

# NE155 Project Final Report

Milos Atz

May 10, 2016

## 1 Introduction

Monte Carlo (MC) methods offer an alternative to deterministic methods of solving numerical problems. Rather than directly solving discretized equations, MC methods utilize repeated random sampling to obtain statistically likely solutions. Given enough repetitions, the MC result will converge to the true value.

This project aims to design and implement an MC method to solve for the neutron flux in a two-region slab. Vacuum boundary conditions are assumed. Constant energy and material properties are assumed; scattering is assumed to be isotropic. The code will use a collision tally mesh to resolve the spatial flux shape. In addition, the relative error will be determined for each mesh space. Variance reduction methods are not implemented in this code.

## 2 Mathematics

Monte Carlo simulations rely on the probabilistic interpretation of physical phenomena by treating them as random variables. Every random variable has a probability density function and a cumulative distribution function. The probability density function (PDF) describes the probability  $\in [0, 1]$  that some value of the random variable will be achieved. The PDF,  $p(x)$ , for a continuous random variable has the following characteristic.

$$p(a \leq x \leq b) = \int_a^b p(x)dx \quad (1)$$

The probability of a variable attaining some value is always greater than 0. Integrating the PDF from  $-\infty$  to  $\infty$  accounts for all possible values and thus should return a value of 1. The integral of the PDF is the cumulative distribution function (CDF),  $F(x) = P(X \leq x)$ , which describes the probability that a continuous random variable  $X$  will be less than or equal to the value  $x$ .

$$P(X \leq x) = F(x) = \int_{-\infty}^x p(x)dx \quad (2)$$

The above equations are for continuous random variables; for discrete random variables, the integrals can be replaced with summations. Individual values for physical phenomena can be resolved by repeated random sampling according to the CDF. Based on a uniformly distributed random variable  $\xi \in [0, 1]$ , values for physical phenomena can be found using the associated CDF:  $F(x) = \xi$ . The challenge involves identifying the PDF/CDF for a given physical phenomena and finding  $F^{-1}$  such that  $x = F^{-1}(\xi)$ .

There are many ways to sample random variables. Some simple methods include direct discrete sampling (i.e. to determine which reaction will take place), direct continuous sampling (determine the number of mean free paths), and rejection sampling (for non-invertible CDF). In this MC solver, as in all MC transport problems, each particle is followed from birth to its death. Through the particle's life, interactions and outcomes are randomly sampled from probability distributions using transport data. In this project, sampling is used to determine values and events. The sampling of these values is described in more detail in the next section.

1. The direction of a particle when it is born or scatters.
2. The type of collision that might occur, either absorption or scattering.
3. The path length from the current particle location to it's next event.

Each particle that is simulated creates a history of interactions throughout the problem space. For this problem, each interaction is a collision between a neutron and a nucleus in the medium. An estimator converts each history into a score,  $x_i$ . As more particles are followed, their scores accumulate into a tally,  $\{x_i\}$ . The expected value of this set is determined by the Strong Law of Large Numbers, which states that as the number of histories,  $N$ , tends toward infinity,  $\bar{x}$  tends to  $E(x)$  [1].

$$\bar{x} = E(x) = \frac{1}{N} \sum_{i=1}^N x_i \quad (3)$$

The scores themselves are random variables, and so the tally has variance defined by the following. The variance is the square of the standard deviation.

$$S_x^2 = \frac{1}{N-1} \sum_{i=1}^N (x_i - \bar{x})^2 \quad (4)$$

Tallies are used to model physical random variables of interest, for which the PDF is unknown and thus cannot directly sampled. In this problem, the physical variable of interest is the neutron flux,  $\phi$ , which is estimated by sampling score  $x$ . Each history,  $i$ , is a sample. As a sufficiently large number of histories are accumulated, the expected value of the tally will converge to the true value of the underlying PDF. In other words, the Monte Carlo attempts to relate  $\bar{x}$  to the physical quantity of interest.

