University of Illinois Urbana-Champaign
Nuclear, Plasma & Radiological Engineering Department

One Speed One Dimensional Monte Carlo Criticality Code

Made By: Mohamed Elkamash
Presented to: Prof. Madicken Munk
Course: NPRE 555
Date: 10/13/2023

Contents

1. INTRODUCTION

The one speed one dimensional Monte Carlo criticality code (OSDMC) is a nuclear code capable of simulating the transport of one speed neutrons inside a 1D medium using the Monte Carlo Method. The following parameters could be controlled by the user: the number of cells in the problem, the cross sections of each cell, the number of bins in each cell, the number of neutron histories per cycle, the number of inactive cycles and the number of active cycles. The code calculates the fission source distribution for each cycle and the convergence of the fission source could be assessed through the relative change in the fission source between two cycles. In the active cycles the code also calculates the flux distribution and k_{eff} for each cycle in addition to the cumulative average of k_{eff}. The number of bins allow the user to achieve the desired resolution of the flux distribution. In the current version of the code the user can control the input parameters through the InputParameters.cpp and Constants.h files. Chapter 2 of this report explains the theory and methodology utilized in the code. Chapter 3 discusses the results of the test case.

2. THEORY AND METHODOLOGY

2.1 Fundamental Formulation of Monte Carlo

The objective of a Monte Carlo Simulation is to simulate a physical process which its underlying physics is understood. If we can generate random numbers or in our case pseudo random numbers, we can sample the outcome of a random process x with $p(x)$. The relation between the random variable and the random number η is

$$p(x)dx = q(\eta)d\eta \qquad where \qquad a \le x \le b \quad and \quad 0 \le \eta \le 1$$

Integrating both sides on the range $[a, x]$ *and* $[0, \eta]$ yields

$$\eta = \int_0^x p(x)\, dx$$

This equation is called the Fundamental Formulation of Monte Carlo (FFMC) [1] and is used to obtain the relation between continuous random variables and random numbers given that the probability distribution of the random variable is known. Several applications of FFMC appears in section 2.2.

2.2 Neutron Simulation

2.2.1 Random Walk Algorithm

This section gives a description of the random walk algorithm of a neutron history. A neutron state is described by two variables: the position defined by the x coordinate and the direction defined by the cosine of the polar angle relative to the x-axis (μ). The first cycle neutrons are generated at a random position sampled from a uniform distribution and at a random direction sampled isotropically. The neutrons of the subsequent cycles are generated at the same position of absorption of the neutron generating them from the previous cycle. The following pseudo code explains the random walk of a single neutron:

```
1. Generate the neutron at the specified position and direction
2. Determine the current cell of the neutron based on its position
3. Get the cross sections of the cell
4. Calculate the distance to next collision
5. Calculate the distance to the nearest surface
6. If (distance to next collision < distance to nearest surface)
     o Transport the neutron to the collision position
     o Sample the reaction type
     o If (reaction is absorption)
         ▪ Neutron lifetime ends (pop neutron from the neutron bank)
         ▪ Sample the number of fission neutrons generated
```

```
            ▪ Put the new neutrons in the neutron bank to be simulated in the next cycle
            ▪ Go back to step 1
        o Else (reaction is scattering)
            ▪ Sample scattering angle
            ▪ Sample azimuthal angle
            ▪ Update neutron direction
            ▪ Go back to step 6
7. Else (distance to next collision > distance to next surface)
        o Transport the neutron to the surface
        o If (surface is the domain boundary)
            ▪ Neutron has leaked (pop the neutron from the bank)
            ▪ Go back to step 1
        o Else if (surface is the boundary of a cell with different material than the old cell)
            ▪ Go back to step 2
        o Else (surface is the boundary of a cell with same material as the old cell)
            ▪ Go back to step 2 and skip step 4
```

2.2.2 Sampling

2.2.2.1 Position of First Cycle Neutrons

The positions of the first cycle neutrons are sampled from a uniform distribution. The uniform probability distribution of position between the limits of the domain x_{min} and x_{max} is

$$p(x) = \begin{array}{ll} \dfrac{1}{x_{max} - x_{min}} & x_{min} < x < x_{max} \\ 0 & otherwise \end{array}$$

Substituting the expression for $p(x)$ in the FFMC and carrying the integration yields

$$\eta = \int_0^x p(x)\, dx = \int_{x_{min}}^x \frac{1}{x_{max} - x_{min}} dx = \frac{x - x_{min}}{x_{max} - x_{min}}$$

Inverting the expression to obtain x as a function of η yields

$$x = \eta(x_{max} - x_{min}) + x_{min}$$

2.2.2.2 Fission Neutrons Direction

The fission neutrons are assumed to be emitted isotropically. The cosine of polar angle relative to the motion axis of fission neutrons is sampled using

$$\mu = 2\eta - 1$$

For the derivation of the previous equation the reader can consult reference [1].

