vec{r}, E, \hat{\Omega}) \chi(\vec{r}', E, \hat{\Omega}) d\vec{r} dE d\hat{\Omega} \\ &= \int\int\int \eta_{\psi}(\vec{r}', E, \hat{\Omega}) T(\vec{r} \to \vec{r}', E, \hat{\Omega}) \chi(\vec{r}, E, \hat{\Omega}) d\vec{r} dE d\hat{\Omega}. \end{aligned} \tag{2.37}$$

With this, the transformation between response functions can be deduced:

$$\begin{aligned}\eta_\chi(\vec{r}, E, \hat{\Omega}) &= T(\vec{r} \to \vec{r}', E, \hat{\Omega})\eta_\psi(\vec{r}', E, \hat{\Omega}) \\ &= \frac{T(\vec{r} \to \vec{r}', E, \hat{\Omega})}{\Sigma_t(\vec{r}', E)}\eta_\phi(\vec{r}', E, \hat{\Omega}).\end{aligned} \tag{2.38}$$

In this equation, $\phi(\mathbf{Y})$ is the angular forward flux. After some additional mathematical manipulations (which, again, can be found in the work by Hoogenboom [4]), the adjoint equation that is used as the framework for sampling in the adjoint Monte Carlo process is as follows:

$$\zeta^\dagger(\mathbf{Y}) = \eta_\phi(\mathbf{Y}) + \int M^\dagger(\mathbf{Y}' \to \mathbf{Y})\zeta^\dagger(\mathbf{Y}')d\mathbf{Y}', \tag{2.39}$$

where $\zeta^\dagger(\mathbf{Y})$ is the emission density of the adjoint pseudoparticles transported in the adjoint Monte Carlo process and the kernel $M^\dagger(\mathbf{Y}' \to \mathbf{Y})$ is as follows:

$$M^\dagger(\mathbf{Y}' \to \mathbf{Y}) = T^\dagger(\vec{r}' \to \vec{r}, E', \hat{\Omega}')P^\dagger(\vec{r}, E')C^\dagger(\vec{r}, E' \to E, \hat{\Omega}' \to \hat{\Omega}). \tag{2.40}$$

Here, we note that we have a framework of adjoint equations to more conveniently produce an adjoint Monte Carlo simulation from. There are multiple notable features that determine this. First, we note that equation 2.39 defines an emission density of the adjoint pseudoparticles that utilizes the response function to the forward flux as its source. This is beneficial because the response function to forward flux, η_ϕ is often readily known as opposed to the response function to the forward emission density, η_χ, which takes additional mathematical effort to produce. Another feature of note is that the collision density of these adjoint pseudoparticles, represented

in equation 2.28, is easily converted to the adjoint to the forward emission density by simply dividing $\tilde{\chi}$ by the total cross section. This gives us a way to utilize the source for the forward emission density as an adjoint response function, which is also more readily known than a quantity such as the density of first collisions that is the source term for the forward collision density. Figure 2.1 shows a more intuitive representation of the adjoint Monte Carlo transport process.

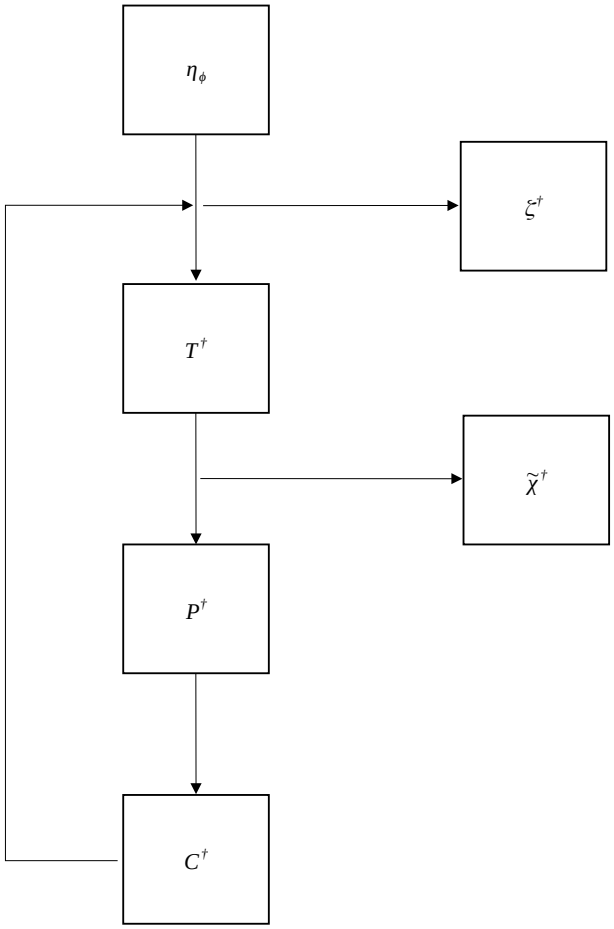


Figure 2.1: Adjoint Monte Carlo Flow Chart

The method of gaining adjoint results using estimators is the same as the forward. However, it is of note that when one uses a track length estimator in an adjoint problem, the adjoint to the forward emission density is being measured instead of the adjoint to the forward flux. By reviewing section 2.4 and examining the collision estimator, we can see why this is the case. After all, we accept that a track length flux estimator used for the forward is directly measuring the forward flux, while the collision estimator must have each score divided by Σ_t in order to also measure the forward flux. An analagous relationship exists between the collision density of the adjoint pseudoparticles and the adjoint to the forward emission density, noted once again in 2.28. Therefore, a collision estimator in the adjoint with a response function of $\frac{1}{\Sigma_t}$ is measuring the adjoint to the forward emission density. Since the track length estimator for the forward with a response function of 1 gives the forward flux, it stands to reason that the track length estimator in the adjoint with an equivalent response function would simply give the adjoint to the forward emission density. A more intricate derivation of the track length flux estimator can be found in the text by Spanier and Gelbard which more explicitly illustrates this concept.[12]

2.6 Importance Sampling

The fundamentals of applying importance sampling are relatively straightforward. However, difficulties can arise if one tries to apply this technique directly to the more mathematically complex portions of the Monte Carlo particle transport process. This section will largely be based on the work by Spanier and Gelbard [12]. It should be noted that this section is not particular to the forward or adjoint simulation, so any nomenclature associated with the forward can be assumed to apply to the adjoint as well.

Basic Importance Sampling Theory

To begin, recall response integrals mentioned in section 2.4 - particularly equation 2.20. In effect, estimation of these response integrals becomes equivalent to an expectation value of the estimators mentioned in section 2.4 in the infinite number of samples, which can be translated into an integral as follows [12]:

$$\langle \Lambda \rangle = \int_P \Lambda(X) d\mu(X). \tag{2.41}$$

As in equation 2.2, $\langle \cdot \rangle$ is the expectation value operator, Λ is an estimator, P is the space of all possible samples, X is a set of samples, and μ is the relevant probability density function that describes the probability of a sample.

This is a bit difficult to parse if one isn't intimately familiar with probability theory and its more abstract expression of the Monte Carlo process. Ultimately, all it means is that the expectation value of the response, which is the desired result we seek, is an integral over all possible events which have probabilities that can be described by a continuous function μ. This makes sense, as in order to gain the expectation value of an estimator as a result of said estimator, then one would have to sample every single possible state in a problem to achieve that result.

To express importance sampling mathematically, equation 2.41 is rewritten in the following form:

$$\langle \Lambda \rangle = \int_P \left[\Lambda \frac{d\mu}{d\overline{\mu}} \right] d\overline{\mu}, \tag{2.42}$$

$$\overline{\Lambda}(X) = \Lambda(X) \frac{d\mu}{d\overline{\mu}}(X), \tag{2.43}$$

$$\langle \overline{\Lambda} \rangle = \int_{P} \overline{\Lambda}(X) d\overline{\mu}(X). \tag{2.44}$$

In this case, $\overline{\mu}$ is an altered probability distribution function. Immediately, it becomes clear that $\overline{\Lambda}$ is also an unbiased estimator of the same statistical parameter measured by Λ. It's just a simple re-expression of the original expectation value, after all. For reference, bias is expressed mathematically in equation 2.2.

So, basically, if an estimator is unbiased then the expectation value of this estimator is equal to the true value of the statistical quantity (in the case of particle transport, F).

This encompasses the fundamental theory of importance sampling. It's a way of rewriting the original distribution that is being sampled in the hope that the new one will be better conditioned and return a result more efficiently by reducing the variance. Note that, in this case, one can heuristically interpret $\frac{d\mu}{d\overline{\mu}}$ as the re-weighting of samples that keep the estimator unbiased.

Some conditions must be applied to this transformation before utilization in order to guarantee the unbiasedness of the sampling is maintained. First, it must be noted that there is an immediately clear restriction that must be set in order to guarantee that the derivative $\frac{d\mu}{d\overline{\mu}}$ exists. If $\mu \neq 0$, then it must also be true that $\overline{\mu} \neq 0$.[12] There are some exceptions to this rule that exist, namely section 2.6 on implicit capture could be regarded as such a case since it is an implementation of importance sampling. The reason this is acceptable will be explained in that section.

While the previously mentioned restriction will be the most important for the purposes listed here, there are two additional restrictions that must be placed on this process to ensure the existence of the derivative $\frac{d\mu}{d\overline{\mu}}$. The first being that the following function is bounded (except for events with zero probability in the modified probability distribution function $\overline{\mu}$) [12]:

$$G(\mathbf{Y}_1, ..., \mathbf{Y}_k) = \prod_{j=2}^{k} \frac{K(\mathbf{Y}_j, \mathbf{Y}_{j-1})}{\overline{K}(\mathbf{Y}_j, \mathbf{Y}_{j-1})} \frac{q(\mathbf{Y}_{j-1})}{\overline{q}_{j-1}(\mathbf{Y}_1, ..., \mathbf{Y}_{j-1})} \frac{X(\mathbf{Y}_{j-1})}{c(\mathbf{Y}_{j-1})} \frac{S(\mathbf{Y}_1)}{\overline{S}(\mathbf{Y}_1)} \frac{p(\mathbf{Y}_k)}{\overline{p}_k(\mathbf{Y}_1, ..., \mathbf{Y}_k)}. \tag{2.45}$$

Here, $q(\mathbf{Y}_i)$ is the probability of a particle history continuing beyond state $\mathbf{Y}_i$, $p(\mathbf{Y}_i)$ is the probability of the history ending at state $\mathbf{Y}_i$ (and hence $q(\mathbf{Y}_i) = 1 - p(\mathbf{Y}_i)$, and $c(\mathbf{Y}_i)$ is the mean number of secondary particles produced at state $\mathbf{Y}_i$ from an initial primary particle.

The last condition is rather abstract. There must exist an integer M such that, for $n \geqslant M$, [12]

$$\prod_{i=1}^{n} \frac{\overline{q}_i(\mathbf{Y}_1, ..., \mathbf{Y}_i)}{X(\mathbf{Y}_i)} \leqslant 1 \quad \forall\, \mathbf{Y}_1, ..., \mathbf{Y}_n. \tag{2.46}$$

In a Monte Carlo radiation transport problem, there are three PDFs that can be altered using importance sampling - the physical source S, the PDF defined by the collision kernel $C(\vec{r}, E' \to E, \hat{\Omega}' \to \hat{\Omega})$, and the PDF defined by the transport kernel $T(\vec{r}' \to \vec{r}, E, \hat{\Omega})$. Of course, the first collision density Q in the equation for collision density 2.6 could also be counted among them, but this function is not typically given in a Monte Carlo problem and so it will not be counted among these functions. Due to the difficulty of applying importance sampling directly to the collision kernel over the entire problem, only the source distribution S will be directly altered using importance sampling in such a way since the PDF is already given. The transport kernel will not have it applied in this work, but doing this is known as "exponential transform".[1] Thus, the source will be transformed as follows:

$$\overline{S}(\mathbf{Y}) = \frac{S(\mathbf{Y})I(\mathbf{Y})}{\int S(\mathbf{Y})I(\mathbf{Y})d\mathbf{Y}}. \tag{2.47}$$

In order to keep this from biasing the estimators, in matching with the theory above, every particle weight will be revised as follows:

$$w = \frac{S(\mathbf{Y})}{\overline{S}(\mathbf{Y})}. \tag{2.48}$$

However, we must move to other methods in order to modify the actual transport process of the particle itself.

Implicit Capture

Implicit capture is an important technique when importance sampling is used. Recall from equation 2.16 that absorption is included in the collision kernel itself. While this does accurately reflect the reality of the physics, it can prevent particles from reaching a region of interest if these particles are absorbed before they can provide any scores.

What is often done then in lieu of stopping the history is to reduce the weight of the particle by multiplying it by the probability that the particle would survive that interaction. In other words, equation 2.17 becomes

$$p_{\hat{C}} = \begin{cases} 0 & absorb, \\ \frac{1}{\frac{\Sigma_s}{\Sigma_t}} \frac{\Sigma_s}{\Sigma_t} = 1 & scatter. \end{cases} \tag{2.49}$$

Thus, to remain unbiased, every particle weight is multiplied by $\frac{\Sigma_s}{\Sigma_t}$, which when equation 2.49 is multiplied by this factor, one can see that the original probability for scattering is regained. This agrees with the reweighting due to an altered PDF used with importance sampling presented in section 2.6.

The conditions under which this gives an unbiased answer is relatively simple.

Unless the estimator specifically requires the absorption process to take samples, such as the last event estimator, then the estimator will remain unbiased given the appropriate weight transformation is applied.[13] Implicit capture has the effect of always reducing variance in any problem, and more on the mathematical proof of this can be read in the work by Lux and Koblinger [13].

Unfortunately, one must also consider the run time issues that can be introduced by utilizing implicit capture. If a particle is never absorbed, and no other variance reduction methods are present, the only method of changing to a new history would be for the particle to escape the geometry. A problem where a particle would scatter to extremely low energies before escaping is not uncommon in the practical realm, and as the energy decreases so does the distance that the particle typically travels. Thus a particle can be "trapped" in the geometry, repeatedly scattering to lower and lower energies and lower and lower weights and not travel any significant distance. Therefore, this technique must be used in conjunction with a technique known as weight cutoff that prevents particles from going below a certain weight, which is described in the next subsection for splitting and stochastic termination.

The effect on the collision kernel is easy to see. The kernel must have absorption removed, and it must be renormalized accordingly. The following is the result [5]:

$$\begin{aligned} C(\vec{r}, E' \to E, \hat{\Omega}' \to \hat{\Omega}) f(\vec{r}, E', \hat{\Omega}') = \\ \left(\frac{\Sigma_s(\vec{r}, E', \hat{\Omega}')}{\Sigma_t(\vec{r}, E')}\right)\left(\frac{\Sigma_s(\vec{r}, E', \hat{\Omega}') g(E \to E, \hat{\Omega}' \to \hat{\Omega})}{\Sigma_s(\vec{r}, E', \hat{\Omega}')}\right) f(\vec{r}, E', \hat{\Omega}') = \\ \left(\frac{\Sigma_s(\vec{r}, E', \hat{\Omega}')}{\Sigma_t(\vec{r}, E')}\right) g(E \to E, \hat{\Omega}' \to \hat{\Omega}) f(\vec{r}, E', \hat{\Omega}'), \end{aligned} \quad (2.50)$$

where $\Sigma_s(\vec{r}, E')$ is defined as the total scattering cross section at position $\vec{r}$ and energy

E'. In other words, it is the differential scattering cross section integrated over all possible outgoing energies and directions. It is of note that the weight transform for implicit capture (that is, the non-absorption probability) is analogous to how the adjoint weight factor described in section 2.5 mathematically presents itself and is applied (at collisions). However, while the adjoint weight factor is not a probability, the weight transform for implicit capture is a well-defined probability of an interaction occuring that is bounded by $[0, 1)$.

Splitting and Stochastic Termination

This section will be concerned with a brief exploration of a well-known technique used in the Monte Carlo method. While a more mathematical proof of it not creating a biased estimator can be found again in the work by Spanier and Gelbard,[12] this thesis instead follows a more simple and straightforward approach to explaining this technique. Essentially, splitting and stochastic termination seek to achieve the same goal as importance sampling without the extra complication of applying importance sampling to more complicated sampling processes (i.e. the collision mechanics of neutrons). A mathematical description follows.

Let $I(\mathbf{Y})$ be an importance function over the entire phase space of a problem - the same kind of function used in equation 2.47. Say that a particle travels from phase space $\mathbf{Y}'$ to $\mathbf{Y}$. The following fraction is calculated:

$$\frac{I(\mathbf{Y})}{I(\mathbf{Y}')} = \gamma. \tag{2.51}$$

There are then two cases to consider - that being where $\gamma > 1$ and where $\gamma < 1$ (if $\gamma = 1$, nothing is done). The cases are as follows:

- If $\gamma > 1$, split particle:
  - $\mathrm{P}\left(n_{split} = \lceil \gamma \rceil\right) = \gamma - \lfloor \gamma \rfloor$
  - $\mathrm{P}\left(n_{split} = \lfloor \gamma \rfloor\right) = 1 - \mathrm{P}\left(n_{split} = \lceil \gamma \rceil\right)$
  - $w(\mathbf{Y}') = \frac{1}{\gamma} w(\mathbf{Y})$
- If $\gamma < 1$, stochastically terminate particle:
  - $\mathrm{P}_{terminate} = 1 - \gamma \quad \longrightarrow \quad w(\mathbf{Y}') = 0$
  - $\mathrm{P}_{survive} = \gamma \quad \longrightarrow \quad w(\mathbf{Y}') = \frac{1}{\gamma} w(\mathbf{Y})$

When splitting, the number of resulting particles n_{split} is probabilistically determined based on the fraction γ with probabilities $\mathrm{P}\left(n_{split} = \lceil \gamma \rceil\right)$ and $\mathrm{P}\left(n_{split} = \lfloor \gamma \rfloor\right)$. The weight w is transformed according to the fraction γ in order to avoid biasing the simulation. When undergoing a stochastic termination procedure, the particle either survives and results with an equivalent weight transform as in splitting, or the particle is terminated and thus the weight of the particle is 0.

Both of these techniques are individually unbiased and do not need to be utilized in conjunction with each other. [12] They also need not be based on an importance value that is a function of the phase space - weight windows use the weight itself to determine when to split and stochastically terminate.[1] Weight windows do, however, use a slightly different splitting procedure.

To give a brief overview of weight windows, the following theory is presented. The weight window shall be denoted as $W(\mathbf{Y})$. A weight window contains three values that are listed in descending order of magnitude - an upper weight, a survival weight, and a lower weight:

$$W(\mathbf{Y}) = \begin{cases} W_u \\ W_s \\ W_l \end{cases} \quad W_u > W_s > W_l. \tag{2.52}$$

The following conditions are then true for a particle of weight w entering $W(\mathbf{Y})$. If $w > W_u$, then the particle is split into $\lceil \frac{w}{W_u} \rceil$ particles, with each particle having a resulting weight of $\frac{1}{\lceil \frac{w}{W_u} \rceil}$. If $W_u \geqslant w \geqslant W_l$, nothing is done and the particle continues. If $W_l > w$, then the particle is stochastically terminated based on the probability $1 - \frac{w}{W_s}$, and so if it survives, it is re-weighted to the survival weight W_s.[1]

Weight cutoff, which is the procedure mentioned in section 2.6 that must be used with implicit capture, is exactly equivalent to using W_l and W_s of a weight window but having no W_u. It is usually a technique applied everywhere in the entire phase space, and simply used to prevent particles from going below a certain weight.