$$\overline{\phi_L} = \frac{1}{L} \frac{1}{N} \sum_i x_i = \frac{\bar{x}}{L} \quad (5)$$

Here,  $L$  is the length of the space over which the flux is being determined. Determining the precision of the Monte Carlo result relies on the Central Limit Theorem: for large values of  $N$  and independent and identically distributed (IID) random variables with finite expected values and variances, the distribution of those random variables approaches a normal distribution. This allows the definition of confidence intervals for  $\bar{x}$  based on its estimated standard deviation. The estimated variance of  $\bar{x}$  is shown below [1].

$$S_x^2 = \frac{S_x^2}{N} \quad (6)$$

Thus, as the number of histories ( $N$ ) is increased, the estimated standard deviation decreases. Another measure of result precision is relative error. The relative error is the estimated standard deviation of the mean relative to the mean, and is given by the following.

$$R = \frac{S_{\bar{x}}}{\bar{x}} \quad (7)$$

It should be noted that no statistical indicator of a Monte Carlo result can tell about the accuracy of the solution; statistics only indicate precision. Accuracy is influenced by the correctness of the physical and mathematical used in the problem, errors in the code, or errors in the problem specification. For most Monte Carlo simulations, the accuracy is unknown and can only be validated by comparison with experiment.

### 3 Algorithms

The algorithm for this code can be described by the flowsheet in Figure 1. To sample for direction, generate a random number  $\xi$  on the uniform distribution  $[0, 1)$ . The direction is positive (right) if  $\xi \geq 0.5$ . The direction is negative (left) if  $\xi < 0.5$ . When direction is sampled, the cosine of the angle is also sampled. Scattering is assumed to be isotropic, so any angle  $\in [0, 2\pi]$  is possible. Because the value of interest is the cosine of the angle,  $[\pi, 2\pi]$  return the same values as  $[0, \pi]$ . In addition, because sampling for direction already eliminates half of the potential angles from consideration, the cosine of the angle is simply identified by generating a random number  $\xi$  on the uniform distribution  $[0, 1)$ . This, multiplied by the direction and the path length, gives the new location (and potentially the collision site) of the particle.

