CP1 Report

Natalie Gaughan

October 7 2020

1 Method

A Monte Carlo method follows each particle through its lifetime, using random sampling to determine each step of the process.

Steps 1-5 listed below, repeating as necessary, make up one neutron history. A cycle (or generation) consists of many histories. To simulate multiple cycles, loop over everything again and store the results from each cycle.

1.1 Step 1. New Particles

In general, we would start by sampling each particle's position, angle, and energy. For this problem, I sample position and direction. In the first cycle, I sample position randomly. In subsequent cycles, I sample position based on the locations of the fissions from the previous cycle. Because this is a one-speed problem, I do not assign energy to the neutrons. Because this is a one-dimensional problem, the only sense of angle we need is just the direction (+x or -x). The direction is random with equal probability for both directions.

1.2 Step 2. Determine the distance to the next boundary

Given the neutron's location and direction, there is only one option for which boundary will be crossed next. For example, if the neutron is in the left region and moving to the left, the relevant boundary is the left edge of the slab. If the neutron is in the left region and moving to the right, the relevant boundary is the regional boundary in the middle of the slab.

1.3 Step 3. Determine the distance to a collision

The probability distribution function (PDF) describing the distance to a collision is

$$C(s) = \Sigma_t(s) e^{-\int_0^s \Sigma_t(s')ds'}$$

Σ_t is constant in each region, so the PDF becomes

$$C(s) = \Sigma_t e^{-\int_0^s \Sigma_t ds'} = \Sigma_t e^{\Sigma_t s}$$

The cumulative distribution function (CDF) is the integral of the PDF from 0 to s

$$CDF = \int_0^s \Sigma_t e^{\Sigma_t s'} ds' = 1 - e^{-\Sigma_t s}$$

To sample the CDF, set it equal to a random number (ξ) and solve for s

$$\xi = 1 - e^{-\Sigma_t s}$$
$$s = \frac{-ln(1-\xi)}{\Sigma_t}$$

1.4 Step 4: Determine whether the next event is a collision or a boundary crossing

Compare the distance to the boundary with the distance to a collision. Whichever distance is smaller, that event happens.

1.5 Step 5a: After a collision

If Step 4 results in a collision, record location of the collision and sample the collision type.

To sample the collision type, first take the cross sections for each reaction: capture, fission, and scattering, then normalize those to Σ_t. Next, split up a number line from 0 to 1 into 3 sections for the 3 reaction types, where each section's width is the corresponding normalized cross section. Generate a random number. The collision type is whichever section the random number falls into.

If the collision is a capture, end the history. If the collision is a fission, end the history and record the location.

If the collision is a scatter, sample the new direction. I am assuming isotropic scattering, so both directions have equal probability again. Because I am not giving my neutrons energy/velocity, my code treats them as staying in the same place after scatter, and the code repeats Step 2. This means that when I record the collision location, sometimes I will get multiple scatters in the exact same location. To fix this, I do not record the collision if that exact location was already stored during that cycle.

1.6 Step 5b: After a boundary crossing

If Step 4 results in a boundary crossing where the neutron leaves the slab, end the history. If Step 4 results in the neutron crossing the regional boundary, set the neutron's new position to be the location of that boundary. In my code, I add or subtract 1e-10 to that location, so that the neutron is actually in one region or the other. This is done so that the code knows what cross sections to apply to the neutron. Then repeat the process from Step 2.

2 RESULTS

2.1 Multiplication factor

The multiplication factor, k, is the ratio between the number of neutrons in one generation and the number in the previous generation.

$$k = \frac{\text{neutrons in gen j}}{\text{neutrons in gen j-1}}$$

In terms of my code, the number of neutrons in generation j-1 is equal to the number of histories, because that is the number of neutrons present before any collisions. The number of neutrons in gen j is equal to the number of fissions times ν. I am assuming that ν, the number of neutrons released per fission, is 2.4.

k seems to fluctuate around 1.2 after about 6 cycles, so I am using 6 inactive cycles.

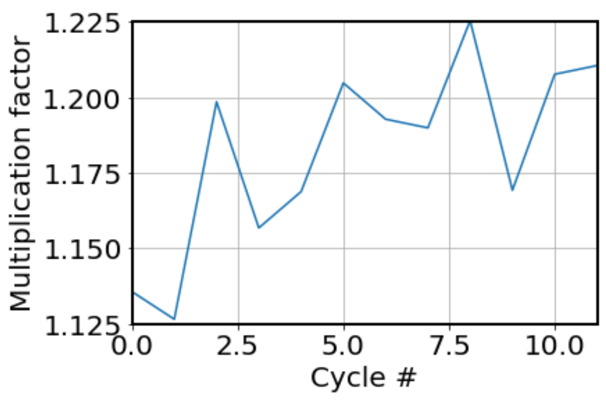

Figure 1: Evoluiton of k with each cycle

It seems reasonable to have at least as many active cycles as inactive cycles, so I ran 12 total cycles. To calculate the final result for k, I average k over the active cycles. I found k = 1.19928.

2.2 Flux

The scalar flux can be given by

$$\phi = \frac{CR}{\Sigma_t V}$$

where CR is the collision rate, Σ_t is the total macroscopic cross section, and V is the volume. The potential flaw with this method is that it does not count the neutrons that leave the slab before colliding. ϕ has units of neutrons/cm^2.

Since this is a 1D problem, it makes sense to just consider the flux as number of neutrons as a function of x. I measure the flux by simply recording the location of each collision. I divide up the slab into bins and sort the locations of collisions into a histogram. As I did with k, I average the flux over the active cycles.

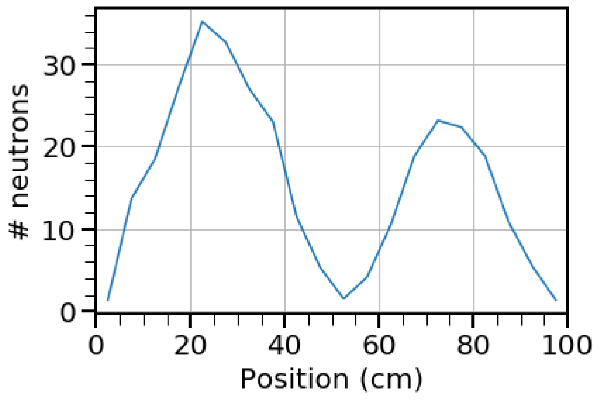

Figure 2: Neutron flux in the slab

The flux is higher in the left region of the slab, which makes sense because that region has a higher fission cross section.