2.7 Zero Variance Using Importance Sampling

The entirety of the mathematics from this section come from the work by Hoogenboom.[14] For simplicity of this section, the integral equations for emission density and collision density expressed earlier will not be used from this point forward. Instead, we express the forward and adjoint integral equations in an abstract fashion as follows:

$$E(\mathbf{Y}) = S(\mathbf{Y}) + \int K(\mathbf{Y}', \mathbf{Y}) E(\mathbf{Y}') d\mathbf{Y}', \tag{2.53}$$

$$E^\dagger(\mathbf{Y}) = \eta_E(\mathbf{Y}) + \int K(\mathbf{Y}, \mathbf{Y}') E^\dagger(\mathbf{Y}') d\mathbf{Y}'. \tag{2.54}$$

These equations are kept generic for simplicity in this section such that E is a generic event density and $E^\dagger$ is its respective adjoint with $\mathbf{Y}$ being a vector variable that contains all relevant dimensions of phase space for a particle transport problem. With this being the case, S is interpreted as some initial event density and K as the integral kernel (which would be composed of the transport and collision kernel in particle transport). η_E is a response function to the forward event density E as demonstrated in equation 2.56. Integrating the kernel leads to the following:

$$\int K(\mathbf{Y}', \mathbf{Y}) d\mathbf{Y} = \sigma(\mathbf{Y}) = 1 - \alpha(\mathbf{Y}), \tag{2.55}$$

where σ is the probability of scattering and α is the probability of absorption. The equation that describes the solution of interest that we are seeking is as follows:

$$F = \int \eta_E(\mathbf{Y}) E(\mathbf{Y}) d\mathbf{Y} = \int S(\mathbf{Y}) E^\dagger(\mathbf{Y}) d\mathbf{Y}. \tag{2.56}$$

Assuming proper normalization of S, we introduce a generic importance function I, and assume that the following equations describe an importance sampled source and kernel:

$$\hat{S}(\mathbf{Y}) = S(\mathbf{Y}) I(\mathbf{Y}), \tag{2.57}$$

$$\hat{K}(\mathbf{Y}', \mathbf{Y}) = K(\mathbf{Y}', \mathbf{Y}) \frac{I(\mathbf{Y})}{I(\mathbf{Y}')}. \tag{2.58}$$

We assume these functions provide proper normalization, such that the following is true:

$$\int \hat{S}(\mathbf{Y}) d\mathbf{Y} = 1, \tag{2.59}$$

$$\int \hat{K}(\mathbf{Y}', \mathbf{Y}) d\mathbf{Y} = \hat{\sigma}(\mathbf{Y}). \tag{2.60}$$

In order to illustrate a point that will become important later, we will briefly cover the zero variance proof for last event estimators. We note the form of the score for a last event estimator as follows:

$$f_{le} = w \frac{\eta_E(\mathbf{Y})}{\alpha(\mathbf{Y})}. \tag{2.61}$$

If the simulation is perfectly analogue in the forward, then the weight will be always be 1. However, if importance sampling has been applied, then the weight will be as follows:

$$w(\mathbf{Y}_i) = \frac{S(\mathbf{Y}_0)}{\hat{S}(\mathbf{Y}_0)} \prod_{c=1}^{i} \frac{K(\mathbf{Y}_{c-1}, \mathbf{Y}_c)}{\hat{K}(\mathbf{Y}_{c-1}, \mathbf{Y}_c)}, \tag{2.62}$$

where c can be interpreted as an index representing the transition in phase space through the collision kernel and transport kernel together and i is the latest transition a particle has undergone. Substituting equations 2.57 and 2.58 into equation 2.62 leads to a convenient simplification:

$$w(\mathbf{Y}_i) = \frac{1}{I(\mathbf{Y}_0)} \prod_{c=1}^{i} \frac{I(\mathbf{Y}_{c-1})}{I(\mathbf{Y}_c)} = \frac{1}{I(\mathbf{Y}_i)}. \tag{2.63}$$

If importance sampling is applied, it is also clear that the absorption probability would change based on equation 2.60, and we label this $\hat{\alpha} = 1 - \hat{\sigma}$. If one utilizes $I(\mathbf{Y}) = \frac{E^{\dagger}(\mathbf{Y})}{F}$ as the importance function in equation 2.58, through some manipulations one may arrive at the following:

$$\hat{\alpha}(\mathbf{Y}) = \frac{\eta_E(\mathbf{Y})}{E^\dagger(\mathbf{Y})}. \tag{2.64}$$

This is important because if we take equations 2.64 and 2.63 with $\frac{E^\dagger(\mathbf{Y})}{F}$ substituted as the importance function in equation 2.63, and then substituting these expressions into the last event estimator score in equation 2.61, one arrives at the following:

$$f_{le} = F, \tag{2.65}$$

which is indicative of a zero variance solution.

2.8 State of the Art Importance Sampling Techniques

There is a variety of different methods that use the above zero variance proof to implement an importance sampling scheme using weight windows or importances to hopefully improve the rate of convergence of a given problem. Here, we cover a few to provide a basis for the rest of this research. First, we note once again the following definition that provides a mathematical description of the rate of convergence of a Monte Carlo estimator - the figure of merit:

$$FOM = \frac{1}{tR^2}, \tag{2.66}$$

where t is the simulation time and R is the relative error of the estimator. A higher measure of this parameter indicates faster convergence for the problem.

CADIS

It is arguable that CADIS (consistent adjoint driven importance sampling) is the most well-known and popular method for implementing importance sampling theory in a Monte Carlo simulation.[8] CADIS performs a discrete ordinates calculate to first find scalar adjoint flux values over a mesh to then set weight windows in a way equivalent to the zero variance importance sampling scheme as presented earlier. As such, the survival weights of the weight windows are set as follows, based on equation 2.63:

$$W_s(\Delta\vec{r}, \Delta E) = \frac{F}{\phi^\dagger(\Delta\vec{r}, \Delta E)}, \tag{2.67}$$

where, typically, $W_l = \frac{W_s}{3}$ and $W_u = \frac{5W_s}{3}$.[1][8] Here, $\phi^\dagger$ is used as opposed to $E^\dagger$ in order to clarify that CADIS specifically seeks to utilize the adjoint flux as opposed to a different specific adjoint event density function used in the zero variance proof expressed earlier. In order to keep these weight windows consistent with the weight of particles being emitted from the source (such that particles do not immediately have stochastic termination or splitting immediately applied), the source is also modified as follows:

$$\hat{S}(\vec{r}, E) = \frac{\phi^\dagger(\Delta\vec{r}, \Delta E) S(\vec{r}, E)}{F}. \tag{2.68}$$

CADIS has been repeatedly shown to achieve an increase in the figure of merit for a variety of Monte Carlo simulations. However, a possible improvement for this method would be to include directional dependence of the importance sampling expressed in equation 2.63. There has been work to include direction information in CADIS such as CADIS-Ω, which utilizes scalar contributon flux, $C = \int \phi(Y)\phi^\dagger(Y) d\hat{\Omega}$, in the importance function with the idea that this function

should have a greater value in areas of phase space where the forward and adjoint flux have greater overlap in direction space.[15] However, this method still utilizes an importance function that is not dependent on direction space since the weight windows it generates are not discretized over direction space.

Other Methods

Other methods similar in nature to CADIS include AVATAR, LIFT (local importance function transform), and the method used in TRIPOLI-4.[9][16][10]. All of these methods use a deterministic adjoint solution in order to increase the figure of merit for a Monte Carlo calculation. In AVATAR, the scalar flux and net current information are calculated from a discrete ordinates solution and used in order to form angularly dependent weight windows.[9]. The LIFT method utilizes adjoint scalar flux in order to form weight windows, an importance sampled source, and an importance sampled transport kernel.[16].

The method utilized in TRIPOLI-4 by Nowak also utilizes a discrete ordinates solution.[10] It uses this solution for importance sampling of the transport kernel, but it also uses adaptive multilevel splitting which relies on an importance map instead of weight windows.[10] MCNP also contains importances, but they must be assigned to the cells defined by the geometry instead of a mesh that overlays the geometry.[1] Adaptive multilevel splitting is particularly interesting because while it utilizes a deterministic adjoint solution to set the importance map, it also uses information from the histories performed by the Monte Carlo simulation to adapt the splitting routine.[10][17]

In summary, all of these different methods utilize deterministic solutions in order to improve convergence of a Monte Carlo simulation. While some do include

angularly dependent weight windows (such as AVATAR), none seem to use actual angular adjoint information in order to set these parameters. In the work by Wagner on CADIS, he states that "The S_N method can determine the scalar (angular-independent) adjoint accurately, but not necessarily the angular-dependent adjoint because of the limited number of directions. Therefore, because of the memory requirements and inaccuracies of the angular-dependent adjoint, we use the space- and energy-dependent (scalar) adjoint function for calculating space- and energy-dependent source biasing and weight window parameters".[8] In light of this, an area of interest would be to simply explore how direction dependent importance information would affect the figure of merit. This is a major topic of this thesis and will be explored in chapter 3.

In addition to this, a similar method to CADIS will be applied to adjoint continuous energy Monte Carlo. This method will use the forward information in an importance function for the adjoint since, after all, the forward is the adjoint of the adjoint function.[4] This will be explored in chapter 4. This will be explored utilizing both a discrete ordinates calculation of the forward information and the Monte Carlo mesh estimator results for the forward information.

Chapter 3

Direction Space Importance Sampling

This chapter explores enhancement of importance sampling based variance reduction techniques through the inclusion of direction discretization and angular adjoint information. First, a new type of importance sampling approximation implementation, henceforth called "weight targets", will be explored due to theoretical questions regarding weight windows. Subsequently, a method by which direction space can be discretized for both estimators and meshes that utilize weight windows or weight targets is described. Lastly, the results of the direction discretization and weight targets will be displayed for an example problem to demonstrate their efficacy towards their intended purpose.

3.1 Weight Targets

Though weight windows have proven to be a reliable method of increasing the efficiency of Monte Carlo simulations when used appropriately, they were originally intended as a way to restrict the range of possible weights in a simulation.[1] Presumably, a wide range of weights would have been caused by various variance reduction techniques being used simultaneously in a given problem. However, when used as a variance reduction technique itself, it raises the question of why an appropriate range of weights defined by the weight window itself is required when one can calculate an exact value for the appropriate weight to be expected of a particle in a certain phase space given an applied importance sampling scheme. It is,

in fact, this exact weight value that is utilized as the survival weight for weight windows generated by CADIS which is exactly at the center of the weight window.[8] Instead, we can choose to effectively "close" the weight window since we already know exactly what weight we should expect for a given portion of phase space. This leads to a relatively simple design. For some phase space, let there be an expected weight for the weight target designated as $W_t(\mathbf{Y})$. A particle entering $\mathbf{Y}$ has a weight w. The fraction $\gamma = \frac{w}{W_t(\mathbf{Y})}$ is immediately calculated. Thus let the following be true:

- If $\gamma > 1$, split particle:
  - $P\left(n_{split} = \lceil \gamma \rceil\right) = \gamma - \lfloor \gamma \rfloor$
  - $P\left(n_{split} = \lfloor \gamma \rfloor\right) = 1 - P\left(n_{split} = \lceil \gamma \rceil\right)$
- If $\gamma < 1$, stochastically terminate particle:
  - $P_{terminate} = 1 - \gamma$
  - $P_{survive} = \gamma$

Here, P is a probability and n_{split} is the number of particles that an individual particle will split into. In any case, whether splitting or stochastically terminating, every particle either has a resulting weight of $W_t(\mathbf{Y})$ or has been terminated, thus allowing all non-terminated particles in a phase space $\mathbf{Y}$ to have the weight expected of that phase space (except in the case of $\gamma >> 1$, which will be explained later). It is of note that this is effectively the same procedure as that detailed in subsection 2.6, but instead uses the weight of the particle to determine γ instead of the importance function $I(\mathbf{Y})$.

There is another method that is quite similar to this, which are importances in MCNP.[1] While in MCNP this technique must be applied to geometrical objects

defined by the combinatorial geometry, theoretically there is no issue with applying them to a mesh overlaying the geometry similar to how weight windows can be currently applied in MCNP.[1] There are a couple of reasons these weight targets are used instead.

The first reason is that accounting for other variance reduction methods is easier when using weight as a variable to determine the splitting/termination procedure. All variance reduction methods alter the weight of particles in some predictable fashion in order to avoid biasing any estimators. If one wanted to apply other variance reduction techniques in such a way that alters the importance sampling techniques presented in this work, then they would more easily be able to accomplish this by using weight-based importance sampling when compared with using importances. This is due to that predictability of the other variance reduction method's weight transformation since weight-based importance sampling directly acts based on the weight of the particle and can easily take said weight transformations into account when weight-based importance sampling is applied. Importances instead would have to be aware of how the other variance reduction method is working to transform weight and take that into account when being applied, so its not quite as simple when using importances.

The second reason has to do with a commonly added constraint to limit the maximum number of particles that can result from a splitting operation. This constraint is typically added to avoid situations where the magnitude of γ is such a value that would produce an exceptionally large amount of additional particles to simulate if left unchecked. If there is no limit to the number of resulting particles, then the application of weight targets produces equivalent results when compared to the application of importances. With the application of a maximum number of allowed resulting particles though, they change slightly in their resulting effects.

Importances rely on phase space transitions to determine the fraction γ mentioned earlier (where $\gamma = \frac{I(\mathbf{Y})}{I(\mathbf{Y}')}$ for a particle starting at $\mathbf{Y}'$ and ending at $\mathbf{Y}$ when using importances). Weight targets only rely on the current phase space location $\mathbf{Y}$ of a particle to determine γ, though. Due to importances functioning based on phase space transition, one can foresee the following scenario:

1. Particle transports from $\mathbf{Y}''$ to $\mathbf{Y}'$.
2. $\gamma = \frac{I(\mathbf{Y}')}{I(\mathbf{Y}'')} > n_{max}$, where n_{max} is the maximum number of allowed resulting particles.
3. Resulting particle transports from $\mathbf{Y}'$ to $\mathbf{Y}$.
4. $I(\mathbf{Y}') \approx I(\mathbf{Y})$.
5. $\gamma \approx 1$, minimal further splitting occurs.

Therefore, if a particle transports from an region with an extremely low importance value to a region of extremely high importance, it is foreseeable that this maximum number of splits will be hit in the splitting routine and thus limit the amount of possible future splitting for resulting particles that remain in the high importance region. This is not the case with weight targets. Since the weight of the particle is used in comparison with the weight target assigned to the current phase space that particle exists in, it will keep splitting until finally the particle reaches the target weight it is supposed to have. This is a more desirable behavior to ensure that further samples are gained in regions of high importance.

3.2 Direction Discretization

In order to discretize direction space, a direction quadrature for discrete ordinates called PQLA (piecewise quasi-linear angular) will be used for its convenient description that allows for a simple discretization of the unit sphere. Reference [18] provides a fundamental overview of the PQLA quadrature.

PQLA Quadrature Fundamentals

For the purposes of this work, only the first type of PQLA will be utilized. The first step of this quadrature is to inscribe an octahedron within the unit sphere such that the vertices of the octahedron lie on the Cartesian dimension axes. Next, given a quadrature order N, the octahedron is divided by $2N-1$ planes along each Cartesian dimension from -1 to 1 (including planes at -1 and 1), orthogonal to said dimension. This results in each triangular face of the inscribed octahedron being divided into smaller triangles of equal areas, such as in figure 3.1 where the quadrature order is 2

We denote the plane indices for each individual Cartesian dimension as i_x, i_y, i_z, and utilize i as notation of a generic index for properties that apply to all dimensions in the same manner to avoid unnecessary duplication of equations. The indices then have the following domain:

$$i = [-N, -N+1, ..., N-1, N]. \tag{3.1}$$

Noting the pattern in figure 3.1, the coordinate of each plane on its respective axis is easily seen. Let us denote s as a generic coordinate representing x, y, or z for properties that apply to each dimension, just as i represents a generic index

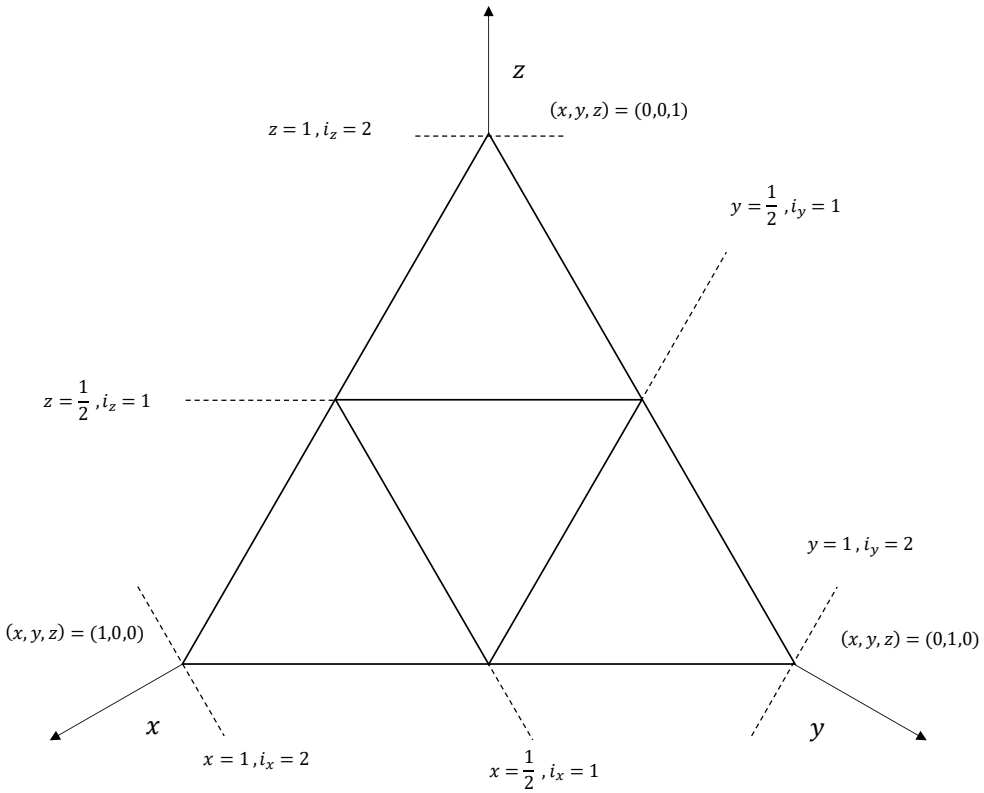


Figure 3.1: Triangular octahedral face subdivided by orthogonal planes with $N = 2$

placeholder for the individual dimension indices. The following is then true:

$$s = \frac{i}{N} \tag{3.2}$$

A triangle is thus denoted as T. For every triangle T, the following relation holds for the plane indices that correspond to the sides of the triangle:

$$N \pm 1 = |i_{x,T}| + |i_{y,T}| + |i_{z,T}|. \tag{3.3}$$

A vertex where these planes intersect on an octahedron face is denoted as $\vec{V}$. Given vertex $\vec{V}$, the following relation holds for said vertex's planar indices:

$$N = |i_{x,\vec{V}}| + |i_{y,\vec{V}}| + |i_{z,\vec{V}}|, \tag{3.4}$$

and for the vertex's spatial coordinates:

$$1 = |x_{\vec{V}}| + |y_{\vec{V}}| + |z_{\vec{V}}|. \tag{3.5}$$

The vectors from the origin to the vertices on the octahedral faces can be found by noting that the surface of this inscribed octahedron represents the space that contains all vectors that originate from the origin and are normalized to the 1-norm for 3-dimensional space. This means that placing the planar indices of a given vertex $\vec{V}_{oct}$ into a vector and normalizing said vector to the 1-norm will result in a vector from the origin to $\vec{V}_{oct}$. The generic definition of the norm for 3-dimensional vectors is as follows:[19]

$$\|\vec{v}\|_n = \left(\sum_{j=0}^{3} |v_j|^n \right)^{1/n}. \tag{3.6}$$

For the purposes of this paper, only the 1-norm and 2-norm (denoted as $\|\cdot\|_1$ and $\|\cdot\|_2$) are important. It is immediately apparent that $\|\cdot\|_2$ simply represents the magnitude of a vector in 3-dimensional space, which is of note because this means that the unit sphere is simply the space that contains all vectors which originate from the origin that are normalized to $\|\cdot\|_2$. It is also of note, based on relationship 3.5, that the following is true.

$$1 = \|\vec{V}_{oct}\|_1 \quad \forall \quad \vec{V}_{oct}. \tag{3.7}$$

The reason that this is important is because, in order to find the equivalent vertices on the surface of the sphere, $\vec{V}_{oct}$ need simply be normalized to $\|\cdot\|_2$, or do as follows:

$$\vec{V}_{sph} = \frac{\vec{V}_{oct}}{\|\vec{V}_{oct}\|_2}. \tag{3.8}$$

This allows us to utilize expression 3.4 and 3.5 to search for a direction bin that a particle is currently contained within, as dividing equation 3.4 by N simply recovers a vector normalized in the 1-norm that exists on the octahedral surface. Converting between the norms is necessary to perform this determination of direction element index, as shown in subsection "Direction Discretization Estimator Indexing". It is also useful in constructing the spherical triangles, as shown in subsection "Triangle Set Construction, using the plane indices so that source direction biasing can be performed in section 3.4.

Triangle Set Construction

The direction-based importance sampling of the source requires the explicit construction of the spherical triangles formed by this quadrature. We start by considering only a single octahedral face in the positive xyz octant. First, a standard method of indexing these triangles must be set. In this work, we divide the triangles into parallel rows starting at the xz plane initially containing $2N - 1$ elements and decreasing by 2 in row size until the row only has 1 triangle with a vertex on the y axis. We start with the bottom-left triangle and move towards the upper-right triangle on a given row. Note figure 3.2 where $N = 3$ for a graphical example. T_{loc} will be used to define the local triangle index on a given octahedral face - specifically the face that includes the positive xyz axes since this information is symmetric for every other face.