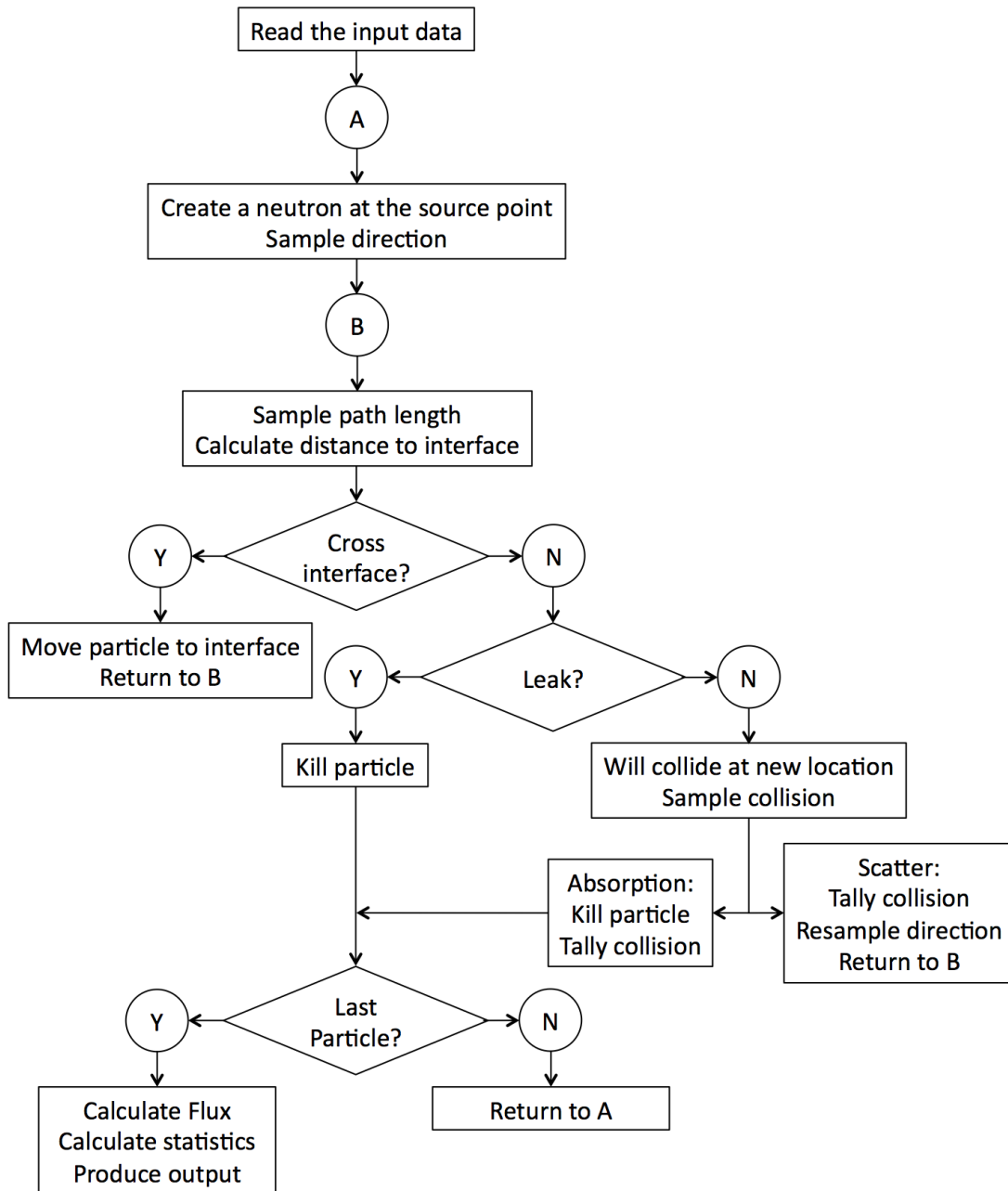


Figure 1: Flowsheet describing the algorithm for the Monte Carlo routine carried out in this code.

The path length is sampled for the region the neutron is currently in (or, if it is crossing the interface between the two regions, for the region the neutron is entering) [3].

$$F_s(s) = \Sigma_t e^{-\Sigma_t s} \quad (8)$$

$$s = -\ln(\xi)/\Sigma_t \quad (9)$$

If the particle is moving toward the interface, the distance between the particle and the interface is measured to determine whether the particle changes region. If the distance between the particle and the interface is less than the sampled path length, the particle will enter the other region. The particle is moved to the boundary and the path length is resampled for the region the particle is entering; the direction and angle remain the same.

If the particle is moving toward a boundary, the sampled path length is compared to the distance between the particle location and the boundary. If the sampled path length is greater than the distance to the boundary, the particle leaks and is terminated.

If the particle does not change regions and does not leak, it will collide within the region it is in. The particle is moved to its new location and the type of collision is sampled. Two types of collision are possible, scattering and absorption; these are characterized by their cross sections,  $\Sigma_s$  and  $\Sigma_a$ , respectively. The total cross section is defined as  $\Sigma_t = \Sigma_s + \Sigma_a$ . Thus, the probability of a reaction  $i$  is given by the following [3].

$$p_i = \frac{\Sigma_i}{\Sigma_t} \quad (10)$$

Each reaction type has a discrete probability, which can be sampled directly by generating a random number  $\xi$  on the uniform distribution  $[0, 1)$  and determining  $k$  such that  $F_{k-1} \leq \xi \leq F_k$ . The reaction  $i_k$  is the one that occurs. If the particle is absorbed, it is terminated. If the particle scatters, direction and angle are resampled and the process is repeated.