2.2.2.3 Distance to Next Collision

The probability of a particle moving distance r without collision in a medium of total cross section Σ_t is $e^{-\Sigma_t r}$ and the probability of the particle to interact in a differential element dr is $\Sigma_t dr$ [2]. Then the probability of a particle interacting in dr at r is $e^{-\Sigma_t r}\Sigma_t dr$. Therefore, the probability distribution function of the distance to next collision is

$$p(r) = e^{-\Sigma_t r}\Sigma_t$$

Substituting this expression in the FFMC and carrying out the integration yields

$$\eta = \int_0^r p(r)\,dr = \int_0^r e^{-\Sigma_t r}\Sigma_t\,dr = 1 - e^{-\Sigma_t r}$$

Inverting the equation to get r as a function of η

$$r = \frac{-\ln(1-\eta)}{\Sigma_t}$$

Since η is uniformly distributed between 0 and 1 then $1 - \eta$ is also uniformly distributed between 0 and 1 therefore to decrease the number of arithmetic operations in the code the equation for the distance to next collision could be modified to

$$r = \frac{-\ln(\eta)}{\Sigma_t}$$

2.2.2.4 Reaction Type

Considering a binary case of reaction types in which the neutron could be absorbed or scattered only, the reaction type could be determined using the following procedure: generate a random number η and if $\eta \le \frac{\Sigma_a}{\Sigma_t}$ the reaction is absorption, otherwise it is scattering [1].

2.2.2.5 Scattering Angle

Scattering events are assumed to be isotropic. If the neutron was moving in direction μ before scattering then the following procedure is used to calculate the new direction of the neutron. First, we sample the scattering angle isotropically using

$$\mu_0 = 2\eta_1 - 1$$

Next, we sample the azimuthal angle

$$\varphi_0 = 2\pi\eta_2$$

Then we use the following identity to calculate the new direction of the neutron

$$\mu' = \mu\mu_0 + \sqrt{1-\mu^2}\sqrt{1-{\mu_0}^2}\cos(\varphi_0)$$

For the full derivation of the equations in this section the reader can consult reference [1].

2.2.2.6 Number of Fission Neutrons

To sample the number of fission neutrons emitted in an absorption reaction we need to calculate the average number of fission neutrons emitted per absorption

$$n = \frac{\nu\Sigma_f}{\Sigma_a}$$

Because n in general will be a float and the number of neutrons should be integer in strictly analog monte carlo, the following procedure is used to calculate an integer number of neutrons per fission: generate a random number η, if $\eta \leq n - Int(n)$ then $Int(n) + 1$ neutrons are emitted otherwise $Int(n)$ neutrons are only generated, where $Int(n)$ yields the largest integer smaller than or equal to n [3].

2.3 Criticality Calculations

In the eigenvalue criticality problem, the source distribution of the neutrons is unknown because it depends on the neutron flux which depends on the source distribution. The code solves the eigenvalue problem using the power iteration technique. This section describes the eigenvalue calculation algorithm implemented in the code. We first start by listing the definition of each symbol used in this section Table 1.

Table 1 Parameters in Criticality Algorithm

Symbol	Definition
n	Cycle index
n_s	Number of inactive cycles
n_t	Total number of cycles
N_0	Number of neutron histories per cycle
i	Bin index
I	Total number of bins
F_i^n	Number of fission neutrons in bin i in cycle n
q_i^n	Number of fission neutrons in bin i in cycle n normalized by N_0
k^n	*keff* in cycle n