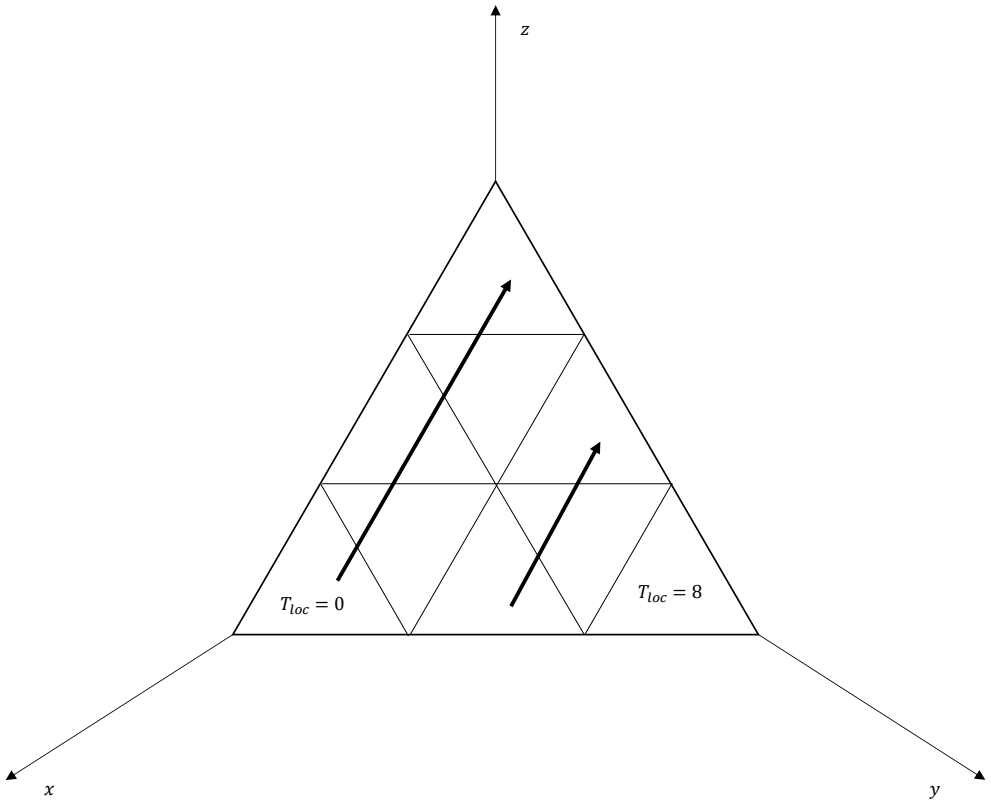


Figure 3.2: Indexing Explanatory Graphic with N = 3

In order to compute and store relevant information about the spherical triangles, the spherical triangle vertices need to be processed and stored relative to the triangle indices. We first note that for PQLA quadrature, there are always two types of triangles in terms of their orientation - those "pointing" in the positive z direction and those "pointing" in the negative z direction. We also note that every row as we defined earlier begins and ends with a triangle pointing in the positive direction. We then define an algorithm that finds these vertices by looping over the vertices of each triangle in the counter-clockwise direction, as depicted in figure 3.3.

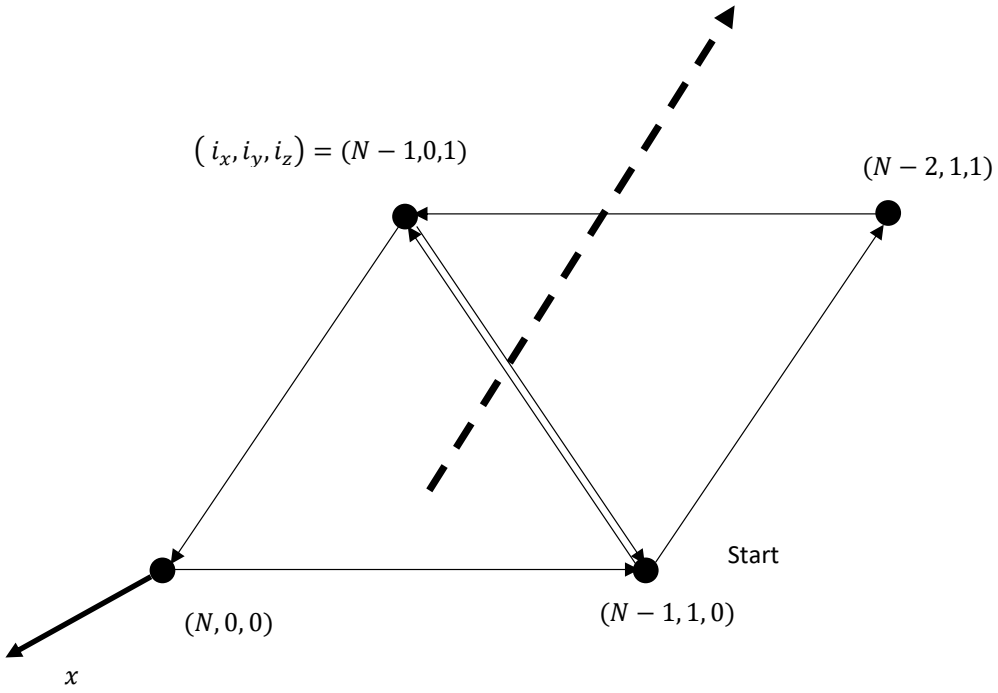


Figure 3.3: Vertex Process Order Graphic

In this work, the processing of each triangle begins with the vertex that lies on the x axis. Given that normalizing a vertex vector to $\|\cdot\|_1$ is only multiplying a vector containing the plane indices by a scalar, there is no need to normalize them to $\|\cdot\|_1$ before finding their $\|\cdot\|_2$ normalized counterpart. Noting this and designating $\mathrm{M}_{\mathtt{tri}}$ as the number of triangles in a given row and T_{row} being the index of the current triangle being processed in a given row, the following is the algorithm used to process these triangle vertices and begin to construct triangle "objects" that contain relevant information for each spherical triangle:

Result: Find and Process Triangle Vertices

for $row = 0; row < N; ++ row$ **do**

  $M_{tri} = 2N - 1 - 2row$

  $i_x = N - row$

  $i_y = row$

  $i_z = 0$

  for $T_{row} = 0; T_{row} < M_{tri}; ++ T_{row}$ **do**

    $V_1 = (i_x, i_y, i_z)$

    if $\mathrm{mod}(T_{row}, 2) == 0$ **then**

      $V_2 = (i_x - 1, i_y + 1, i_z)$

      $V_3 = (i_x - 1, i_y, i_z + 1)$

      $i_x = i_x - 1$

      $i_z = i_z + 1$

    else

      $V_2 = (i_x, i_y + 1, i_z - 1)$

      $V_3 = (i_x - 1, i_y + 1, i_z)$

    end

    $V_1 = \frac{V_1}{\|V_1\|_2}$

    $V_2 = \frac{V_2}{\|V_2\|_2}$

    $V_3 = \frac{V_3}{\|V_3\|_2}$

    // Store triangle information

  end

end

Algorithm 1: Triangle Vertex Processing

After these vertices are stored, one may use basic spherical geometry relationships to calculate quantities such as the triangle area, side-length, angles, etc. The

other 7 octants can be processed by transforming the positive-domain octant onto the other octants and indexing appropriately.

Direction Discretization Estimator Indexing

The next step to this is to remove any negative signs of each direction dimension value for a particle as follows, where Ω_x, Ω_y, and Ω_z are the components of a particle direction vector:

$$\hat{\Omega}_+ = (|\Omega_x|, |\Omega_y|, |\Omega_z|) \quad , \tag{3.9}$$

such that $\hat{\Omega}_+$ is now a vector that lies in the octant with all positive dimensions. This vector is then renormalized to $\|\cdot\|_1$ as follows:

$$\hat{\Omega}_{+,1} = \frac{\hat{\Omega}_+}{\|\hat{\Omega}_+\|_1}. \tag{3.10}$$

The reason for this is simply that renormalizing it to $\|\cdot\|_1$ allows for it to be contained within the original axis-orthogonal planes that intersect the octahedron to form the triangles. The original vector $\hat{\Omega}_+$ is normalized to $\|\cdot\|_2$ which is contained under the spherical triangle that results from a projection from the triangle formed on the octahedral face, but is not necessarily contained within those planes. The next step is to determine the lower-bounding planes of the triangle. It is of note that the three lower bounding planes either include the sides of a given octahedral triangle or intersect at the vertices, depending on the orientation of the triangle. For example, see the central triangle in figure 3.1. Because of the orientation of this triangle, the lower bounding planes are actually the ones that intersect at the vertices as opposed to the others where the lower bounding planes are the ones that

form the sides. With this, it's relatively straight forward to find the lower-bounding place indices:

$$\vec{i} = \left\lfloor N\hat{\Omega}_{+,1} \right\rfloor . \tag{3.11}$$

Now that i_x, i_y, and i_z are known, the triangle index can be calculated. First, noting that there are 2 fewer triangles per row, the complexity of this algorithm can be made constant through a bit of algebra. Let N_{tri} be the total number of triangles up to row j:

$$N_{tri} = \sum_{j=0}^{i_y-1} (2(N-j)-1) \tag{3.12}$$

$$= 2i_y N - i_y - 2\sum_{j=0}^{i_y-1} j = 2i_y N - i_y - i_y(i_y-1) = i_y(2N-i_y).$$

The local index on a given row is as follows:

$$N + i_z - i_y - i_x - 1. \tag{3.13}$$

As described in section 3.2, the indexing order assumes that the triangles are divided into N rows starting at the yz plane moving towards the x axis. Let the local octant triangle index be T_{loc}:

$$T_{loc} = i_y(2N-i_y) + N + i_z - i_y - i_x - 1. \tag{3.14}$$

After this, the triangle index T_i is easily calculated (k_{oct} is the index of the overall octant that a particle is contained in):

$$T_i = T_{loc} + k_{oct}N^2. \tag{3.15}$$

Secondary Binning

The above indexing procedure is used only for directions in the octant that contains the positive domain of the x, y, and z dimensions. In order to cover the entirety of direction space, it is noted that the positive domain triangle set is reflected perfectly into every other octant. Given this, the index of the octant can be determined separately from the local triangle index. We present the below algorithm to find this secondary index for the octant.

Result: Determine k_{oct}

$k_{oct} = 0$

if $\Omega_x < 0$ **then** $k_{oct} = k_{oct} + 1$

if $\Omega_y < 0$ **then** $k_{oct} = k_{oct} + 2$

if $\Omega_z < 0$ **then** $k_{oct} = k_{oct} + 4$

return k_{oct}

Algorithm 2: Secondary Index Algorithm

Processing triangles of other octants

In order to then store the vertices for the other octants, one need only utilize a consistent indexing scheme for the triangles of other octants and the fact that the numerical coordinates of the vertices only change in sign between the different octants. In this work, the following algorithm was used to determine other octant

indices (& is used as the bitwise-AND operator in this case):

Result: Store other octant vertices

for $octant = 1; octant < 8; ++octant$ **do**

  $x_m = 1$

  $y_m = 1$

  $z_m = 1$

  if $octant \,\&\, 1$ **then**

    $x_m = -1$

  if $octant \,\&\, 2$ **then**

    $y_m = -1$

  if $octant \,\&\, 4$ **then**

    $z_m = -1$

  for $triangle = 0; triangle < N; ++triangle$ **do**

    for $vertex = 0; vertex < 3; ++vertex$ **do**

      $V_{1,new} = x_m V_{1,tri,vert}$

      $V_{2,new} = y_m V_{2,tri,vert}$

      $V_{3,new} = z_m V_{3,tri,vert}$

    end

    // Store triangle however is desired

  end

end

Algorithm 3: Processing Other Octants

3.3 Estimator Results as the Importance Function

For the demonstration of this method, we are utilizing the estimator results of the adjoint Monte Carlo simulation to form the importance function $I(\mathbf{Y})$. For the purposes of clarity, we denote the adjoint estimator result as $\hat{\chi}^{\dagger}(\Delta \mathbf{Y}_i)$, noting that

the track length flux estimator for the adjoint simulation is, in fact, measuring an estimate of the function adjoint to the forward emission density. The importance function for the forward is thus as follows:

$$I(\Delta \mathbf{Y}_i) = \frac{\hat{\chi}^{\dagger}(\Delta \mathbf{Y}_i)}{A_{\Delta Y_i}}, \tag{3.16}$$

where $A_{\Delta Y_i} = \int_{\Delta Y_i} d\mathbf{Y}$ is a constant that describes the overall bin "size" over all phase space dimensions. Since Monte Carlo estimator results are integral in nature, this constant is necessary to ensure that each importance function value is properly scaled with the size of the bin itself. For instance, if the mesh is discretized over space, energy, and direction (as is the case here), then given a mesh index of j, energy element index of k, and direction element of l, $A_{\Delta Y_i} = V_j \Delta E_k \Delta S_l$, where V_j is the mesh element volume, ΔE_k is the energy element size, and ΔS_l is the surface area of the spherical triangle describing the direction element. These importances can then be used in a manner equivalent to CADIS with direction information, where the weight target is set to the same value as the survival weight of a weight window, $W_t(\mathbf{Y}) = W_s(\mathbf{Y})$.

It is important to mention that typically it is the case that the estimator result is already normalized to the volume of the region it encompasses, so typically one need not include the volume when calculating $A_{\Delta Y_i}$.

3.4 Source Direction Importance Sampling

In order to ensure that the outgoing weight of particles from the source match the weight windows or weight targets of the phase space region that they're emitted from, the source must have a method of importance sampling over direction that

matches the spherical triangles from section 3.2. Determining which triangle to sample from is trivial - it can simply be expressed as a discrete distribution with each element of the distribution being a spherical triangle element on the unit sphere which has values matching the estimator results. However, the individual triangles must also be isotropically sampled once the triangle to sample from is known. If the triangles in use have probabilities of generating a directional sample equal to their respective spherical triangle area, then the source is isotropic. One can approximate nonisotropic sources by changing the values of these probabilities, but the method detailed here only allows for isotropic sampling within a given triangle.

In order to do this, the method of isotropically sampling from a spherical triangle by Arvo is used.[20] We first label the angles and corners of the spherical triangle as in figure 3.4 taken from the work by Arvo.[20] The vectors $\mathbf{A}$, $\mathbf{B}$, and $\mathbf{C}$ are easily found through the algorithms detailed in section 3.2. The values of a, b, and c can be found by using simple dot product relationships while α, β, and γ are found through spherical trigonometry. Lastly, we denote the area of a spherical triangle as ω which is as follows:

$$\omega = \alpha + \beta + \gamma - \pi. \tag{3.17}$$

Using this information and two random variables ξ_1 and ξ_2, the algorithm for finding $\hat{\Omega}$ of an outgoing particle from the source is described in the following algorithm.

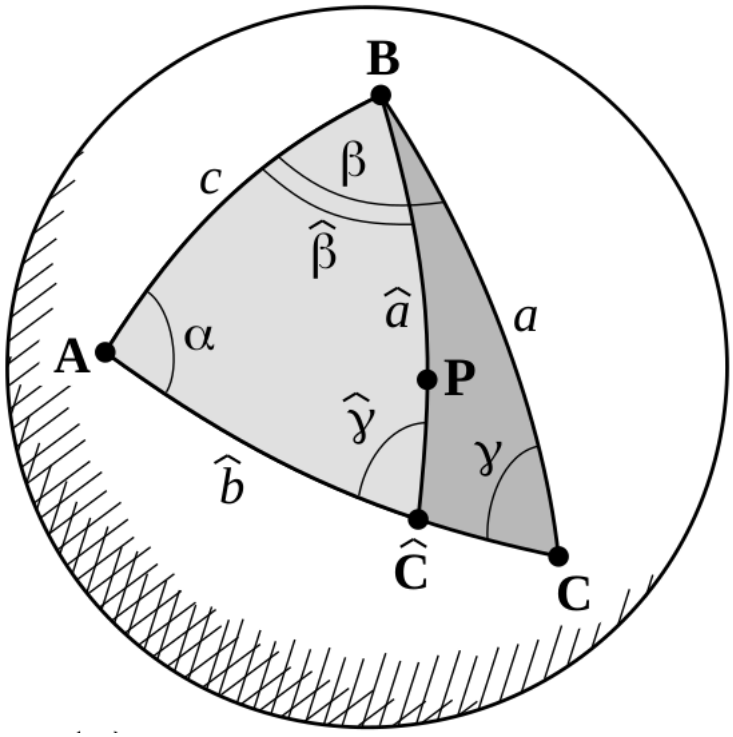


Figure 3.4: Spherical Triangle

Result: Find Source Particle Direction $\hat{\Omega}$

// find random fraction $\hat{\omega}$ of the total triangle area ω

$\hat{\omega} = \xi_1 \omega$

// store values associated with calculating $\cos(\hat{b})$

$s = \sin(\hat{\omega} - \alpha)$

$t = \cos(\hat{\omega} - \alpha)$

$u = t - \cos(\alpha)$

$v = s - \sin(\alpha)\cos(c)$

// calculate $\cos(\hat{b}) = q$

$q = \frac{[vt - us]\cos(\alpha) - v}{[vs + ut]\sin(\alpha)}$

// calculate $\hat{C}$

$\hat{\mathbf{C}} = q\mathbf{A} + \sqrt{1 - q^2}\frac{\mathbf{C} - (\mathbf{C}\cdot\mathbf{A})\mathbf{A}}{\|\mathbf{C} - (\mathbf{C}\cdot\mathbf{A})\mathbf{A}\|_2}$

// find random distance along greater circle arc length from B to $\hat{C}$

$z = 1 - \xi_2\left(1 - \hat{\mathbf{C}}\cdot\mathbf{B}\right)$

// use arc length to calculate $\hat{\Omega}$

$\hat{\Omega} = z\mathbf{B} + \sqrt{1 - z^2}\frac{\hat{\mathbf{C}} - \left(\hat{\mathbf{C}}\cdot\mathbf{B}\right)\mathbf{B}}{\|\hat{\mathbf{C}} - \left(\hat{\mathbf{C}}\cdot\mathbf{B}\right)\mathbf{B}\|_2}$

Algorithm 4: Spherical Triangle Isotropic Sampling

Given this algorithm, the source can be made once again such that the emission

weight is within bounds of the importance sampling technique being used. This is so long as the source is approximated to be isotropic within each respective triangle. This method effectively results in a piecewise constant discrete PDF over the triangle indices, where each triangle is individually isotropically sampled. Presumably, once importance sampling is applied, the discrete importance sampled PDF will be anisotropic while each triangle itself will still be isotropically sampled after the discrete importance sampled PDF is sampled. The weight would be determined by the discrete importance sampled PDF and whatever original discrete PDF is applied over the spherical triangle indices by the user.

3.5 Demonstration

The demonstration utilizes photons exclusively. Figure 3.5 depicts the geometry used for demonstrative purposes in this work. The entire geometry size is $50\text{cm} \times 50\text{cm} \times 50\text{cm}$. There are two plates of natural lead on either end of the x axis of the geometry, spanning the entirety of the y and z axes that are each 8cm thick (in this case, natural refers to the natural composition of isotopes for that specific element). These were placed in order to encourage greater direction biasing of the forward and adjoint sources since, theoretically, any particle that travels in the incorrect direction would end up wasting time in these high attenuation regions where the particles are bound to only disappear if they reach the other side. In the center of the geometry, there is another lead block that is 4cm thick in the x direction, 15cm tall in the y direction, and spans the entirety of the z axis for the geometry. This was constructed so that the forward and adjoint particles will reach the detector and source, respectively, by going around this shield in the y direction. The source and detector are made of natural manganese and germanium respectively, and are

$2cm \times 2cm \times 2cm$ in size. They are placed such that they touch the surfaces of the lead plates, are in the center of the y and z axes and are completely contained within estimator mesh cells (and, therefore, the weight target mesh since both meshes must be the same). The volumetric cell surrounding the center lead plate, source, and detector is natural hydrogen. The density of the lead is $11.35\frac{g}{cm^3}$, manganese $7.26\frac{g}{cm^3}$, and germanium $5.5\frac{g}{cm^3}$. The density of hydrogen was set to a value much higher than what is normally for STP in order to encourage more attenuation in the gaseous medium so that particles do not almost always escape the geometry if they're not traveling in the direction towards a lead shield.

The mesh used in this problem is composed of $2cm \times 2cm \times 2cm$ mesh elements, and is identical for the weight targets, weight windows, and estimator. The mesh estimator has energy bin boundaries at $[0, 0.045, 0.1, 0.2, 0.3, 0.4, 0.6, 0.8, 0.835]MeV$, which match the energy bounds that ADVANTG uses to apply its weight windows per the Denovo discrete ordinates energy boundaries.[21] The forward source is isotropic and has an emission energy of $0.835MeV$, and the response function (equivalent to the adjoint source) is simply 1 in the detector region from $0MeV$ to $0.835MeV$ in order to match a simple total flux calculation in the detector region. For the angular weight windows and weight targets, a quadrature order of $n = 2$ was utilized to form the direction discretization which corresponds to 32 direction bins.