For every history, a tally vector is generated to track the scores. The entries in the vector correspond to the spatial bins specified in the input (the user inputs problem bounds and the width of the spatial bins). Every time a collision occurs, a score is added to the tally corresponding to the spatial bin in which the collision took place. Each score has the same weight. After each history, the scores in each bin are added to a cumulative tally. Once all the histories have been completed, this cumulative tally is used to calculate the  $\bar{\phi}$ . In addition, it is used to determine the sample variance. For this, the sum of the squared scores is also required. Thus, for each history, the square of each score in the tally is added to a cumulative total. After all histories are completed, this is equal to the sum of the squared scores and can be used to calculate the sample variance.

The flux result given needs to be scaled by source strength, which is input by the user. Once all histories are complete, the flux and statistics are evaluated for each spatial bin.

## 4 Code Use

The code is executed by running the python script "run" from the Terminal. All of the necessary python subscripts are included in the same directory and are imported and called in when needed. The user controls many aspects of the problem by adjusting the input values at the top of "run.py". The user specifies coordinates of the left and right bound of the slab, the location of the interface, the strength of the source, and the width of the tally mesh bins. In addition, the user can modify the scattering and absorption cross sections in each region.

One important parameter controlled by the user is the number of particles (i.e. histories) to be run in the problem. The appropriate number of particles depends on the required statistical precision, which in turn depends on the width of the tally mesh bins. Running more histories will always yield a more precise result but requires increasingly more time.

Running the code results in the generation of a scatter plot showing the flux at the

center of each tally mesh bin. Error bars are included on the points, showing the relative error of the estimated mean. If the error is sufficiently small, the error bars will not clear the points but are still overlaid on the points. The maximum relative error for the problem is printed in the Terminal, giving the user a bound for the precision of the result.

## 5 Testing and Numerical Results

As previously mentioned, it can be very difficult to check the accuracy of results from Monte Carlo methods. However, for this simple problem, the problem can be solved using diffusion theory. Comparing the diffusion result for flux against the Monte Carlo result can yield insight into the quality of the simulation. Equation 11 shows the one-speed diffusion equation in one dimension (x) [2].

$$\frac{1}{v} \frac{\partial \phi}{\partial t} - \nabla D(x) \nabla \phi(x, t) + \Sigma_a(x) \phi(x, t) = S(x, t) \quad (11)$$

To solve this equation, it is assumed that material properties are homogeneous and that the flux does not change with time (steady state). The source is considered as a plane source that occurs at some value  $x_s$ . The solution is obtained by solving around the location of the source, which splits the slab into two regions. The flux in each region is solved separately. The boundary condition at the plane source is defined in terms of the current given by Equation 12.

$$\lim_{x \rightarrow x_s} J(x) = \frac{S}{2} \quad (12)$$

This represents a special case of the general interface boundary condition based on assuming symmetry in material properties and geometry across the interface. The boundary condition says that the current on either side of the origin must be half of the source strength. Vacuum boundary conditions are utilized at the boundaries on either side of the slab. The vacuum boundary condition is given by Equation 13.



$$\phi(a_{ext}) = 0 \quad (13)$$

Here,  $a_{ext}$  represents the extrapolated distance for whichever side of the slab, right or left, on which the boundary is being considered. Applying these boundary conditions, the flux in the slab defined by left and right boundaries  $a$  and  $b$ , respectively, with  $x_s = 0$  is found to be defined by Equations 14 and 15.

$$\phi(x) = \frac{SL}{2D} \left( 1 + e^{-2\frac{b_{ext}}{L}} \right)^{-1} \left( e^{\frac{-x}{L}} - e^{-2\frac{b_{ext}}{L}} e^{\frac{x}{L}} \right) \quad 0 < x \leq b_{ext} \quad (14)$$

$$\phi(x) = \frac{SL}{2D} \left( 1 + e^{-2\frac{a_{ext}}{L}} \right)^{-1} \left( e^{\frac{-x}{L}} - e^{-2\frac{a_{ext}}{L}} e^{\frac{x}{L}} \right) \quad a_{ext} \leq x < 0 \quad (15)$$