The positions of neutrons of the first cycle are sampled from a uniform distribution and the directions are sampled from an isotropic distribution as stated previously. Each neutron generated in the first cycle enters the neutron bank. The neutron bank represents a queue that contains the neutrons to be simulated. After the bank is populated with randomly generated N_0 neutrons, the code is ready for simulation of neutron histories. Each neutron in the bank is simulated using the random walk algorithm described in section 2.2.1. The neutron life ends when it leaks out of the system or get absorbed in a collision. If the neutron leaks it is simply removed out of the neutron bank. On the other hand, if the neutron is absorbed, then the number of fission neutrons emitted is sampled as described in section 2.2.2.6. The same number of neutrons emitted in the reaction will be created from the same position of absorption and with random directions that are sampled from isotropic distribution. These new generated neutrons will populate the bank and represents the neutrons for the next cycle. In general, the number of neutrons in the new cycle will be different than the initial cycle. If the system is supercritical this will lead to an increase in the number of neutrons in each cycle and the run time of each cycle will be greater than the previous until the computing power is exceeded. If the system is subcritical, the number of neutrons will decrease in each cycle and the uncertainty in the results will increase with each cycle. In the worst case, the number of neutrons will reach zero and no neutrons will be simulated at all. To avoid the problems with the changing of the number of neutrons between cycles, after each cycle the number of neutrons in each cycle is normalized to the initial number of neutrons per cycle. The following equation is utilized to calculate the normalized number of neutrons in each bin at the beginning of each cycle

$$q_i^n = \frac{F_i^n}{\sum_i^I F_i^n} N_0$$

The procedure of simulating each cycle continues until $n = n_s$. During the inactive cycles the tallies and the criticality are not computed. The inactive cycles serve the purpose of converging the neutron source before doing any tallies computations. One method to assess the quality of convergence in the code is the maximum relative change in the number of normalized fission neutrons in a certain bin between two successive cycle defined by the following equation

$$R_q = max \left| \frac{q_i^n - q_i^{n-1}}{q_i^{n-1}} \right|$$

This quantity should be expected to decrease by increasing the number of inactive cycles and can be used as a convergence condition.

After completing the inactive cycles, the code starts to simulate the active cycles using the same procedure but this time the k_{eff} and the neutron flux are computed in each cycle. The criticality can be computed using the following equation

$$k^n = \frac{\sum_i^I F_i^n}{\sum_i^I F_i^{n-1}}$$

And to assess the convergence of k_{eff} the following quantity is computed by the code

$$R_n = \left| \frac{k^n - k^{n-1}}{k^{n-1}} \right|$$

Also to decrease the uncertainty in the value of k_{eff}, the code evaluates a cumulative average k_c

$$k_c = \frac{1}{n - n_s} \sum_{n'=n_s+1}^{n} k^{n'}$$

.
For more information about the power iteration technique in Monte Carlo, the reader can consult references [3] and [4].

2.4 Flux Tallying

The flux is calculated in the code using a track length estimator [4].

$$\varphi_i^n = \frac{1}{L_i N_0} \sum_{j=1}^{N_0} l_j^n$$

3. TEST CASE

3.1 Problem Description

3.1.1 Assumptions

- 1D domain
- One speed neutrons
- Isotropic scattering
- Fission neutrons are emitted isotropically
- Neutrons react by absorption or elastic scattering only
- Fission neutrons are generated from absorption
- Each region in the domain is homogeneous

3.1.2 Geometry and Materials

The geometry and the material properties of the domain are found in Table 2.

Table 2 Geometry and Material Properties of The Test Case

	Region 1	Region 2
Coordinates (cm)	$0 < x < 50$	$50 < x < 100$
Σ_a (cm^{-1})	0.12	0.10
Σ_s (cm^{-1})	0.05	0.05
$\nu\Sigma_f$(cm^{-1})	0.15	0.12

3.1.3 Boundary Conditions

The boundary conditions at both ends of the domain are vacuum boundary conditions.

3.2 Results and Discussion

Two scenarios were run with different input parameters shown in Table 3.

Table 3 Parameters of The Two Scenarios of The Test Case

Parameter	Scenario 1	Scenario 2
N_0	10^4	10^5
n_s	500	200
n_{active}	500	100
$I_{region\ 1}$	5	10
$I_{region\ 2}$	5	10

3.2.1 Scenario 1

Figure 1 shows the normalized fission neutrons distribution after each 100 cycle. It can be observed from the figure that the change in fission distribution decreases significantly after each 100 cycle. From the graph we can also deduce that convergence could be accepted after 200 inactive cycles.