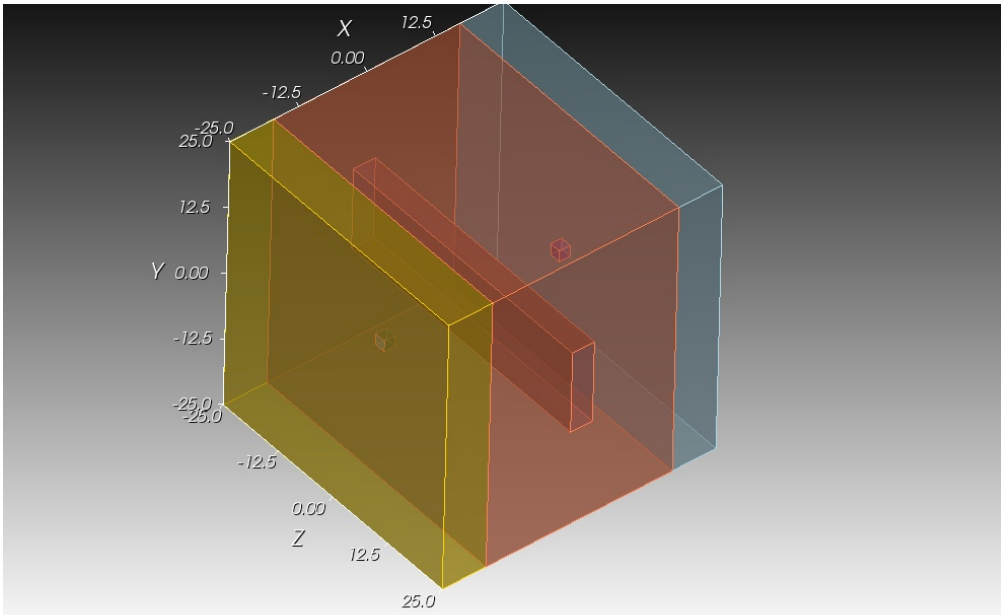


Figure 3.5: Test Geometry

The FOM will be used to measure performance between the different cases presented here. The FOM is a measure of the rate of convergence for a Monte Carlo simulation, given as follows:

$$\text{FOM} = \frac{1}{tR^2}, \tag{3.18}$$

where t is the simulation run time and R is the resulting estimator's relative error.

The adjoint continuous energy Monte Carlo solution took 36.8 hours to form for this problem while the ADVANTG solution was nearly instantaneous due to discrete ordinates performance being dependent on the resolution of discretizations in phase space instead of the attenuation and geometric complexity present in the problem itself. All results here were formed utilizing MPI capabilities of FRENSIE using 7 nodes with 20 cores each.

According to the scalar importance sampling results in table 3.2, there does not seem to be a significant difference in performance for using deterministic and stochastic transport results to form the importance function. Moreover, there does not seem to be a significant advantage to using weight targets over weight windows in these problems. The weight targets seem to have the benefit of gaining a lower relative error in this problem, but end up taking longer to complete in the Monte Carlo simulations. It is not surprising that the simulations using weight targets take longer to complete - since they do not have a range of acceptable weights and require application of the stochastic splitting/termination routines upon every collision, it is easy to see how they would lengthen the time.

What is plain to see is that including the application of a direction discretization has a noticeable benefit in this problem. Utilizing direction-based weight windows and weight targets, a FOM increase of 3 to 5 times seems to be possible in this case. This can be seen in the speedup column, which is a fraction of the FOM result for that importance sampling method with respect to the FOM result in table 3.1 for its respective estimator type being used.

Results

Estimator	*Mean*	*RE*	*FOM*	*VOV*	*Simulation Time (h)*
collision	1.23337120E-06	0.01367071	0.1902866	0.00083786	7.8
track length	1.23042739E-06	0.01144335	0.27157147	0.00033844	7.8

Table 3.1: Forward Results (Analogue, 1e9 histories)

VR Input Source	*Estimator*	*Mean*	*RE*	*FOM*	*VOV*	*Simulation Time (h)*	*Speedup*
Scalar Weight Windows, 1e9 histories							
deterministic	collision	1.26580861E-06	0.00485984	1.65689275	0.00696032	7.1	8.71
	track length	1.26256540E-06	0.00392189	2.54417622	0.00256757	7.1	9.37
stochastic	collision	1.25546240E-06	0.00424709	1.75369081	0.00515392	8.8	9.22
	track length	1.25656698E-06	0.00348308	2.60741841	0.00208084	8.8	9.60
Scalar Weight Targets, 1e9 histories							
deterministic	collision	1.26657542E-06	0.00468733	1.78755678	0.0055168	7.1	9.39
	track length	1.26628780E-06	0.00385106	2.6482024	0.00341617	7.1	9.75
stochastic	collision	1.25748332E-06	0.00415963	1.75284005	0.00562598	9.2	9.21
	track length	1.25660576E-06	0.00336764	2.67424867	0.00206675	9.2	9.85
Angular Weight Windows, 1e9 histories							
stochastic	collision	1.20314546E-06	0.00215156	7.15170958	0.00165113	8.4	37.58
	track length	1.20143032E-06	0.00172456	11.13175643	0.00067279	8.4	40.99
Angular Weight Targets, 1e9 histories							
stochastic	collision	1.20379542E-06	0.00213923	6.17839125	0.00199582	9.8	32.47
	track length	1.20022396E-06	0.00167337	10.09730343	0.000679	9.8	37.18

Table 3.2: Forward Results with Importance Sampling

The following figures display the survival weights generated using ADVANTG, the survival weights generated using Monte Carlo, the errors of the adjoint mesh estimator utilized to generate the survival weights, figures that display the "reverse current" of the adjoint Monte Carlo simulation in that order. The survival weights are used instead of the lower weights due to their values being equivalent to the weight target values used for their respective simulation. Areas in the Monte Carlo simulation where there were no results were replaced by the resulting survival weight calculated using the lowest non-zero Monte Carlo mean result found in the mesh.

The reverse adjoint current was calculated utilizing the centroid vectors of the spherical triangles that are formed from the PQLA quadrature direction discretization as follows:

$$\vec{J}(\Delta\vec{r}_i, \Delta E_j) = \sum_{k=0}^{N} \hat{\Omega}_{c,k} A_k \psi(\Delta\vec{r}_i, \Delta E_j, \Delta\hat{\Omega}_k), \tag{3.19}$$

where this is just a basic approximation of the definition of current utilizing the angular mesh estimator results. As such, $\vec{J}$ is the current, k is the direction bin, $\hat{\Omega}_{c,k}$ is the spherical triangle centroid vector for direction element k, A_k is the area of the spherical triangle of direction element k, and ψ is the flux for that bin. We note that this is the "reverse" current due to the fact that this estimator is actually taking scores for the $-\hat{\Omega}_k$ element due to adjoint pseudoparticles traveling in the $-\hat{\Omega}$ direction respective to forward particles,[4] so ψ is actually the estimator result for the $-\hat{\Omega}$ direction. If this were not the case the forward particles would be encouraged away from the region of interest. Cells with a relative error above 0.2 in the scalar mesh track length flux estimator are excluded from these images, as they tend to exhibit statistical noise. The results are presented on the xy plane at $z = 0$.

Upon examination of the survival weights generated using the CADIS method and comparing them with the survival weights generated by Monte Carlo methods, the greatest difference is the lack of statistical noise in the mesh results for the CADIS results. This results in less abrupt transitions in weight windows or weight targets between different mesh elements. There are also many mesh elements that have no adjoint Monte Carlo estimator data which are made obvious by the zero error values in the respective Monte Carlo error figures and abrupt changes in orders of magnitude of survival weights. This is more apparent in the lower energy groups and the only energy groups that have full sets of data for stochastically formed

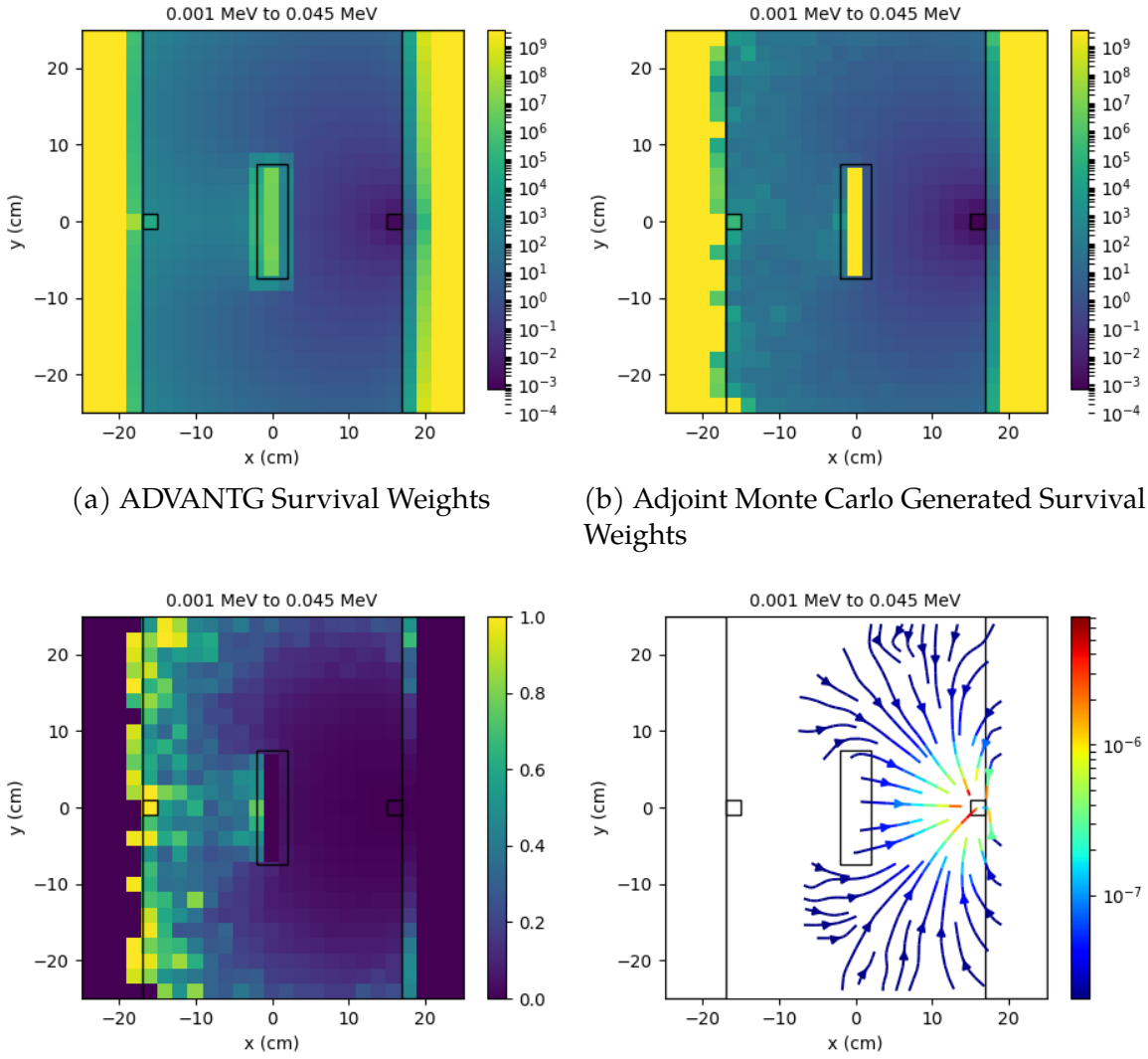
survival weights are those presented in figure 3.12b and 3.13b. There are some ray effects that start to become more noticeable in energy groups 4 through 8 (due to higher energy particles experiencing less attenuation).

In the lower energy groups, there is particularly high error in the Monte Carlo results behind the shield with respect to the detector, as seen in figures 3.6c, 3.7c, and 3.8c (the higher energy groups still have high error behind the shield, but not as high). However, one can assume that the weight windows and weight targets are not as important in these lower energies in this region of space. After all, the forward source is a point in energy that exists in the highest energy group, so it stands to reason that a large amount of particles might not experience these weight windows/targets.

The reverse current figures show exactly what one might expect with respect to the energy groups. In the lower energy groups, the reverse current flows around the shield, which when translated to angular weight windows/targets would mean that they would encourage splitting for particles going around the shield instead of through it. However, as energy increases and attenuation decreases, the reverse current curves less around the shield and simply starts to go straight through the shield, which again when weight windows/targets are applied based on this would mean that they would encourage more particles to simply try to travel directly through the shield - although towards the edge of the shield they still would rather slightly curve around the shield. The highest current (and hence the lowest angular weight windows/targets) are inside the detector region, encouraging particles to split more in that region. Once again, given that data with especially high (or zero) error is omitted from these figures (not in the weight windows/targets applied in the simulations themselves) due to the statistical noise that data presents, we note that there are large chunks of information missing from the lower energy

groups. However, these are again in regions where one might assume they are not as important due to the high energy-point nature of the forward source as mentioned before.

Energy Group 1 Figures (0.001 MeV to 0.045 MeV)



(a) ADVANTG Survival Weights

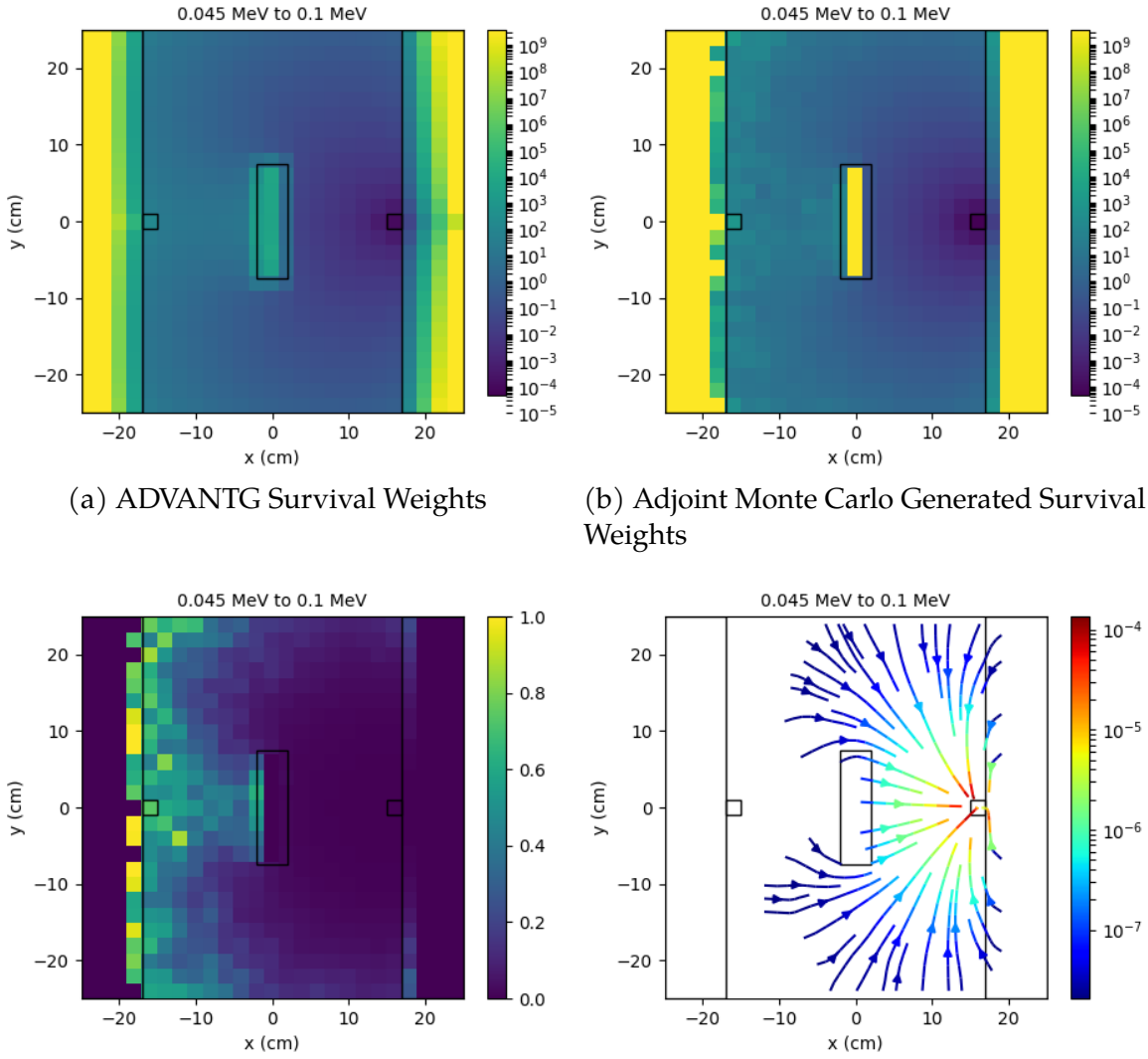
(b) Adjoint Monte Carlo Generated Survival Weights

(c) Adjoint Monte Carlo Relative Error

(d) Adjoint Monte Carlo Reverse Current

Figure 3.6: Forward Variance Reduction, Energy Group 1

Energy Group 2 Figures (0.045 MeV to 0.1 MeV)



(a) ADVANTG Survival Weights

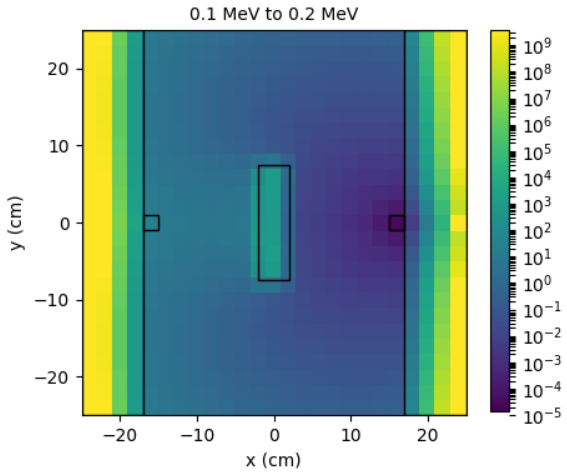
(b) Adjoint Monte Carlo Generated Survival Weights

(c) Adjoint Monte Carlo Relative Error

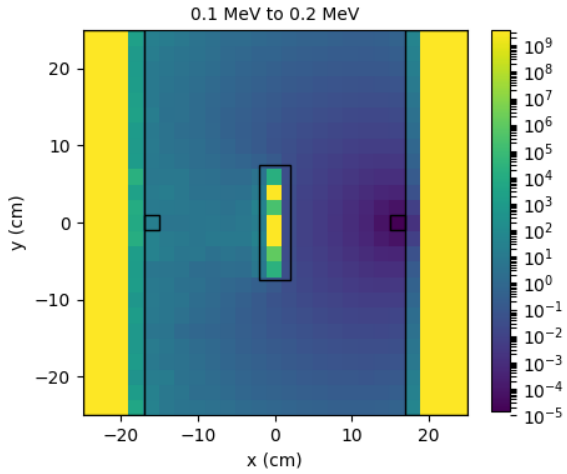
(d) Adjoint Monte Carlo Reverse Current

Figure 3.7: Forward Variance Reduction, Energy Group 1

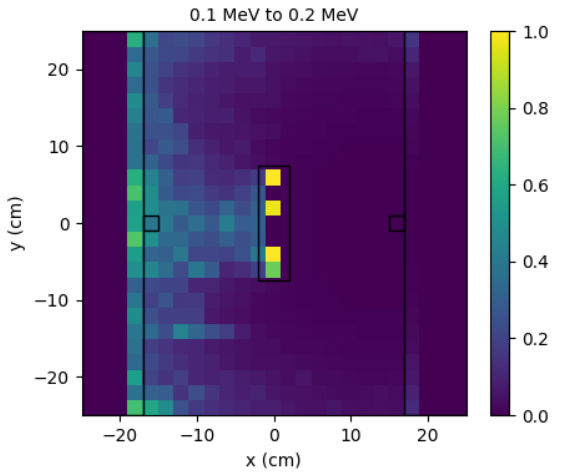
Energy Group 3 Figures (0.1 MeV to 0.2 MeV)