Additionally, if it is assumed that back scattering across the plane source is negligible, the location of the plane source can be used to define the interface between two regions. The Monte Carlo solution considers a two region slab with variable placement of source; this diffusion solution makes nearly perfect agreement between the problems. The only difference is that for the diffusion solution, the interface between the regions is fixed at the location of the plane source. In the Monte Carlo solution, the locations of the interface and source are independent. Equations 14 and 15 can be modified to allow for variable placement of the location of the source and interface, as demonstrated by Equations 16 and 17, respectively.

$$\phi(x) = \frac{SL}{2D} \left( 1 + e^{-2\frac{(b_{ext}-x_s)}{L}} \right)^{-1} \left( e^{-\frac{(x-x_s)}{L}} - e^{-2\frac{(b_{ext}-x_s)}{L}} e^{\frac{(x-x_s)}{L}} \right) \quad 0 < x \leq b_{ext} \quad (16)$$

$$\phi(x) = \frac{SL}{2D} \left( 1 + e^{-2\frac{(a_{ext}-x_s)}{L}} \right)^{-1} \left( e^{-\frac{(x-x_s)}{L}} - e^{-2\frac{(a_{ext}-x_s)}{L}} e^{\frac{(x-x_s)}{L}} \right) \quad 0 < x \leq b_{ext} \quad (17)$$

When specifying the input, the user chooses whether the analytical solution should be included. If it is included, the interface is forced to be at the location of the source.

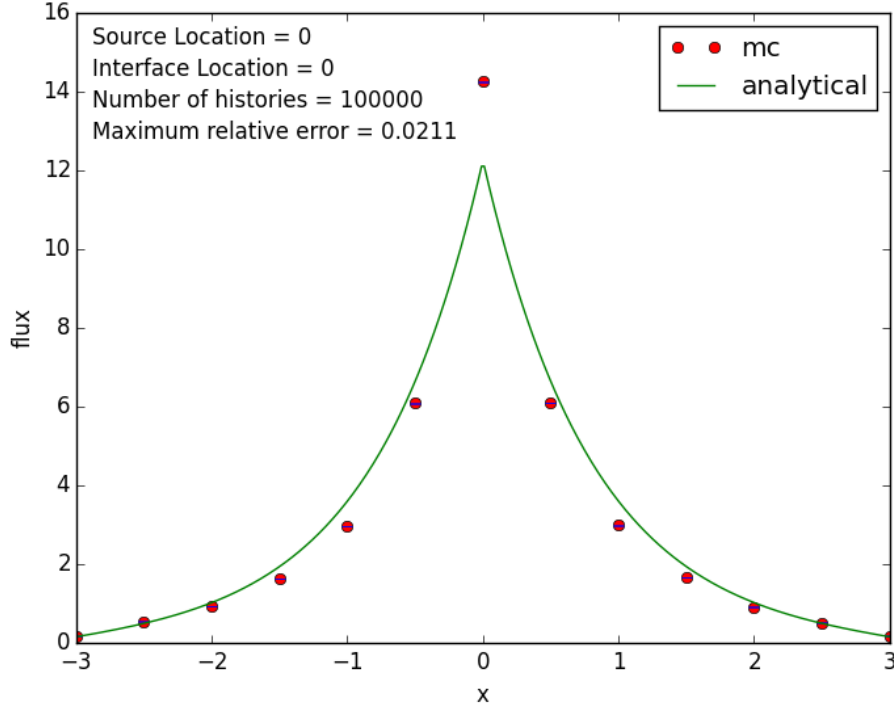


Figure 2: Example of code output comparing the Monte Carlo and analytical diffusion solutions for flux in the system. Here,  $x_s = 0$  and the material properties are the same on either side of the source plane.