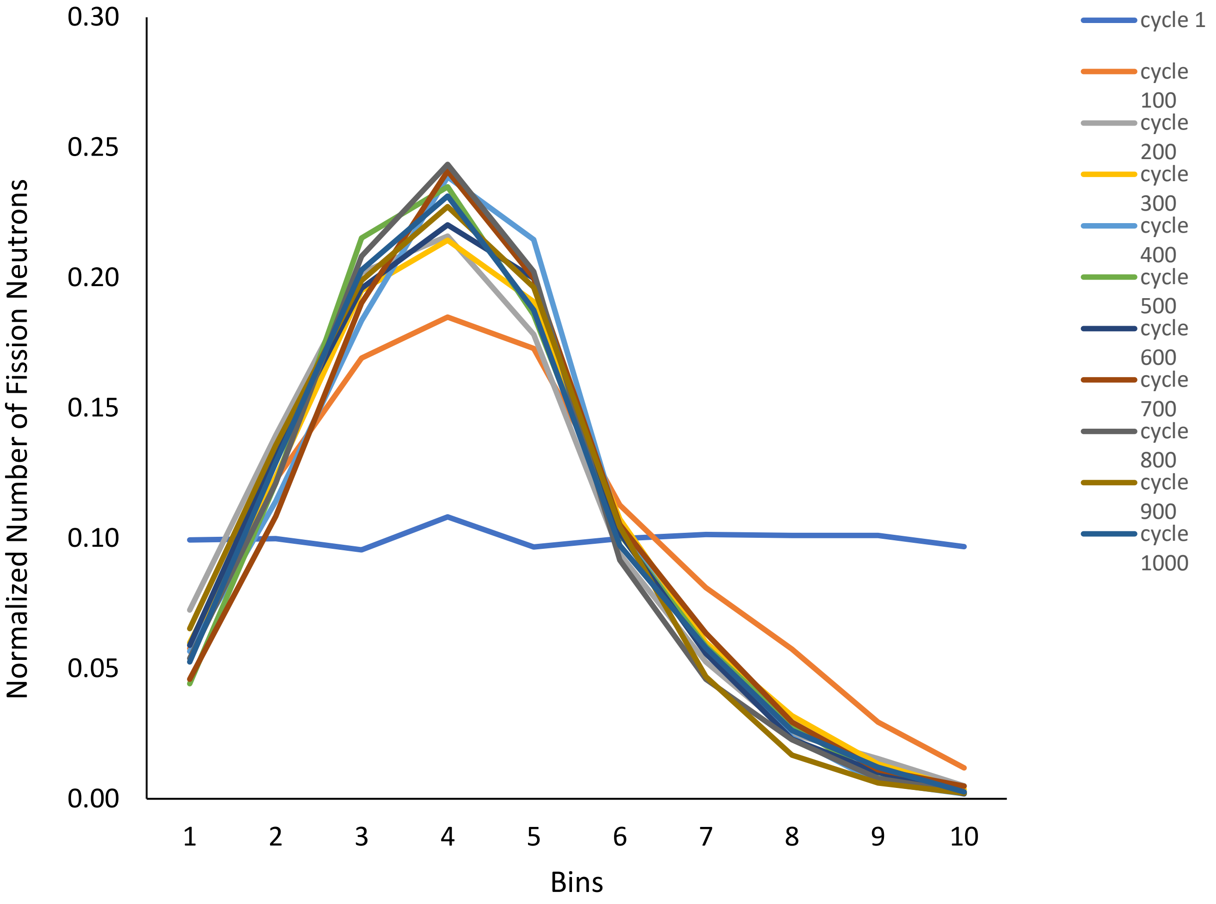

Figure 1 Normalized Number of Fission Neutrons (Scenario 1)

Figure 2 shows the normalized flux distribution after each 100 active cycle and the average flux over all cycles. It is obvious from the figure that the differences between cycles is relatively small and this is a reflection of the good convergence achieved by the source during the 500 inactive cycles.