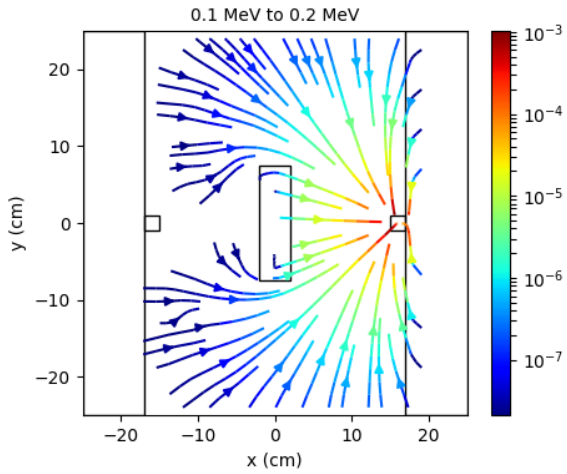
(a) ADVANTG Survival Weights



(b) Adjoint Monte Carlo Generated Survival Weights



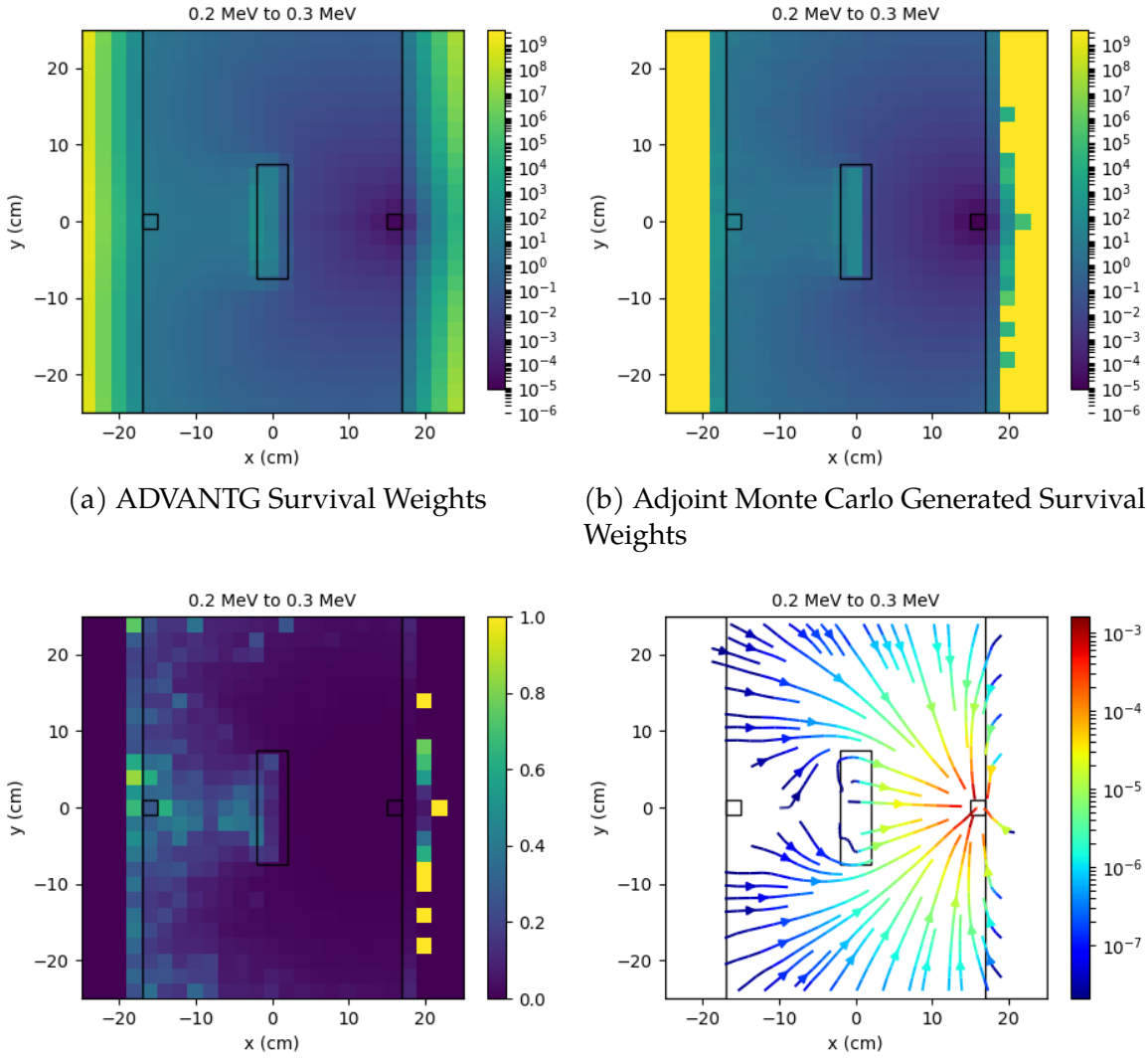
(c) Adjoint Monte Carlo Relative Error



(d) Adjoint Monte Carlo Reverse Current

Figure 3.8: Forward Variance Reduction, Energy Group 3

Energy Group 4 Figures (0.2 MeV to 0.3 MeV)



(a) ADVANTG Survival Weights

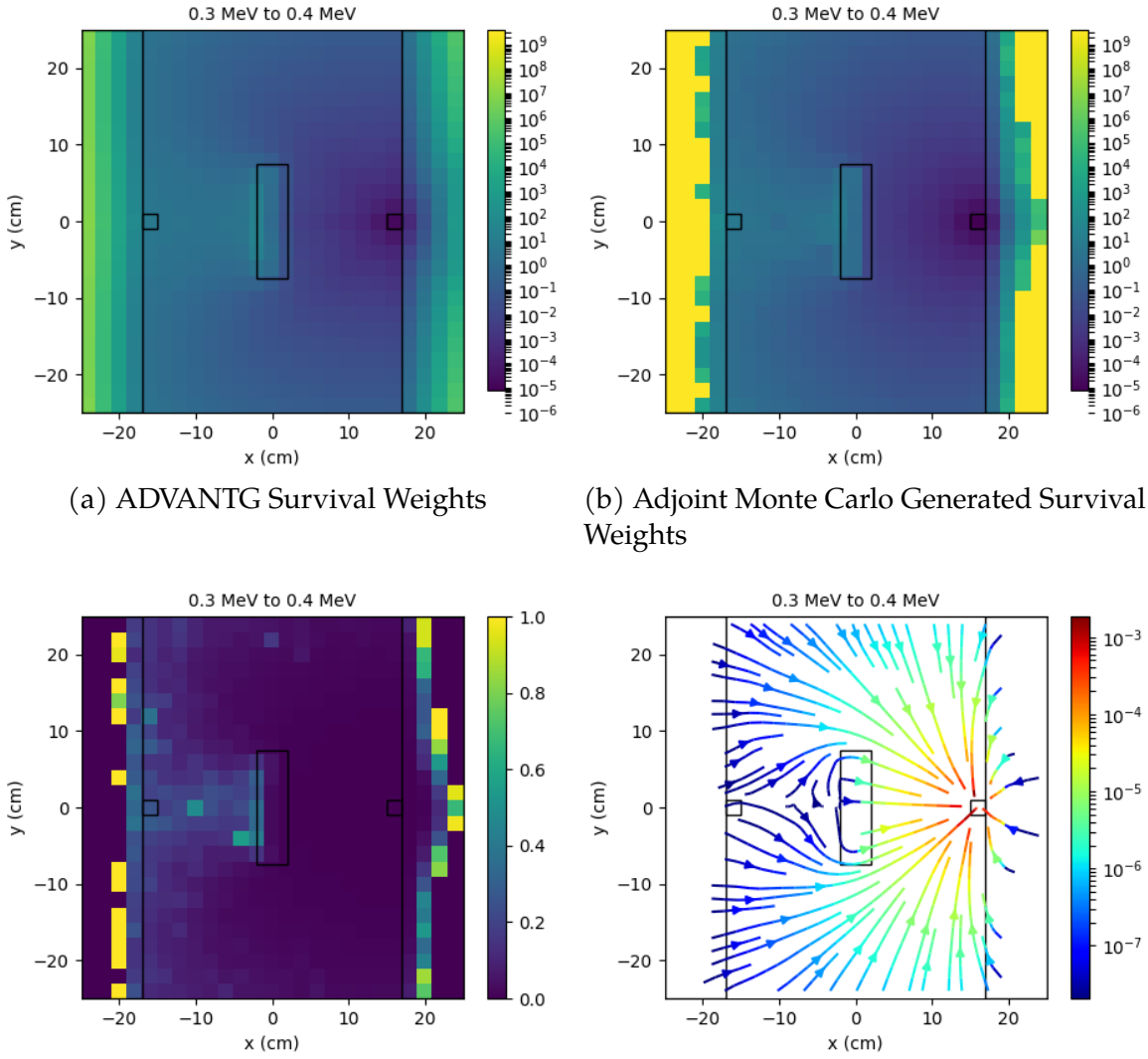
(b) Adjoint Monte Carlo Generated Survival Weights

(c) Adjoint Monte Carlo Relative Error

(d) Adjoint Monte Carlo Reverse Current

Figure 3.9: Forward Variance Reduction, Energy Group 4

Energy Group 5 Figures (0.3 MeV to 0.4 MeV)



(a) ADVANTG Survival Weights

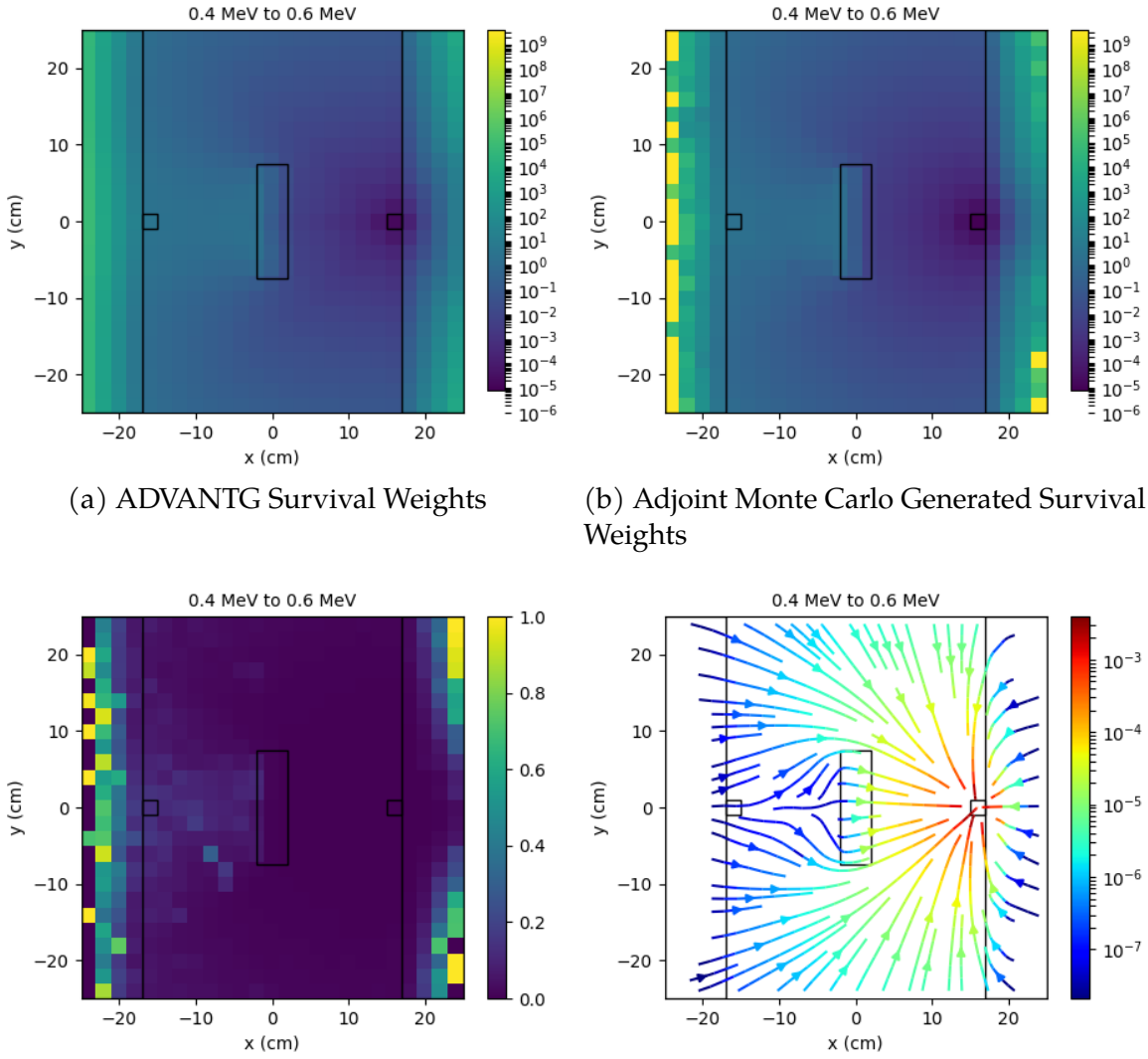
(b) Adjoint Monte Carlo Generated Survival Weights

(c) Adjoint Monte Carlo Relative Error

(d) Adjoint Monte Carlo Reverse Current

Figure 3.10: Forward Variance Reduction, Energy Group 5

Energy Group 6 Figures (0.4 MeV to 0.6 MeV)



(a) ADVANTG Survival Weights

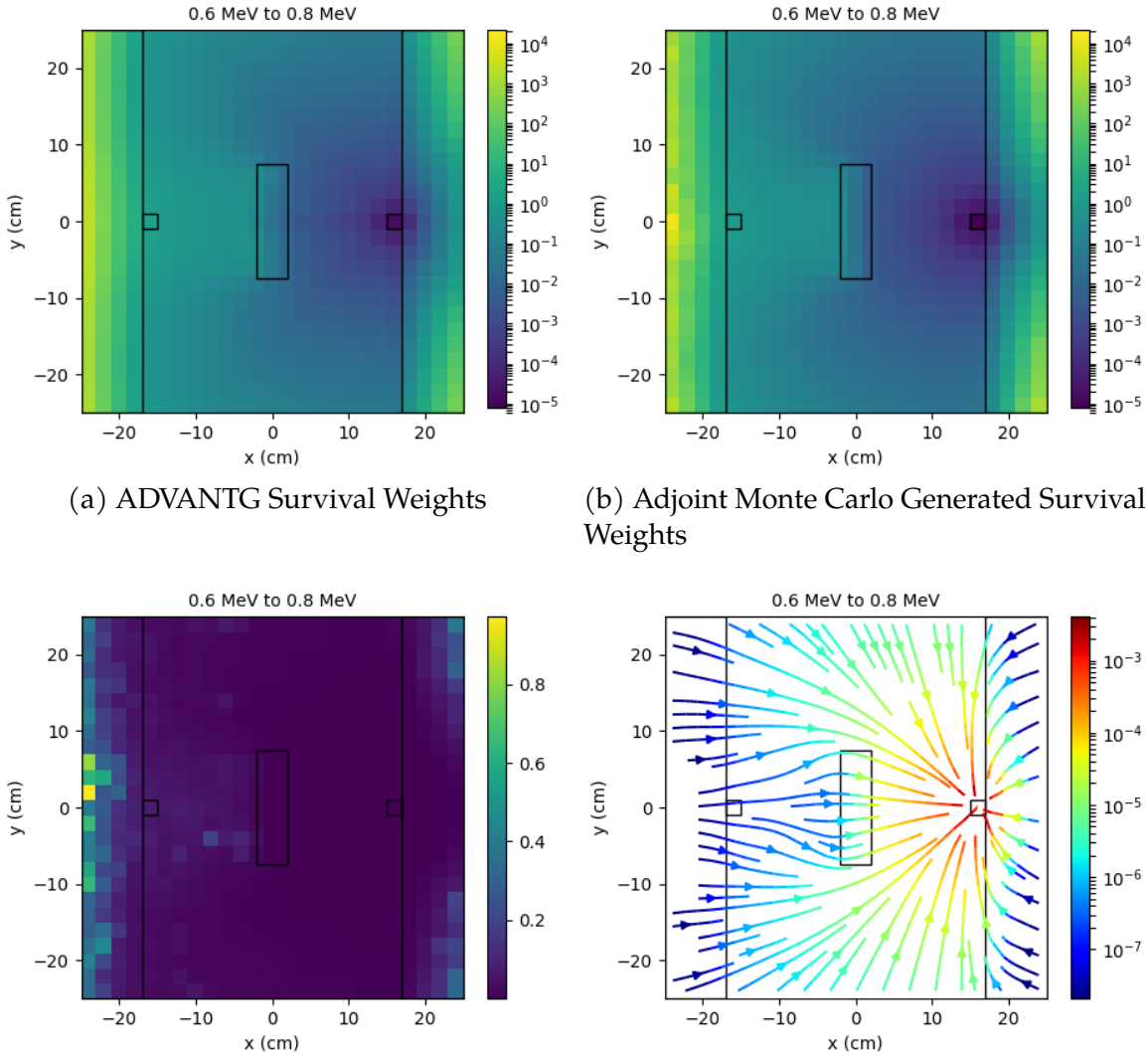
(b) Adjoint Monte Carlo Generated Survival Weights

(c) Adjoint Monte Carlo Relative Error

(d) Adjoint Monte Carlo Reverse Current

Figure 3.11: Forward Variance Reduction, Energy Group 6

Energy Group 7 Figures (0.6 MeV to 0.8 MeV)



(a) ADVANTG Survival Weights

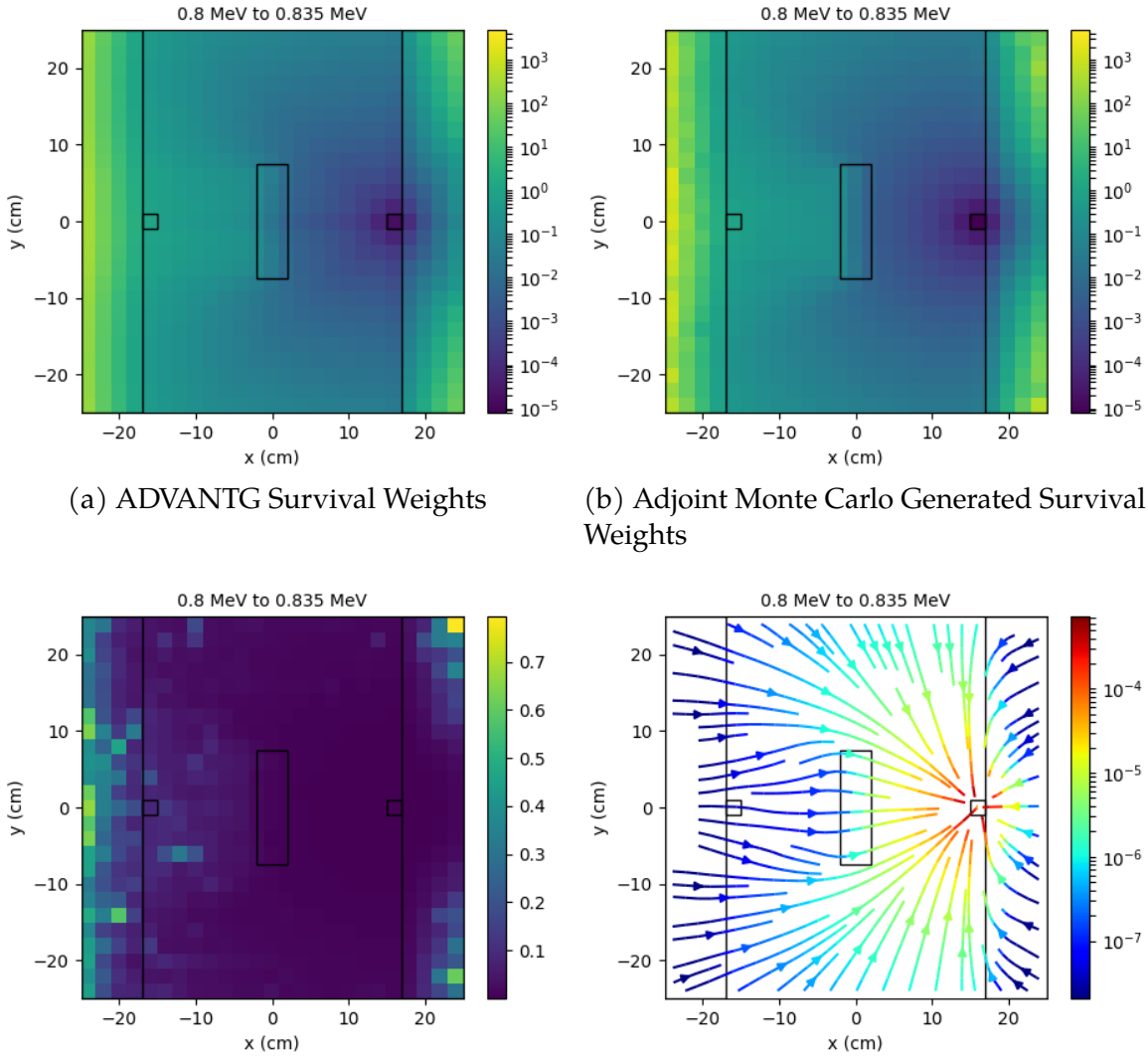
(b) Adjoint Monte Carlo Generated Survival Weights

(c) Adjoint Monte Carlo Relative Error

(d) Adjoint Monte Carlo Reverse Current

Figure 3.12: Forward Variance Reduction, Energy Group 7

Energy Group 8 Figures (0.8 MeV to 0.835 MeV)



(a) ADVANTG Survival Weights

(b) Adjoint Monte Carlo Generated Survival Weights

(c) Adjoint Monte Carlo Relative Error

(d) Adjoint Monte Carlo Reverse Current

Figure 3.13: Forward Variance Reduction, Energy Group 8

3.6 Discussion

With everything considered through examining the survival weight figures and the results in table 3.2, it seems that both stochastic and deterministic methods of generating adjoint results present with their own benefits and drawbacks that don't seem to make a noticeable difference in the FOM results of the simulations themselves. However, the adjoint Monte Carlo results took 36.8 hours to produce while the deterministic method was nearly instantaneous. This presents a conclusion that it is much more efficient to generate scalar weight windows/targets from deterministic methods rather than continuous energy Monte Carlo.

However, what is also clear is that in problems where direction of the particle is especially important, the FOM can be increased by large amounts if angular data is present in the weight windows/targets as is made evident by the results in table 3.2. In light of this, it is concluded that if one generates reliable angular weight windows/targets from a deterministic method, the benefit in problems where high anisotropy in the adjoint data would be immense. It is therefore deemed successfully proven that angular adjoint data is effective when used with importance sampling.