Together, the Monte Carlo and analytical solutions give results like those demonstrated in Figures 2 and 3.

The different cross-sections on either side of the source plan in 3 lead to the discontinuity in the analytical solution at that point. Though not rigorously correct, it gives an approximation of the solution and serves to validate the Monte Carlo result. If the analytical result is not included, the interface can be located anywhere, generating flux profiles like that in Figure 4.

In Figure 4,  $\Sigma_{t,1} = 1.0$  and  $\Sigma_{s,1} = 0.9$ , while  $\Sigma_{t,2} = 3.0$  and  $\Sigma_{s,2} = 0.9$ . This results in the flux increase seen at the interface location. This can be understood by considering

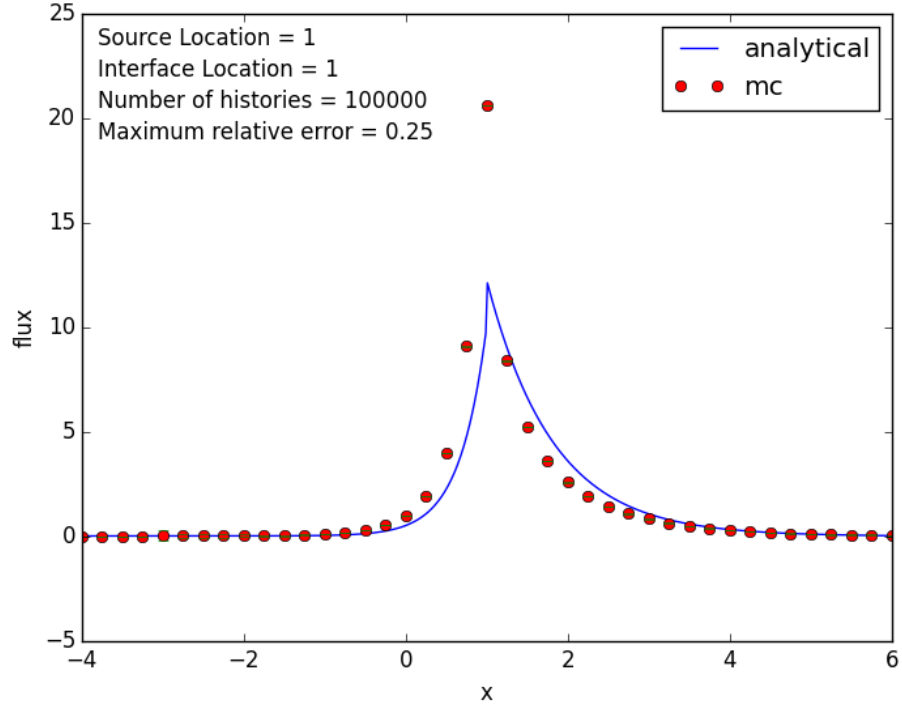


Figure 3: Example of code output comparing the Monte Carlo and analytical diffusion solutions for flux in the system. Here,  $x_s = 1$  and the cross-sections are different on either side of the source plane.