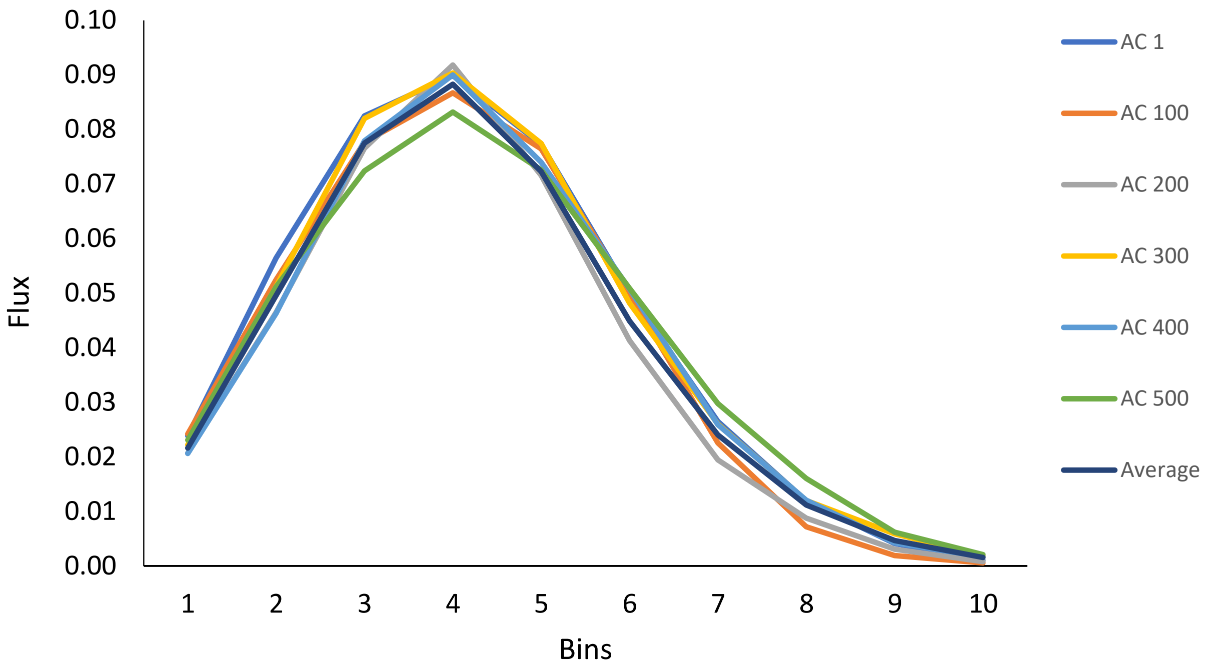

Figure 2 Flux Distribution (Scenario 1)

Finally the value of $k_{eff(cumulative\ average)}$ at the last cycle was found to be 1.13960. For the full numerical results of scenario 1, the reader should review the file results10bins.txt in the results directory.

3.2.2 Scenario 2

In scenario two, the number of bins were increased to capture the flux distribution with finer resolution. In order to avoid the higher uncertainties due to the smaller bin width, the number of neutron histories per cycle was increased from 10^4 to 10^5. Figure 3 shows the normalized fission neutrons distribution after each 50 cycles. Again it can be noticed that the source is achieving convergence. Figure 4 shows the average flux distribution with higher resolution than scenario 1. Finally the value of $k_{eff(cumulative\ average)}$ at the last cycle was found to be 1.16443. For the full numerical results of scenario 2, the reader should review the file results20bins.txt in the results directory.