Chapter 4

Adjoint Importance Sampling

Adjoint continuous energy Monte Carlo is a useful tool for modeling transport problems with certain properties. For example, say a problem has a very spatially large forward source region made of material that causes high attenuation with a relatively spatially small region of interest (where a detector is present, for example). In these types of problems when analogue forward transport is utilized, it is foreseeable that a large number of particles would be generated at disadvantageous phase space locations in the source region where a large amount of attenuation is present between it and the region of interest. Other combinations of similar types of phase space discrepancies between the source and detector could also occur with respect to energy or direction space, such that the source energy or direction functional dependence degrades efficiency of the calculation in some way (having a much larger energy range of possible source emission values when compared to a small domain of detector energy response, for example). Of course, one could potentially alleviate these issues by using variance reduction tactics such as importance sampling of the source or implicit capture, as mentioned in the literature review section of this work.

However, there also exists the possibility of utilizing the adjoint for this type of problem. The adjoint would have the benefit of generating samples that start from the detector itself (which, as mentioned earlier, has a relatively small spatial domain in this hypothetical problem) and instead would generate scores for the estimator in the forward source, for which its relatively large spatial domain potentially gives it the potential to converge much more quickly and, in turn, have a higher figure

of merit.[3] FRENSIE was developed with adjoint continuous energy Monte Carlo capabilities in mind, and has its adjoint capabilities for photons benchmarked with the thesis of Alex Robinson. [3]

It's therefore foreseeable that one might want to use the adjoint to model a problem instead of the forward. However, just because the adjoint might be more efficient for certain problems when compared with the forward, it does not mean that exploring variance reduction for the adjoint itself in other problems where it is less efficient is not without merit. It is, after all, those problems where variance reduction should make the biggest impact. The potential to improve the rate of convergence of an adjoint problem of this type is therefore a useful vehicle to explore how one might reliably improve the figure of merit for all adjoint simulations expressed in equation 2.66. There is scant information present in the literature about variance reduction for adjoint Monte Carlo, and no information about any zero variance importance sampling schemes for the adjoint integral equations derived in the work by Hoogenboom and the work by Gabler et. al.[4][5] In fact, the only mention of any variance reduction method for the adjoint found was in the work of Hoogenboom mentioning using the forward equation for photons (and a slightly altered scheme for neutrons, which is irrelevant here).[4] Here, we attempt a variance reduction scheme utilizing the equivalent of CADIS for the adjoint by instead using the forward flux information to generate weight windows and weight targets and explore how utilizing this scheme in a manner that interacts with the effects of the adjoint weight factor $P^{\dagger}$ effects the figure of merit.

4.1 Adjoint Importance Sampling Theory

In chapter 3, the adjoint solution generated from discrete ordinates using Denovo and the adjoint Monte Carlo capabilities of FRENSIE were used to generate an importance function for the forward Monte Carlo simulation and explore the effects that discretizing direction space had on the figure of merit. In this section, we do the reverse - utilize the forward solution from Denovo (a discrete ordinates code that ADVANTG utilizes for CADIS)[21] and FRENSIE in order to generate an importance function for the adjoint in order to explore the effects this has on the figure of merit for the adjoint simulation. The logic behind this is that the forward is the function that is adjoint to the adjoint function, and such an idea has been mentioned in other literature.[4] The same implementation objects (weight windows and weight targets) will be explored, as well as scalar vs. angular weight windows and weight targets. For clarity, the survival weights and weight targets used for the adjoint is as follows:

$$W_s^\dagger(\Delta \boldsymbol{Y}) = W_t^\dagger(\Delta \boldsymbol{Y}) = \frac{F}{\phi(\Delta \boldsymbol{Y})}, \tag{4.1}$$

where ϕ is either generated through Denovo or forward Monte Carlo mesh estimator results. ADVANTG was used again to run Denovo.[21] Angular weight windows and weight targets were once again explored with the adjoint using forward angular results.

The adjoint source also has importance sampling applied to it in a manner equivalent to CADIS as follows:

$$\hat{\eta}_\phi(\boldsymbol{Y}) = \frac{\phi(\Delta \boldsymbol{Y})\eta_\phi(\boldsymbol{Y})}{\int \phi(\Delta \boldsymbol{Y})\eta_\phi(\boldsymbol{Y})d\boldsymbol{Y}'}, \tag{4.2}$$

which makes it such that adjoint pseudoparticles emitted from the adjoint source match the weight window or weight target in that region of phase space.

Using weight windows and weight targets in this manner has the interesting effect of reversing the weight transformations made by the adjoint weight factor. During a typical adjoint simulation without any importance sampling, the particle weight transformations are as follows during a collision:

$$w^{\dagger}(\mathbf{Y}) = P^{\dagger}(\mathbf{Y})w^{\dagger}(\mathbf{Y}'), \tag{4.3}$$

where $P^{\dagger}(\mathbf{Y}) = \frac{\Sigma^{\dagger}(\mathbf{Y})}{\Sigma_t(\mathbf{Y})}$. With importance functions, equation 4.3 is expressed as follows:

$$w^{\dagger}(\mathbf{Y}) = P^{\dagger}(\mathbf{Y})w^{\dagger}(\mathbf{Y}')\frac{I^{\dagger}(\mathbf{Y}')}{I^{\dagger}(\mathbf{Y})} \quad , \tag{4.4}$$

and for this to be true in conjunction with equation 4.1, the following has to be true:

$$I^{\dagger}(\mathbf{Y}_N) = \frac{\phi(\mathbf{Y}_N)}{F}\prod_{i=1}^{N} P^{\dagger}(\mathbf{Y}_i), \tag{4.5}$$

which means the importance function is effectively path dependent. Substituting equation 4.5 into equation 4.4, the following occurs (F is omitted since it cancels out):

$$w^{\dagger}(\mathbf{Y}) = P^{\dagger}(\mathbf{Y})w^{\dagger}(\mathbf{Y}')\frac{\phi(\mathbf{Y}')\prod_{i=1} P^{\dagger}(\mathbf{Y}_1,...,\mathbf{Y}')}{\phi(\mathbf{Y})\prod_{i=1} P^{\dagger}(\mathbf{Y}_1,...,\mathbf{Y})} = w^{\dagger}(\mathbf{Y}')\frac{\phi(\mathbf{Y}')}{\phi(\mathbf{Y})}. \tag{4.6}$$

With equation 4.6 being true, the only weight transformations occurring are now due to importance sampling utilizing the forward flux. As such, the survival weight and weight target values proposed in equation 4.1 are recovered as follows when using importance function in equation 4.5:

$$w^{\dagger}(\mathbf{Y}_N) = \frac{1}{I^{\dagger}(\mathbf{Y}_0)} \left[\prod_{i=1}^{N} P^{\dagger}(\mathbf{Y}_i)\right] \left[\prod_{i=1}^{N} \frac{I^{\dagger}(\mathbf{Y}_{i-1})}{I^{\dagger}(\mathbf{Y}_i)}\right] = \frac{\prod_{i=1}^{N} P^{\dagger}(\mathbf{Y}_i)}{\frac{\phi(\mathbf{Y}_N)}{F} \prod_{i=1}^{N} P^{\dagger}(\mathbf{Y}_i)} = \frac{F}{\phi(\Delta \mathbf{Y})}. \quad (4.7)$$

The hope for utilizing such an importance function is that the combination of reversing the large weight fluctuations from the adjoint weight factor, which can have a value of $[0, \infty)$, and using an importance function proportional to the forward function will result in an increase in the figure of merit and faster convergence. From the results to come, we can clearly see that this is the case.

4.2 Demonstration

The test problem used to showcase this method is the same as in section 3.2. This problem does not match the description of the type of problem described in the initial discussion where the adjoint might be more useful than the forward, as demonstrated by the results in the below tables when compared with that in section 3.5. This is likely largely due to the fact that the adjoint source has a large energy domain when compared with the forward's monoenergetic source. This actually presents a useful example where one might examine the effects of variance reduction on the adjoint calculation. After all, improving the rate of convergence is the primary concern of variance reduction methods, and this problem has much room for improvement when it comes to the rate of convergence.

There is also the fact that the adjoint simulation has the adjoint weight factor which can cause a wide variation in weights. In particular, this problem utilizes

hydrogen, and the adjoint weight factor generally has a higher maximum value in regions where the medium has a lower Z-value, which also makes it an excellent test problem for this method.[3] This wider fluctuation of weights due to the adjoint weight factor can contribute to a lower figure of merit. If there is a higher likelihood of a larger variation of weights, then the previously outlined method can also provide additional benefit in restricting the fluctuations resulting from a larger range of possible adjoint weight factor values. Therefore, this problem presents an ideal case in which to examine the effects of the earlier mentioned method with the addition of angular importance sampling information.

A difference of note for this problem is that it uses an energy point detector in the forward source (adjoint estimator) region due to the forward source being monoenergetic at $0.835MeV$. The energy point detector is described in the work by Alex Robinson.[3], but it essentially works in a very similar manner to the spatial point detector utilized in MCNP,[1] except for energy. It effectively means that adjoint pseudoparticles with a higher energy that produce samples in the forward source region give larger score values for the adjoint estimator, which would presumably translate to particles having a higher energy in the region of interest contributing more to the result of interest and therefore being "more important".

Due to the fact that the forward and adjoint answers should match and that this is the same problem as in chapter 3, the forward answers from chapter 3 can be used as a benchmark for what counts as a reference answer. Immediately, it is clear that the scalar importance sampling implementations give a 10-20 factor increase in the figure of merit regardless of whether Denovo or the Monte Carlo results are utilized from table 4.1 and the scalar importance sampling results displayed in table 4.2. Similarly, the angular results give better results showing a 20-40 factor increase in the figure of merit as shown in the angular results in table 4.2. As for the difference

between weight windows and weight targets, it is still not definitively clear that one is better than the other, nor is it clear that using Monte Carlo results are inherently better than using a discrete ordinates solution for importance sampling from solely examining the resulting figure of merit results produced.

In the result tables 4.1 and 4.2, there seems to be somewhat of a range of mean answers - some being questionable in their acceptability given the means produced in chapter 3. While most are in the rough "ballpark" of the answers presented for the forward, some means are roughly a couple of standard deviations from what might be expected to be a reasonable answer. The number of standard deviations from a reference answer (the forward mean result with no variance reduction applied, in this case) is displayed in table 4.3. D from table 4.3 is expressed as follows:

$$D = \frac{\hat{F}_{ref} - \hat{F}^{\dagger}}{S_{diff}} \quad , \tag{4.8}$$

where S_{diff} is defined as follows through propagation of error:

$$S_{diff} = \sqrt{\left(\hat{F}_{ref} R_{ref}\right)^2 + \left(\hat{F}^{\dagger} R^{\dagger}\right)^2} \quad , \tag{4.9}$$

where $\hat{F}_{ref}$ is the forward analogue solution for the respective estimator type, R_{ref} is the relative error of said forward solution, $\hat{F}^{\dagger}$ is the adjoint result of a given simulation and estimator type, and $R^{\dagger}$ is the adjoint solution's relative error. It is of note from chapter 1 that $\hat{F}R = S_{\hat{F}}$, where $S_{\hat{F}}$ is the standard deviation of mean result $\hat{F}$.

Of particular interest might be that of the analogue track length estimator result in table 4.1 and the results utilizing scalar weight windows formed from a forward stochastic solution in table 4.2, where the rest are easily explainable through their

respective relative errors. This is most likely simply due to the fluctuation of weight due to the adjoint weight factor causing instabilities in the estimator result. Even though the stochastically formed results in table 4.2 for scalar weight windows are roughly a couple of standard deviations outside of a somewhat expected range of results based on the results of chapter 3, they are still better than the analogue result of the track length estimator in table 4.1, which is about 3.5 standard deviations away from an expected range of results. The reason that the results from utilizing stochastically formed weight windows in table 4.2 are off in spite of utilizing the variance reduction method outlined in section 4.1 is likely due to the fact that these variance reduction methods utilized in FRENSIE are only applied after a particle (forward or adjoint) leaves a collision. If they were applied at geometric boundary crossings as they are in MCNP,[1] then it is likely that the range of weights would be restricted and the answers would be even closer to a reasonable range. In spite of this, the scalar weight windows utilizing stochastic results still do present a reasonable improvement in the answers over the analogue. This discrepancy also likely has to do with why the figure of merit results for stochastically formed scalar weight windows in table 4.2 do not present a larger improvement in the figure of merit when compared stochastically formed angular weight windows in table 4.2 when comparing collision estimators despite utilizing angular information (though they clearly do show an improvement in the figure of merit regardless). If one desired to determine the veracity of these likelihoods, one would have to implement a way to measure the distribution of weights that is observed by the estimator itself. FRENSIE would have to be made capable of performing weight window/target checks at geometric boundary crossings. Turning this capability on while measuring the range of weights observed and comparing with the FOM and distance from the reference mean would prove the veracity of these claims. This is left to future work.

However, what is clear is that utilizing the forward function to set weight windows and weight targets provides a clear benefit in the rate of convergence according to the increased figure of merit values. This is all while still approaching the reference answer. It is clear that this method does provide a sizable increase in the figure of merit while preserving unbiasedness in the simulation.

Estimator	*Mean*	*RE*	*FOM*	*VOV*	*Simulation Time* (h)
Collision	1.21842491E-06	0.13207582	0.00035758	0.13076203	44.5
Track Length	9.06490629E-07	0.10417542	0.00057476	0.1183347	44.5

Table 4.1: Adjoint Results (Analogue, 4e9 histories)

VR Input Source	*Estimator*	*Mean*	*RE*	*FOM*	*VOV*	*Simulation Time* (h)	*Speedup*
Scalar Weight Windows, 1e9 histories							
deterministic	Collision	1.25693983E-06	0.07210153	0.00372433	0.39238345	14.3	10.42
	Track Length	1.27419958E-06	0.05259472	0.00699927	0.11444542	14.3	12.18
stochastic	Collision	1.11492552E-06	0.05598032	0.00560026	0.31082981	15.8	15.66
	Track Length	1.15767093E-06	0.0437473	0.00917015	0.19390512	15.8	15.95
Scalar Weight Targets, 1e9 histories							
deterministic	Collision	1.18973103E-06	0.07782158	0.00308933	0.37011389	14.8	8.64
	Track Length	1.22682260E-06	0.04829253	0.00802241	0.14652526	14.8	13.96
stochastic	Collision	1.22365107E-06	0.04751114	0.00746019	0.07167815	16.5	20.86
	Track Length	1.26859048E-06	0.05791548	0.00502055	0.25421606	16.5	8.74
Angular Weight Windows, 1e9 histories							
stochastic	Collision	1.23249271E-06	0.0487874	0.00736807	0.30063666	15.8	20.61
	Track Length	1.24097540E-06	0.03418152	0.0150102	0.17564253	15.8	26.12
Angular Weight Targets, 1e9 histories							
stochastic	Collision	1.20344956E-06	0.0345817	0.01327771	0.1778516	17.5	37.13
	Track Length	1.19880210E-06	0.02724162	0.02139686	0.06853305	17.5	37.23

Table 4.2: Adjoint Results (Importance Sampled)

Adjoint Simulation VR	*Histories*	*Estimator*	*D*
Analogue	4e9	Collision Track Length	-0.074 -3.424
Deterministic Scalar Weight Windows	1e9	Collision Track Length	0.256 0.639
Stochastic Scalar Weight Windows	1e9	Collision Track Length	-1.832 -1.384
Deterministic Scalar Weight Targets	1e9	Collision Track Length	-0.464 -0.059
Stochastic Scalar Weight Targets	1e9	Collision Track Length	-0.161 0.510
Stochastic Angular Weight Windows	1e9	Collision Track Length	-0.014 0.236
Stochastic Angular Weight Windows	1e9	Collision Track Length	-0.666 -0.889

Table 4.3: Adjoint Mean Precision

In addition to these results, we can also explore having an importance function as follows instead of that in equation 4.5. The importance function could be instead set such that it's strictly proportional to the forward flux:

$$I^{\dagger}(\mathbf{Y}) = \frac{F}{\phi(\mathbf{Y})}, \tag{4.10}$$

which would result in the following target weight values:

$$W_t(\mathbf{Y}) = \frac{F}{\phi(\mathbf{Y})} \prod_{i=1}^{N} P^{\dagger}(\mathbf{Y}_i). \tag{4.11}$$

This translates into the weight target values being dependent on the path that an adjoint pseudoparticle has taken. As such, the weight targets are allowed to change value based on the product of a given adjoint pseudoparticle's adjoint weight factor occurring at each collision it has experienced up until a point. The results of such a

simulation are found in table 4.4, and it is immediately clear when comparing these results to that of angular weight targets in table 4.2 that utilizing the importance function as strictly proportional to the forward solution causes a significant decrease in the figure of merit. While the answers in table 4.4 present a higher figure of merit than the analogue, it contains a lower FOM than the angular results in table 4.2, indicating that reversing the effect of the adjoint weight factor can significantly impact the performance of the problem.

VR Input Source	*Estimator*	*Mean*	*RE*	*FOM*	*VOV*	*Simulation Time (h)*	*Speedup*
stochastic	Collision	1.17424407E-06	0.05153616	0.0049819	0.10175382	21.0	13.93
	Track Length	1.24169296E-06	0.03912978	0.00864181	0.07128306	21.0	15.04

Table 4.4: Adjoint Results (Angular Weight Targets with Adjoint Weight Factor Transforms, 1e9 histories)

The following figures present survival weight values (once again, equivalent to the weight target values) utilized for the adjoint Monte Carlo simulation found by utilizing Denovo's forward discrete ordinates capabilities and the forward Monte Carlo results in the importance function mentioned above. First, the images of the survival weights produced using Denovo's forward flux results are presented, followed by the survival weights generated from the forward flux results generated by FRENSIE, and then followed by the forward reverse current providing a visual aid for the direction dependence of the survival weights being used for these results. Cells with a relative error above 0.2 are excluded from these images, as they tend to exhibit quite statistical noise. The results are presented on the xy plane at $z = 0$. For the reverse currents, the colorbar indicates the magnitude of the current vector at that position.

When inspecting the results from Denovo, it is of note that, once again, they are somewhat smoother than the stochastic results as a result of statistical error and

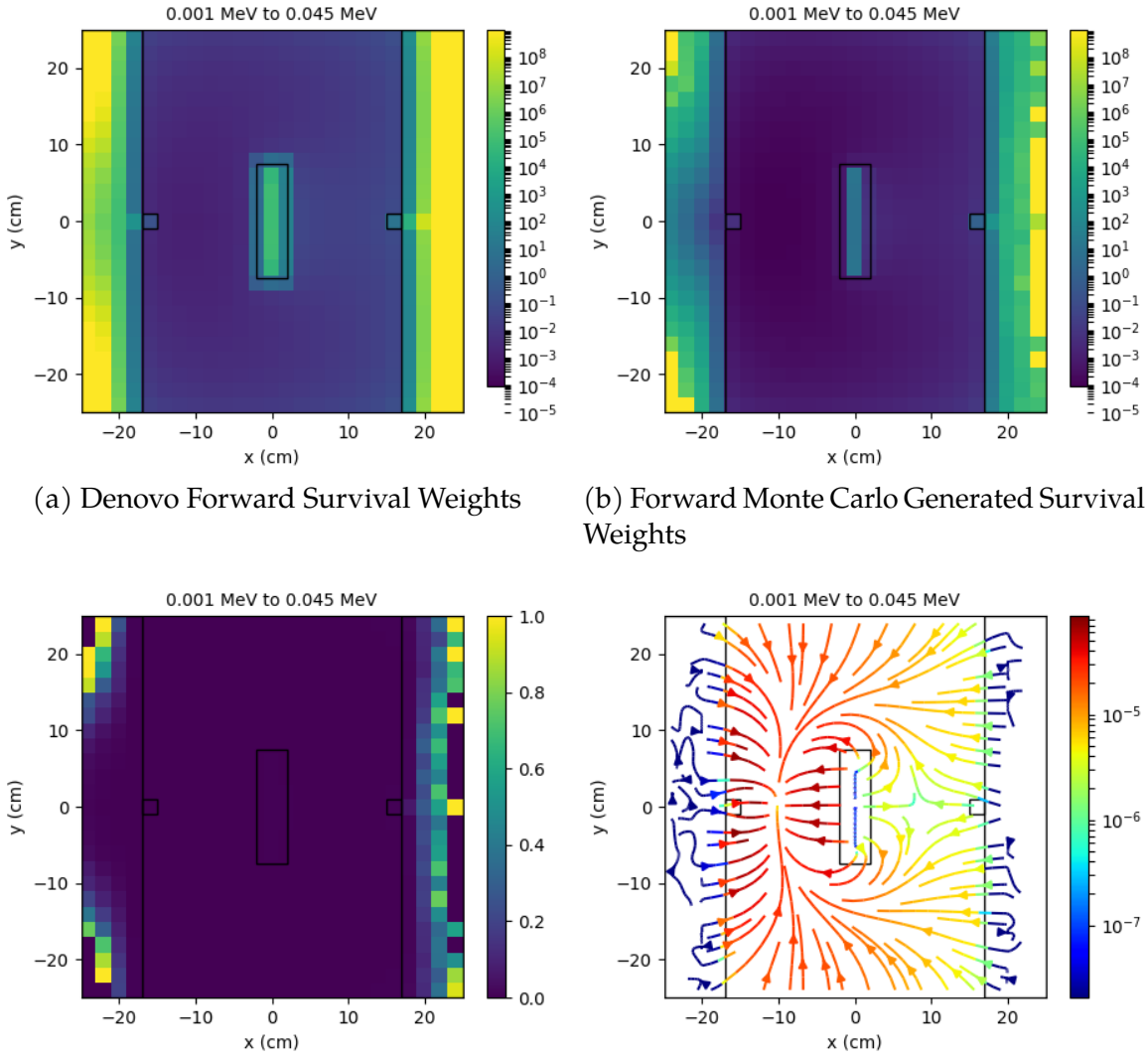
"holes" in the forward estimator data. This is mostly only true in regions of high attenuation though, where one might not expect the adjoint particles to easily reach due to attenuation alone, and so it most likely has negligible effects on the results. The ray effects from discrete ordinates are more prevalent in the forward calculation when compared with that in section 3.5 - especially in higher energy groups as displayed in figures 4.7a and 4.8a. Since these energy groups are arguably more important for the adjoint simulation due to the energy point detector at $0.835MeV$, this might explain some of the FOM improvements in tables 4.2 when Monte Carlo results are utilized for the importance functions, given that the "holes" in the data and lack of smoothness for the Monte Carlo results are much less sparse and in very high attenuation regions. In fact, the error is quite low for most of the problem in other regions, as seen in all of the figures depicting relative error. However, it is ultimately important to note that the Denovo calculation was, once again, almost instantaneous when generating these forward results where the forward Monte Carlo results took about 12.6 hours to generate. Therefore, it seems obvious to conclude that finding a way to generate angular variance reduction information using deterministic results would be more beneficial.