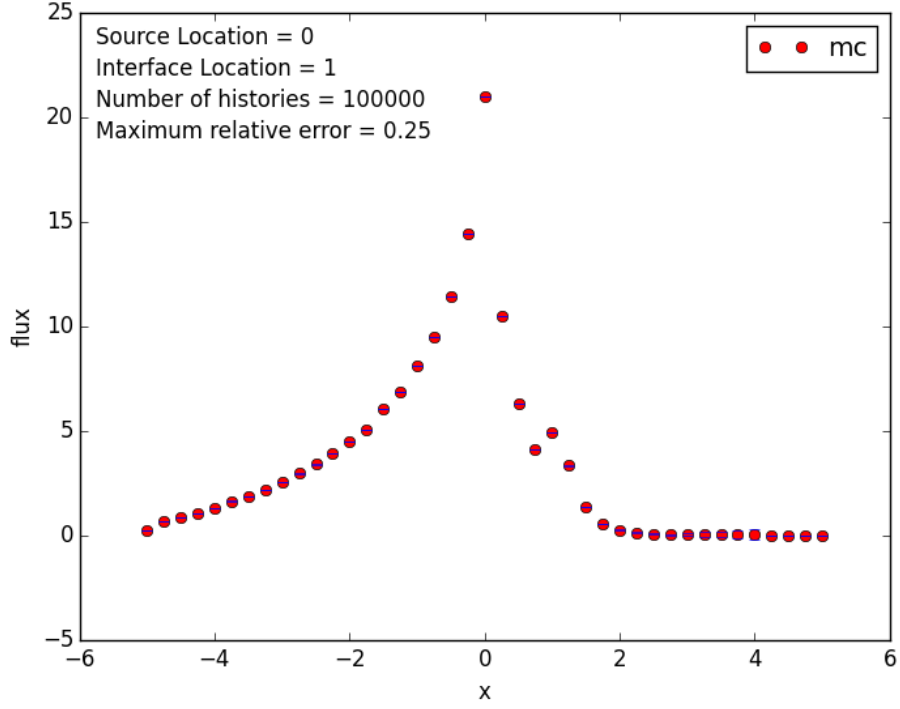


Figure 4: Example of code output showing Monte Carlo output with the different locations for the source ( $x = 0$ ) and the interface ( $x = 1$ ).  $\Sigma_{t,1} = 1.0$ ,  $\Sigma_{s,1} = 0.9$ ;  $\Sigma_{t,2} = 3.0$ ,  $\Sigma_{s,2} = 0.9$ .

conceptually how the flux is influenced by the cross-sections. The total cross-section to the right of the interface is higher, meaning that neutrons are more likely to interact in general. Because the scattering cross-sections are equal, all of the increase in total cross-section contributes to an increase in the absorption cross-section. The probability for absorption is higher to the right of the interface, so neutrons traveling from left to right across the interface are much more likely to get absorbed than scatter. In other words, not only are particles crossing the interface from left to right more likely to interact in general but they are much more likely to be absorbed. Many particles likely pass through the bins immediately to the left of the interface and are then absorbed to the right of the interface, leaving those bins with a lower collision tally.

The statistical quality of the result can be altered by careful problem setup. Increasing the number of histories decreases the relative error and standard deviation. The relative error can also be decreased by making the spatial bins more coarse. If there are less spatial bins, more particles will fall into each, improving the statistics in each bin. The standard deviation is largest where the flux is highest; conversely, because the relative error is equal to the standard deviation divided by the mean flux, the relative error is largest near the boundaries where the flux is near zero. This indicates the significance of the relative error. Because more particles exist in the high flux regions, the result should be best resolved where the flux is highest, leading to the lowest relative error.

For this simple problem, it does not take many particles to arrive at a solution with acceptable statistics; the analog Monte Carlo is satisfactory. This works well when most of the particles contribute to the tallies. When this is not the case, the statistical uncertainty of the answer can be enormous. To avoid this, non-analog Monte Carlo is used, which attempts to follow interesting particles more than non-interesting ones, therefore increasing the chances that more particles will contribute to the tally. To avoid artificial inflation of the tally results, the contribution of each particle to the tally is modified to remove the effect of this biasing. There are many ways to do this, referred to generally as variance

reduction [1]. Though these methods are not included in this code, many can be readily implemented in future versions.

## References

- [1] *MCNP5 - A General Monte Carlo N-Particle Transport Code, Version 5*. Volume 1: Overview and Theory. X-5 Monte Carlo Team, Los Alamos National Laboratory (rev. 2008).
- [2] R. Slaybaugh. *UCBNE250 Class Notes Fall 2015*. October 26, 2015.
- [3] J. J. Duderstadt, L. J. Hamilton *Nuclear Reactor Analysis*. John Wiley & Sons, Inc. 1976.