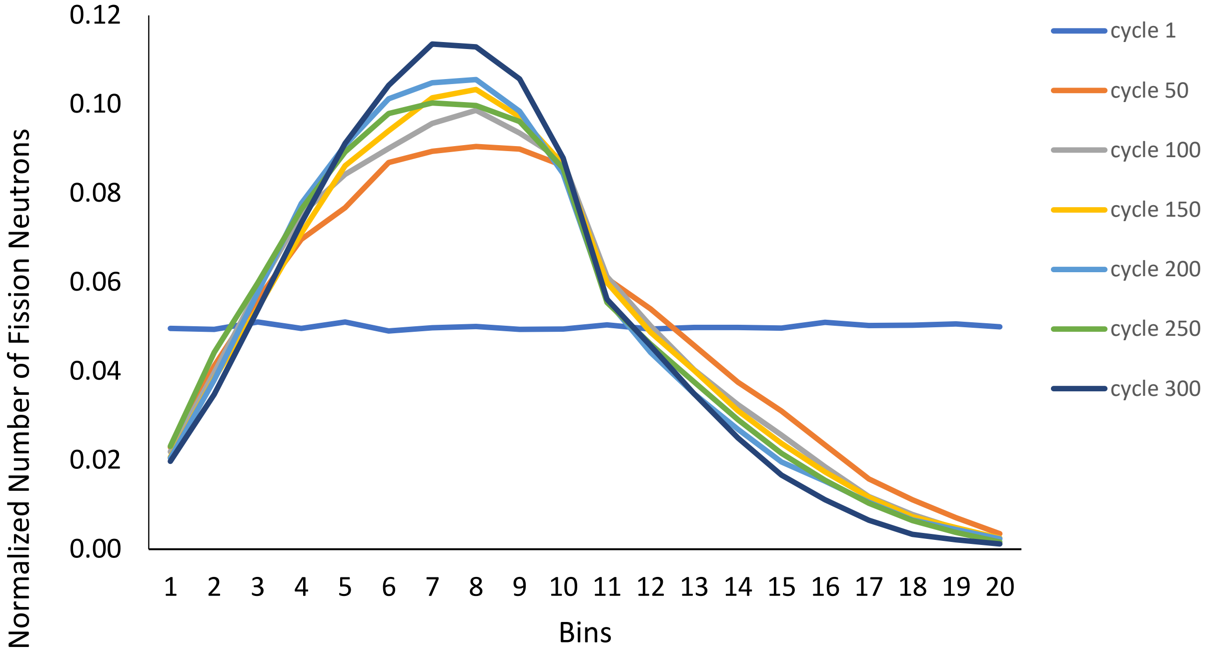

Figure 3 Normalized Number of Fission Neutrons (Scenario 2)

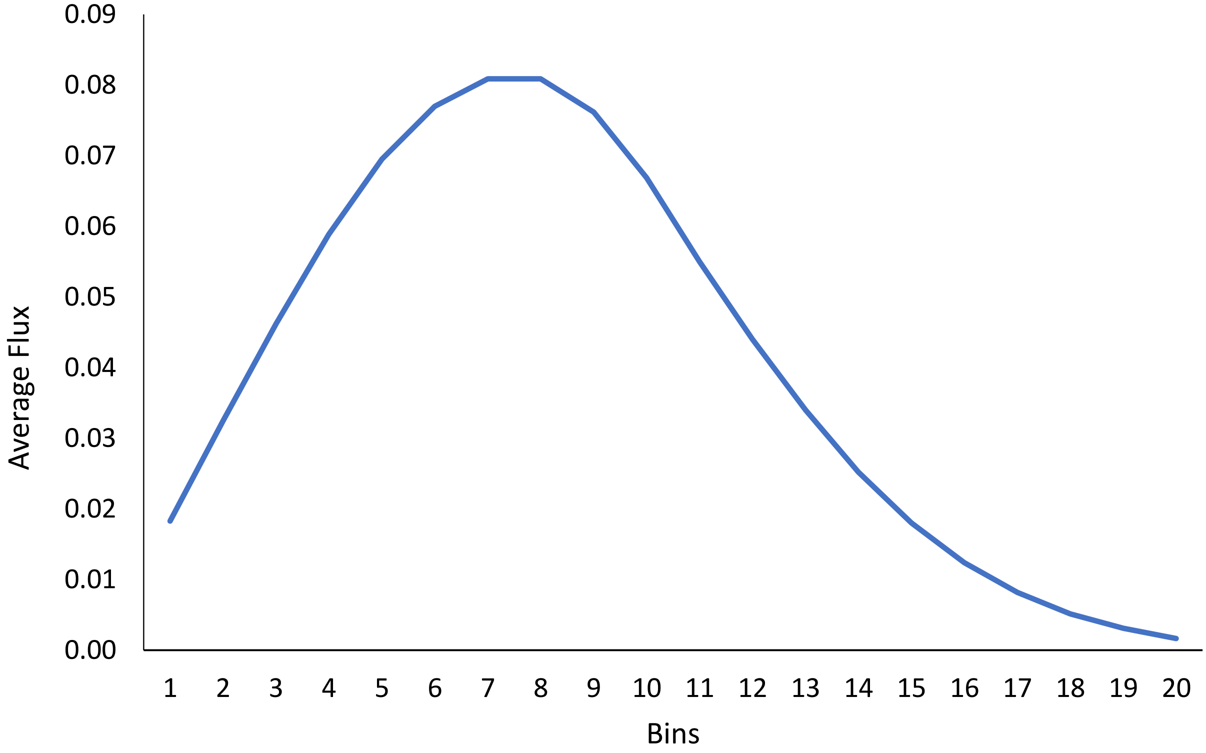

Figure 4 Average Flux Distribution (Scenario 2)

4. REFERENCES

[1] A. Haghighat, *Monte Carlo Methods for Particle Transport*, 2nd ed. New York: CRC Press, 2021.

[2] J. R. Lamarsh, A. J. Baratta, *Introduction to Nuclear Engineering*, 3rd edition. New Jersey: Prentice-Hall, Inc, 2001.

[3] W. M. Stacey, *Nuclear Reactor Physics*, 2nd ed. Weinheim: Wiley-VCH, 2007.

[4] E. E. Lewis, W. F. Miller, Jr., *Computational Methods of Neutron Transport*, New York: John Wiley & Sons, Inc., 1984.