The reverse currents show an increased preference for adjoint pseudoparticles traveling around the shield in the center at lower energies and through the shield at higher energies. This is expected given the lower attenuation for higher energy adjoint photons. Inside areas of high attenuation, there seems to be more noise in lower energy groups, as seen in energy groups 1 through 4. This is not statistical noise though, since regions of high relative error have been omitted from these figures. Instead, an explanation that is more likely to be appropriate would be that it simply does not matter much which direction that an adjoint history in these regions travels. It is of note that forward histories that have reached the relevant energies in

those regions are likely scattering in a variety of directions without much preference given that no forward histories are generated at any other energy than $0.835MeV$, which could explain that noise.

Another note of interest is that it seems the reverse current "source" in energy groups 0 and 1 (figures 3.6d and 3.7d) seem to be slightly in front of the forward source. This likely has an explanation related to that given earlier for the noise in high attenuation regions. It is likely simply the fact that particles emitted from the forward source escape the source region relatively quickly and mostly scatter into the relevant energy regions in those energy groups in that spatial region. Since adjoint pseudoparticles scatter into higher energy groups, it is foreseeable that it might be beneficial for adjoint histories to have a higher importance in front of the source instead of inside of it so that, once it does scatter into a higher energy, it will be more likely to scatter into the source region as evidenced in the other reverse current figures.

Energy Group 1 Figures (0.001 MeV to 0.045 MeV)



(a) Denovo Forward Survival Weights

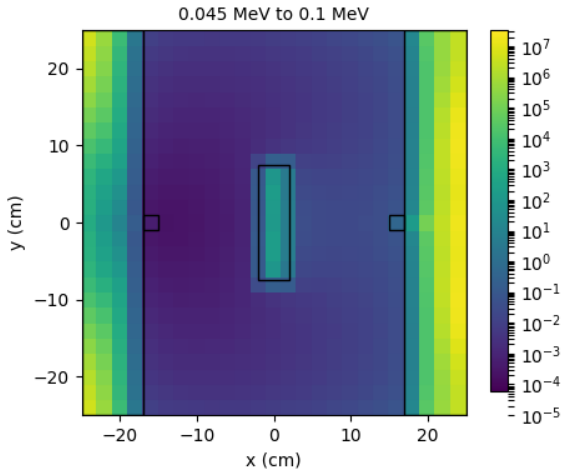
(b) Forward Monte Carlo Generated Survival Weights

(c) Forward Monte Carlo Relative Error

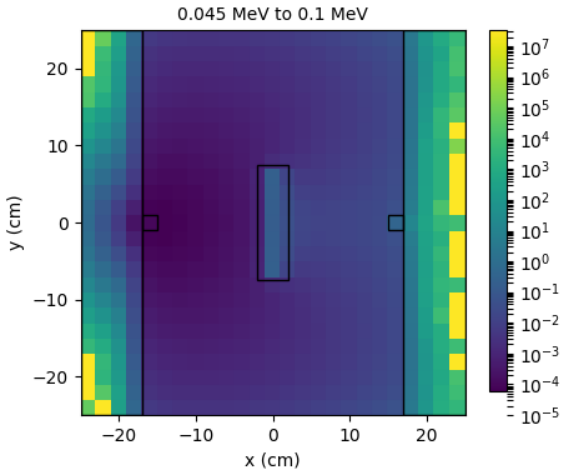
(d) Forward Monte Carlo Reverse Current

Figure 4.1: Adjoint Variance Reduction, Energy Group 1

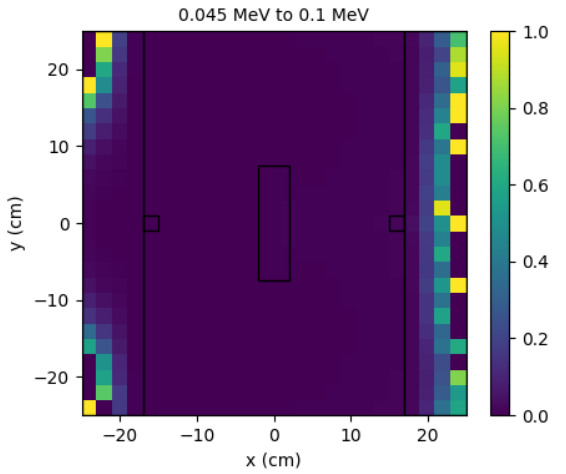
Energy Group 2 Figures (0.045 MeV to 0.1 MeV)



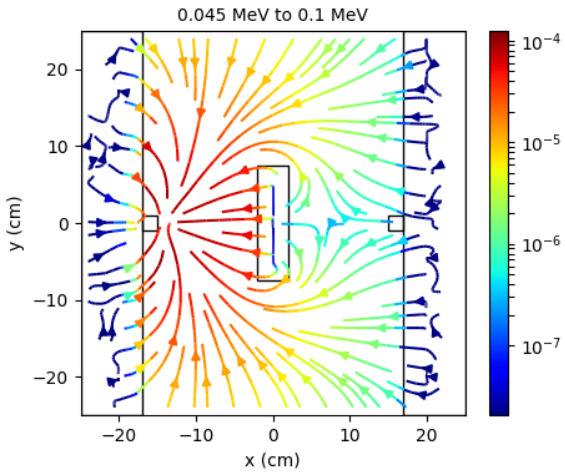
(a) Denovo Forward Survival Weights



(b) Forward Monte Carlo Generated Survival Weights



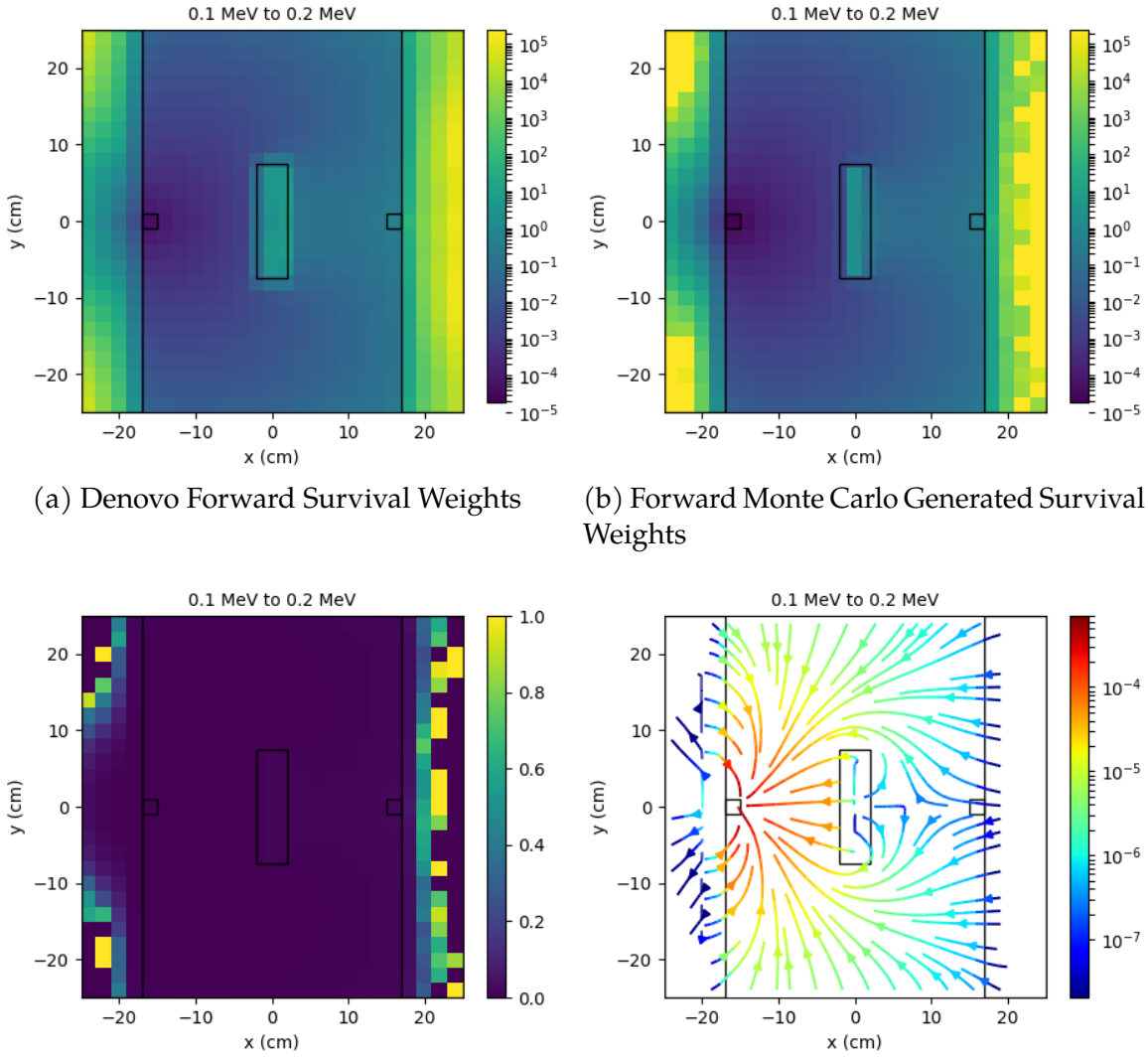
(c) Forward Monte Carlo Relative Error



(d) Forward Monte Carlo Reverse Current

Figure 4.2: Adjoint Variance Reduction, Energy Group 2

Energy Group 3 Figures (0.1 MeV to 0.2 MeV)



(a) Denovo Forward Survival Weights

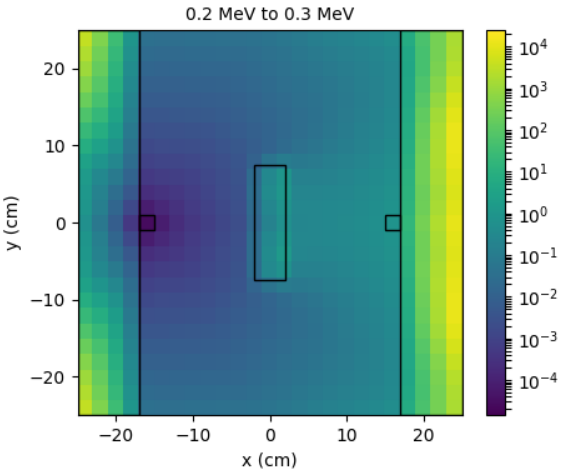
(b) Forward Monte Carlo Generated Survival Weights

(c) Forward Monte Carlo Relative Error

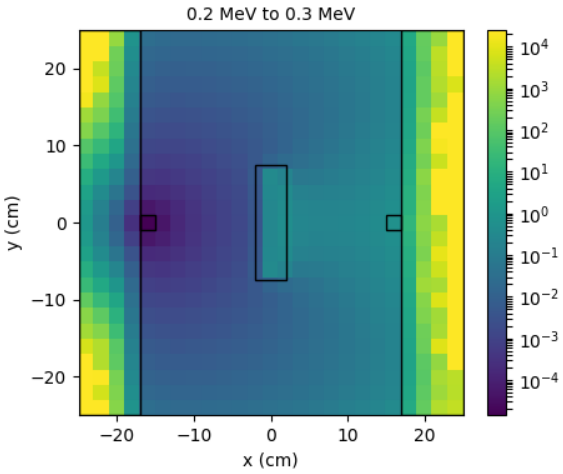
(d) Forward Monte Carlo Reverse Current

Figure 4.3: Adjoint Variance Reduction, Energy Group 3

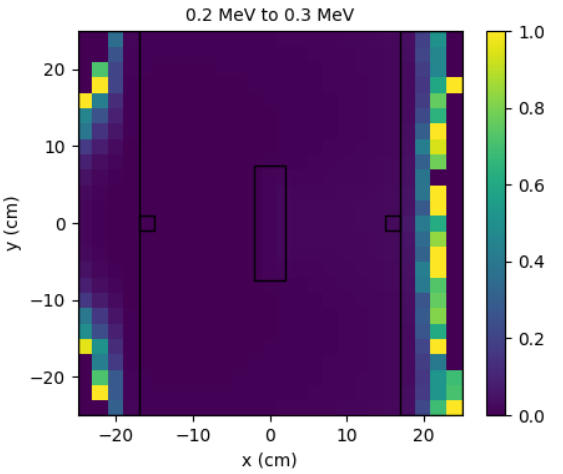
Energy Group 4 Figures (0.2 MeV to 0.3 MeV)



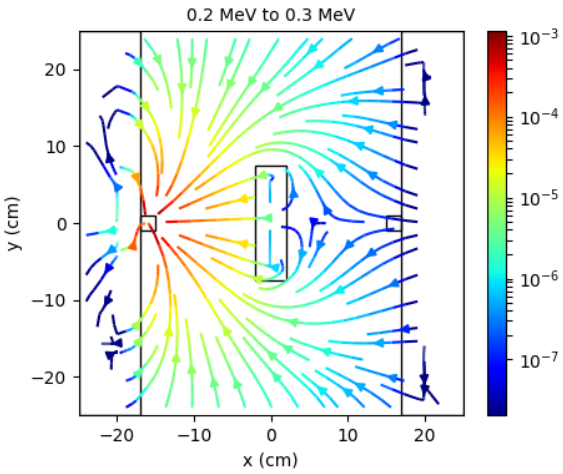
(a) Denovo Forward Survival Weights



(b) Forward Monte Carlo Generated Survival Weights



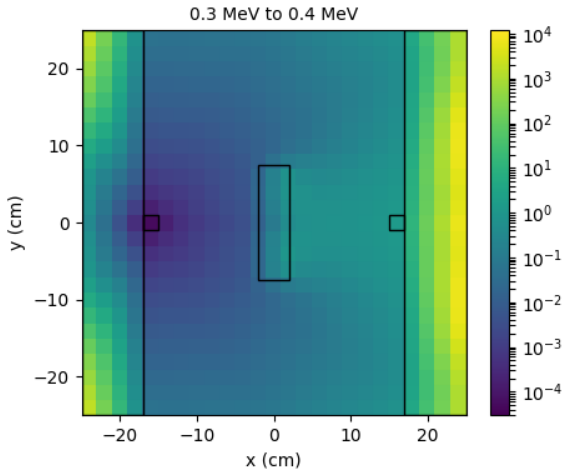
(c) Forward Monte Carlo Relative Error



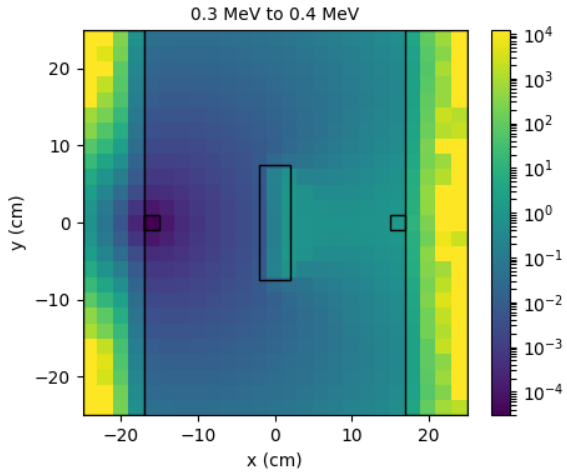
(d) Forward Monte Carlo Reverse Current

Figure 4.4: Adjoint Variance Reduction, Energy Group 4

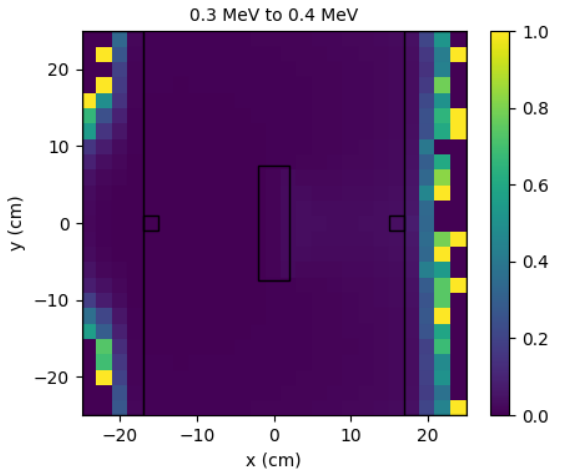
Energy Group 5 Figures (0.3 MeV to 0.4 MeV)



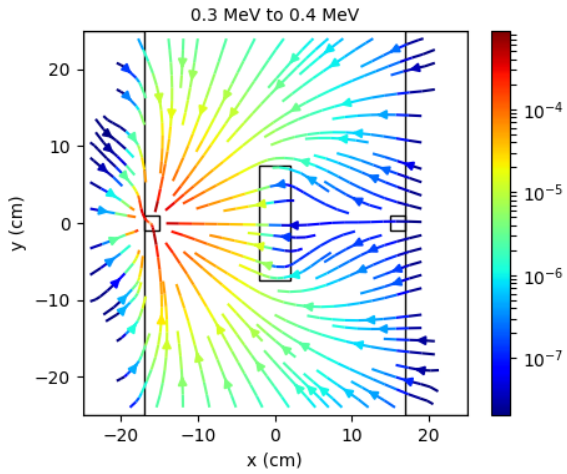
(a) Denovo Forward Survival Weights



(b) Forward Monte Carlo Generated Survival Weights



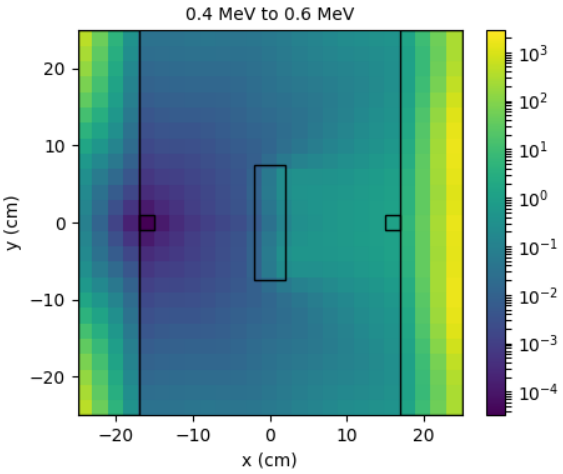
(c) Forward Monte Carlo Relative Error



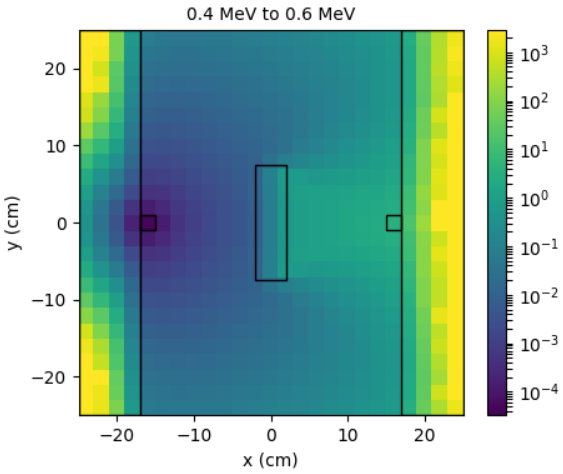
(d) Forward Monte Carlo Reverse Current

Figure 4.5: Adjoint Variance Reduction, Energy Group 5

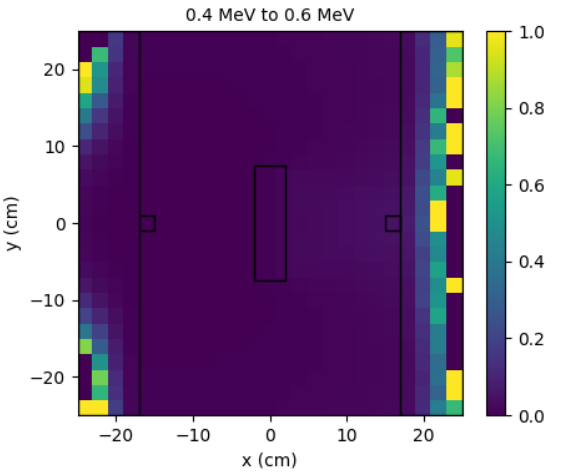
Energy Group 6 Figures (0.4 MeV to 0.6 MeV)



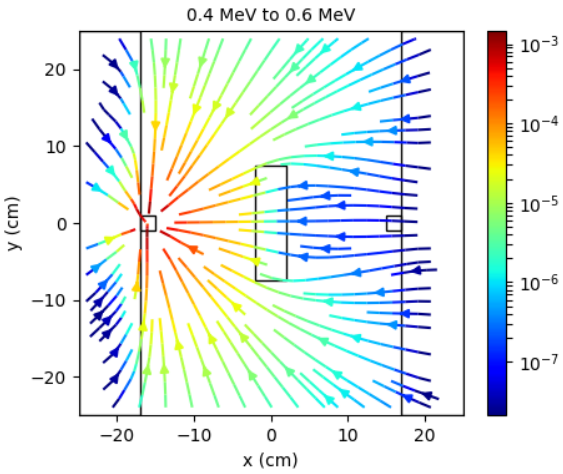
(a) Denovo Forward Survival Weights



(b) Forward Monte Carlo Generated Survival Weights



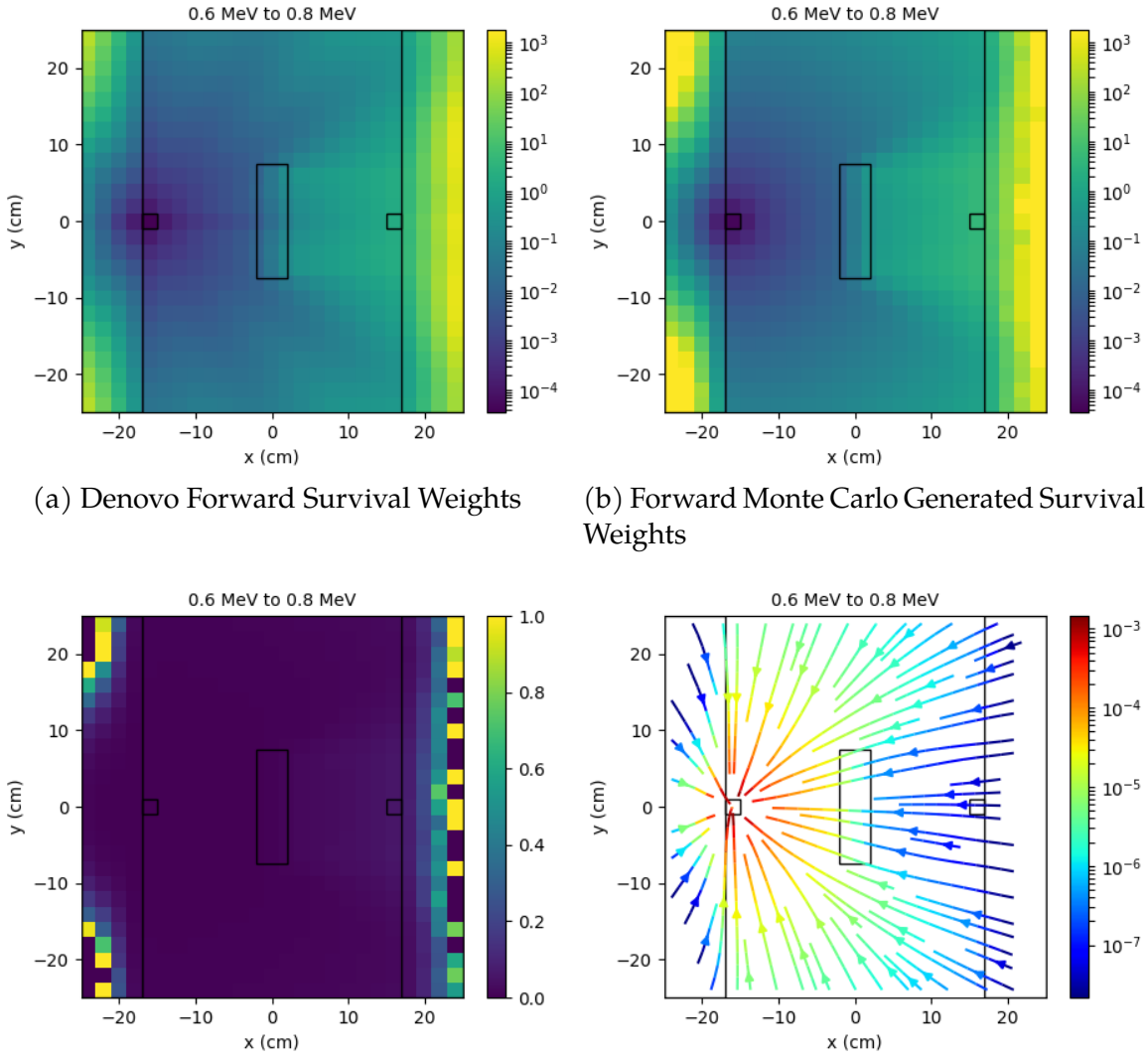
(c) Forward Monte Carlo Relative Error



(d) Forward Monte Carlo Reverse Current

Figure 4.6: Adjoint Variance Reduction, Energy Group 6

Energy Group 7 Figures (0.6 MeV to 0.8 MeV)



(a) Denovo Forward Survival Weights

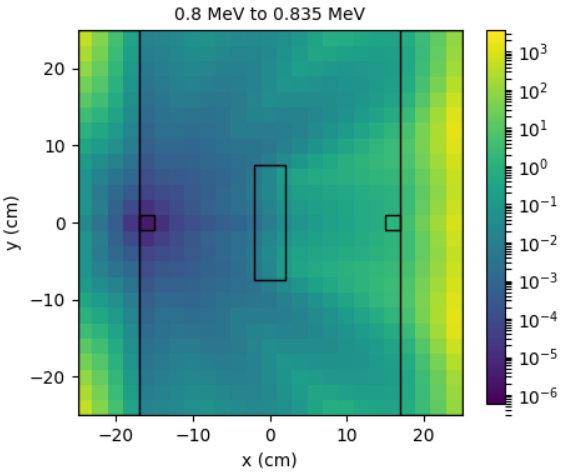
(b) Forward Monte Carlo Generated Survival Weights

(c) Forward Monte Carlo Relative Error

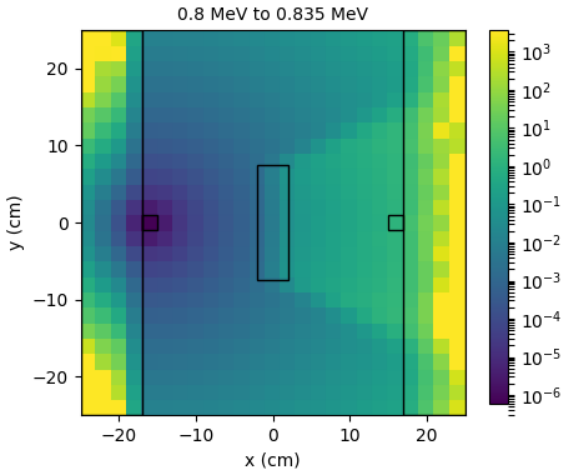
(d) Forward Monte Carlo Reverse Current

Figure 4.7: Adjoint Variance Reduction, Energy Group 7

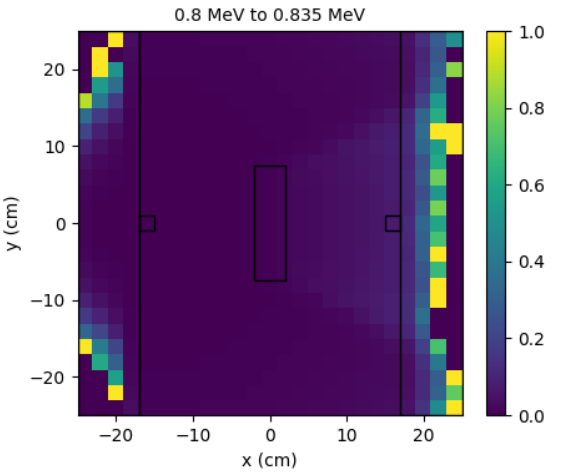
Energy Group 8 Figures (0.8 MeV to 0.835 MeV)



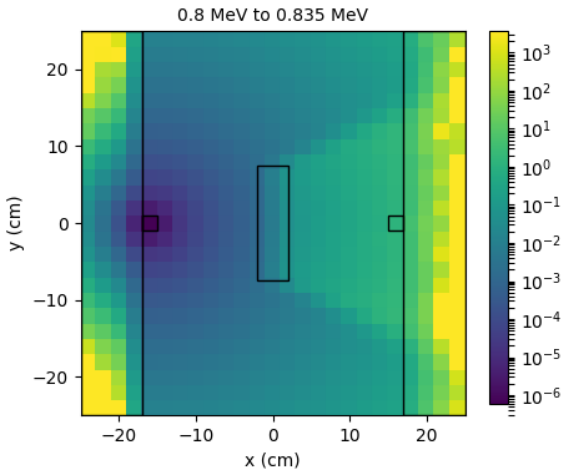
(a) Denovo Forward Survival Weights



(b) Forward Monte Carlo Generated Survival Weights



(c) Forward Monte Carlo Relative Error



(d) Forward Monte Carlo Reverse Current

Figure 4.8: Adjoint Variance Reduction, Energy Group 8

4.3 Discussion

It has been ultimately demonstrated that this method of variance reduction for adjoint Monte Carlo simulations is effective, and can be utilized right now by simply generating forward results from Denovo and producing weight windows from that information. The problem used in the demonstration contains a large amount of relatively high density hydrogen, which can cause large variations in the adjoint pseudoparticle weight due to the adjoint weight factor as explained earlier, and evidenced by table 4.1 for the track length estimator result. In spite of this, the method was able to provide an increase in the figure of merit while producing answers that were closer to the reference answer presented in section 3.5, most of which were within statistical error of this reference answer.

However, once again, it is clear that providing angular information to these weight windows would be greatly beneficial in the convergence rate of problems. If one were able to generate this from deterministic methods, it would be greatly beneficial in other problems too where directional dependence of histories is important (such as needing to go around a shield, as demonstrated in this problem). This has been a definitive example of how using forward information is beneficial to the rate of convergence of adjoint simulations.

Chapter 5

Conclusion

Variance reduction techniques are an important part of Monte Carlo methods for particle transport calculations. They improve the figure of merit and, for some problems with particularly troublesome configurations, provide the ability to actually gain any solution at all within a reasonable time frame. CADIS and other state-of-the-art methods for variance reduction have repeatedly shown large increases in the figure of merit, providing the increased computational efficiency necessary to solve various problems.

The goals of this work included demonstrating the benefit of including direction space in these state-of-the-art importance sampling techniques that are used today and to present a method of implementing importance sampling for adjoint Monte Carlo. In chapter 3, it was shown that for problems where direction is important, a dramatic improvement in the figure of merit can be achieved just by utilizing directional importance sampling. Thus, the benefit has been shown for providing direction information based on angular adjoint information in importance sampling techniques. In addition to this, the discretization provided can be used for estimators which can provide other directional information such as the reverse current visualizations presented at the end of chapter 3 used to provide a visual representation of the importance function being used over direction space.

In addition to this, a method that implements importance sampling for adjoint Monte Carlo was utilized in chapter 4. This method was effectively equivalent to CADIS for adjoint Monte Carlo, where the forward flux was used in place of

the adjoint information in order to provide an importance function for adjoint Monte Carlo. It was shown mathematically that utilizing the forward flux in this manner to set weight windows or target weights was actually reversing the weight transformations caused by the adjoint weight factor. In doing this, the adjoint pseudoparticles were encouraged towards the forward source to provide results and the wide range of weights occurring because of the adjoint weight factor were restricted, thus showing a large increase in the figure of merit in the results in section 4.2. This was also performed using forward flux results from Denovo, the discrete ordinates solver utilized by ADVANTG for the CADIS method, showing that this increase in the figure of merit is also possible utilizing a hybrid method for the adjoint.[21]

Future work would include exploring how to utilize direction information from deterministic solutions to the adjoint equation in order to utilize that information in importance sampling. It is clear that direction based importance sampling can have a large impact on the rate of convergence for a Monte Carlo problem, and utilizing a Monte Carlo simulation to apply importance sampling to another Monte Carlo simulation is not a viable real-world solution to increase the rate of convergence compared to a hybrid method due to the additional time it takes to generate the adjoint Monte Carlo results. Also, an addition to FRENSIE where weight window/-target checks could be performed at geometric boundary crossings along with the ability to measure the range of weights that an estimator observes would make it so that the effects of the spread of the weight distribution could be observed and clearer claims be made on how they affect estimator convergence. However, the goals of this goals of this thesis were to show how direction-based importance sampling and adjoint importance sampling using forward information have been demonstrated and proven, and thus the purpose of this thesis has been completed.

Appendix A

Importance Sampling Utilizing Collision Estimator Zero Variance Theory

Introduction

An original topic of this thesis was also utilizing the zero variance theory of collision estimators to implement importance sampling in order to explore its effects on convergence. In truth, the appropriate zero variance importance sampling scheme is actually dependent on the type of estimator used in a problem.[14][22][11] A small change to the zero variance importance sampling scheme presented in section 2.7 must be made in order to express the zero variance scheme for collision estimators, and is as follows:[14]

$$\tilde{K}(\mathbf{Y}', \mathbf{Y}) = \frac{\hat{K}(\mathbf{Y}', \mathbf{Y})}{\hat{\sigma}(\mathbf{Y}')}, \tag{A.1}$$

where $\hat{\sigma}(\mathbf{Y}')$ is simply the non-absorption probability of a particle at $\mathbf{Y}'$ for the adjoint importance sampled kernel $\hat{K}(\mathbf{Y}', \mathbf{Y})$. This is equivalent to implementing implicit capture for the importance sampled kernel by normalizing the kernel $\hat{K}$ to the non-absorption probability.

This small change induces a zero variance Monte Carlo simulation as shown by Hoogenboom.[14] With an importance function once again set to $\frac{\Phi^{\dagger}(\mathbf{Y}}{F}$, the weight targets are as follows:

$$T_w(\mathbf{Y_N}) = \frac{F}{\phi^\dagger(\mathbf{Y_N})} \prod_{i=1}^{N} \sigma_i(\mathbf{Y}_i). \tag{A.2}$$

The result of this is that the weight targets are changing their values based on the history of the particle up to a given point. With the same adjoint function from chapter 3 utilized as an importance function for $\frac{F}{\phi^\dagger(\mathbf{Y_N})}$, table A.1 contains the results from implementing these weight targets that alter their value based on implicit capture. Evidently, this method results in a drastically worse figure of merit than that presented in chapter 3, likely due to the fact that this method now favors splitting since σ has a range of $[0, 1)$. A conclusion of this is that while zero variance theory provides a theory for how a single history can provide the exact answer to a Monte Carlo problem, it does not necessarily state how long that history would take to complete, nor does it say how long histories utilized in an approximation to it would take vs. the theory for another type of estimator. Future work would be to utilize this for neutrons and adjoint neutrons. However, adjoint neutrons are not yet complete in FRENSIE, and hence this could not be tested.

VR Input Source	*Estimator*	*Mean*	*RE*	*FOM*	*VOV*	*Simulation Time (h)*
stochastic (forward Monte Carlo)	Collision	1.20269274E-06	0.00226012	2.42031345	0.02707566	22.5
	Track Length	1.20045800E-06	0.00185871	3.57859209	0.05816048	22.5

Table A.1: Forward Results (Angular Weight Target with Implicit Capture Transforms, 1e9 histories)

Bibliography

[1] X-5 Monte Carlo Team, "MCNP - A General Monte Carlo N-Particle Transport Code, Version 5, Volume 1: Overview and Theory," Apr. 2003.

[2] J. Shao, *Mathematical statistics*, 2nd ed., ser. Springer Texts in Statistics. Springer, 2003.

[3] Alex P. Robinson, "Development and Implementation of Continuous Energy Adjoint Monte Carlo Methods for Photons." Ph.D. dissertation, UNIVERSITY OF WISCONSIN–MADISON, Jul. 2019.

[4] J. E. Hoogenboom, "Methodology of Continuous-Energy Adjoint Monte Carlo for Neutron, Photon, and Coupled Neutron-Photon Transport," *Nuclear Science and Engineering*, vol. 143, pp. 99–120, 2003.

[5] D. Gabler, J. Henniger, and U. Reichelt, "AMOS – An effective tool for adjoint Monte Carlo photon transport," *Nuclear Instruments and Methods in Physics Research Section B: Beam Interactions with Materials and Atoms*, vol. 251, no. 2, pp. 326–332, Oct. 2006.

[6] M. H. Kalos and G. Goertzel, "Monte Carlo Methods in Transport Problems," *Progress in Nuclear Energy*, vol. 2, pp. 315–369, 1958.

[7] R. R. Coveyou, V. R. Cain, and K. J. Yost, "Adjoint and Importance in Monte Carlo Application," *Nuclear Science and Engineering*, vol. 27, no. 2, pp. 219–234, Feb. 1967.

[8] J. C. Wagner and A. Haghighat, "Automated Variance Reduction of Monte Carlo Shielding Calculations Using the Discrete Ordinates Adjoint Function," *Nuclear Science and Engineering*, vol. 128, no. 2, pp. 186–208, Feb. 1998.

[9] S. A. Turner and E. W. Larsen, "Automatic Variance Reduction for Three-Dimensional Monte Carlo Simulations by the Local Importance Function Transform—I: Analysis," *Nuclear Science and Engineering*, vol. 127, no. 1, pp. 22–35, Sep. 1997.

[10] M. Nowak, D. Mancusi, D. Sciannandrone, E. Masiello, H. Louvin, and E. Dumonteil, "Accelerating Monte Carlo Shielding Calculations in TRIPOLI-4 with a Deterministic Adjoint Flux," *Nuclear Science and Engineering*, vol. 193, no. 9, pp. 966–981, Sep. 2019.

[11] S. R. Dwivedi, "Zero Variance Biasing Schemes for Monte Carlo Calculations of Neutron and Radiation Transport Problems," *Nuclear Science and Engineering*, vol. 80, no. 1, pp. 172–178, Jan. 1982.

[12] J. Spanier and E. M. Gelbard, *Monte Carlo Principles and Neutron Transport Problems*. Addison-Wesley, 1969.

[13] I. Lux and L. Koblinger, *Monte Carlo Particle Transport Methods: Neutron and Photon Calculations*. CRC Press, 1991.

[14] J. E. Hoogenboom, "Optimum Biasing of Integral Equations in Monte Carlo Calculations," *Nuclear Science and Engineering*, vol. 70, no. 2, pp. 210–212, May 1979.

[15] M. Munk, "FW/CADIS-Ω: An Angle-Informed Hybrid Method for Neutron Transport," Ph.D. dissertation, UC Berkeley, 2017.

[16] S. A. Turner and E. W. Larsen, "Automatic Variance Reduction for Three-Dimensional Monte Carlo Simulations by the Local Importance Function

Transform—II: Numerical Results," *Nuclear Science and Engineering*, vol. 127, no. 1, pp. 36–53, Sep. 1997.

[17] E. Brun, F. Damian, E. Dumonteil, F.-X. Hugot, Y.-K. Lee, F. Malvagi, A. Mazzolo, O. Petit, J.-C. Trama, T. Visonneau, and A. Zoia, "TRIPOLI-4R Version 8 User Guide," CEA Saclay, Tech. Rep. CEA-R–6316, 2013.

[18] S. A. Rukolaine and V. S. Yuferev, "Discrete ordinates quadrature schemes based on the angular interpolation of radiation intensity," *Journal of Quantitative Spectroscopy and Radiative Transfer*, vol. 69, no. 3, pp. 257–275, May 2001.

[19] Lloyd Trefethen and David Bau, *Numerical Linear Algebra*. Society for Industrial and Applied Mathematics, 1997.

[20] J. Arvo, "Stratified sampling of spherical triangles," in *SIGGRAPH*, 1995.

[21] S. W. Mosher, S. R. Johnson, A. M. Bevill, A. M. Ibrahim, C. R. Daily, T. M. Evans, J. C. Wagner, J. O. Johnson, and R. E. Grove, "ADVANTG—AN AUTOMATED VARIANCE REDUCTION PARAMETER GENERATOR."

[22] H. C. Gupta, "A Class of Zero-Variance Biasing Schemes for Monte Carlo Reaction Rate Estimators," *Nuclear Science and Engineering*, vol. 83, no. 2, pp. 187–197, Feb. 1983.