NENG 685, Fall 17
Introduction to Monte- Carlo
October 30, 2017

Learning Objectives

After the class and assignments related to this material, you should be able to

1. Track particles through a geometry
2. Use mean free paths to sample the distance to the next physics event
3. Sample the physics event that occurred
4. Translate interations into a score

Tracking Particles

Recall from lesson 8, that a basic MC algorithm can be specified as Figure 1.

After we have read in the data and sampled our starting location, we have a neutral particle that is

- At point (x_p, y_p, z_p)
- Moving in direction (u, v, w)
- With energy E.

The next possible events are a collision (interaction) - shown in Figure 2

or a surface crossing - shown in Figure 3

Collisions are probabilistic (which is why MC works so well!), and can be defined as the probability of occurring at a distance, s from the start according to

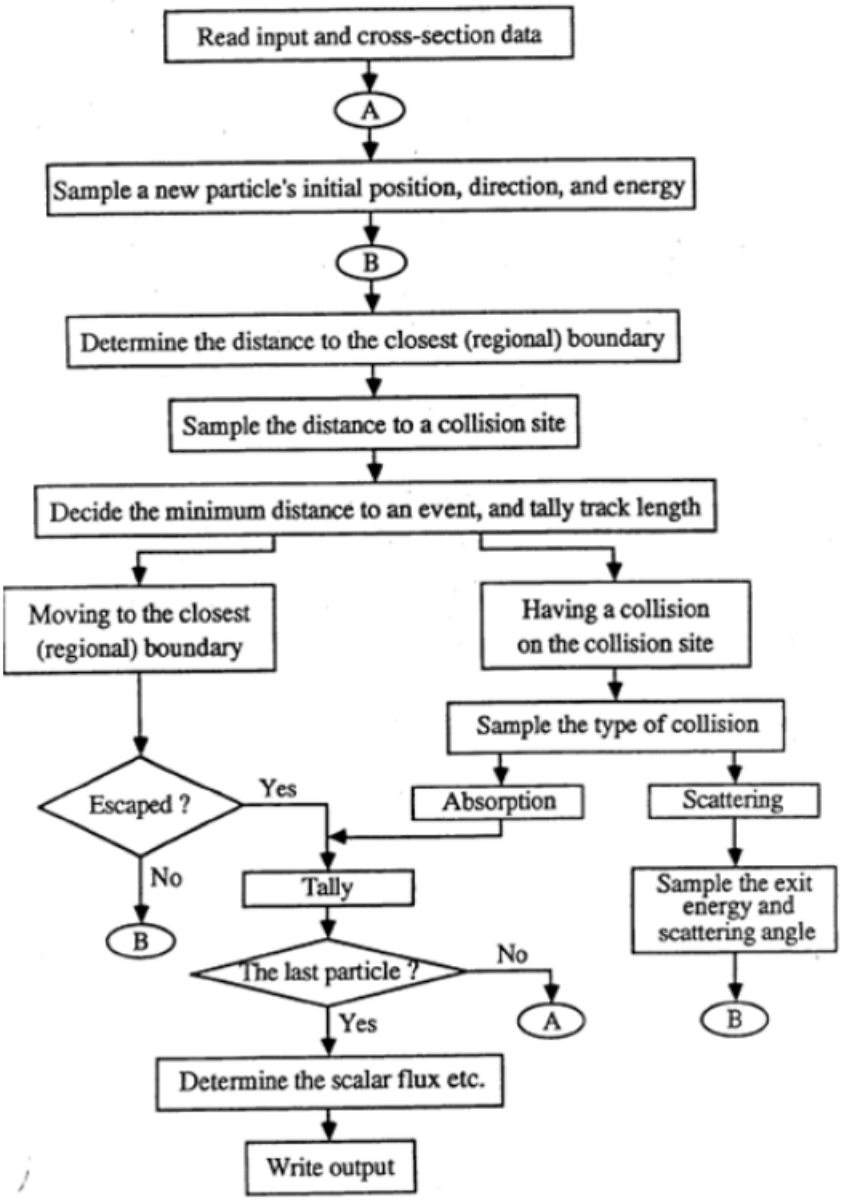


Figure 1: Basic MC algorithm for radiation transport

$$p_c(s)ds = \Sigma_t(s)e^{-\Sigma_t(s)s}ds$$
$$P_c(s) = \int_0^s \Sigma_t(s)e^{-\Sigma_t(s)s'}ds' = -e^{-\Sigma_t(s)s'}|_0^s = 1 - e^{-\Sigma_t(s)s}$$

The total cross section, $\Sigma_t(s)$ is piece-wise constant, but changes with material. To overcome this, we can do a variable transformation and measure distances in units of *mean free path (MFP)*:

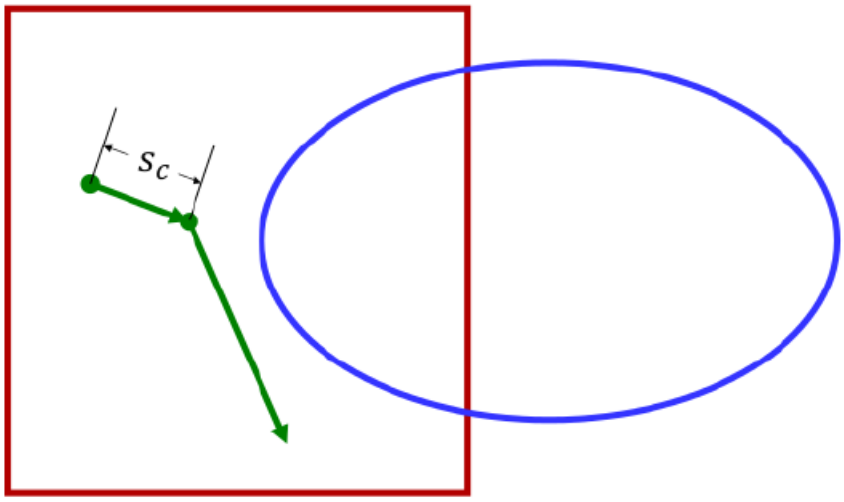


Figure 2: Collision Event

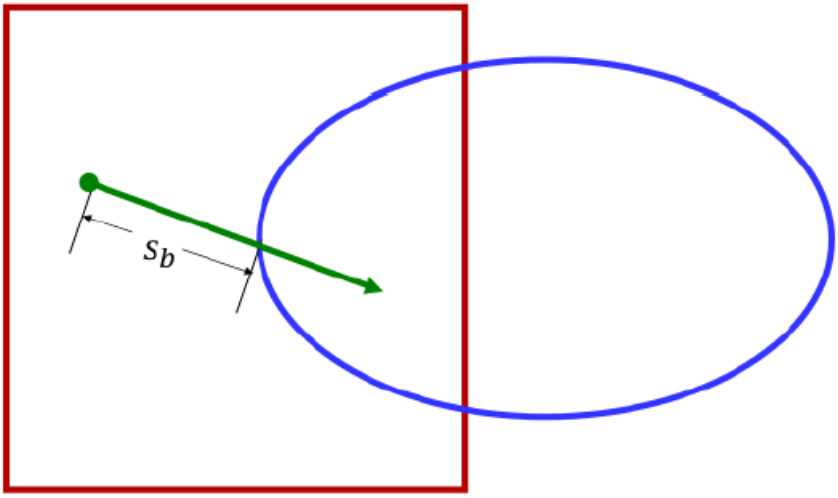


Figure 3: Surface Crossing Event

$$n = \Sigma_t(s)s$$
$$dn = \Sigma_t(s)ds.$$

We can then define the PDF and CDF as

$$p_c(n)dn = e^{-n}dn$$
$$P_c(n) = \int_0^n e^{-n'}dn' = -e^{-n'}|_0^n = 1 - e^{-n}.$$

This is now independent of material, and we can sample the number of MFPs until next collision, n_c :

- $g(n_c)dn_c = e^{-n_c}dn_c$
- $G(n_c)dn_c = 1 - e^{-n_c}$
- Directly invert to get: $\boxed{n_c = -\ln(1-\xi)}$
  [note $(1-\xi)$ is equivalent to ξ]
- In the absence of material boundaries ($\Sigma_t \neq f(s)$), the distance to a collision, s_c, is

$$s_c = \frac{n_c}{\Sigma_t}$$

However, we usually have more than one material (hence the need to go to MFP sampling in the first place). Given that our particles move in predicable straight lines, the distance to the next boundary is deterministic and can be calculated using Algebra to determine the distance between a point and the next surface, s_b. We can then convert this to units of mean free path for the current cell's material:

$$n_b = s_b\sigma_t \tag{1}$$

In MC codes, the geometry is typically represented using

- Combinatorial Surfaces
  - Define surfaces
  - Boolean operations combine surfaces to create cells
- Combinatorial Solids

  - Choose solid objects
  - Boolean operations combine objects to create regions

- B-Rep (Vertex-Edge)
  - Each object is a single set of vertices and edges connecting them

We will not define how to calculate s_b for each situation.

Ok, we have sampled the distance to the next collision, so what's next? If we refer to Figure 1, we see we are at a decision point. The options are:

$\underline{n_b > n_c}$:

- Boundary is further away than collision
- Collision occurs
- Using physics models and/or cross-sections
  - Sample target nuclide
  - Sample reaction type
  - Sample new direction
  - Sample new energy
  - Sample exiting particles
- Some of these may depend on one another
- Repeat
  - Sample new n_c following collision
  - Calculate new n_b in new direction

$\underline{n_b < n_c}$:

- Boundary is closer than collision
- Boundary crossing occurs
- Move particle along ray

  - Update $n_c = n_c - n_b$

- **DO NOT SAMPLE** for new n_c

- Calculate new n_b in new cell
  - New set of boundaries
  - New value of Σ_t

If we have a collision, we then need to determine which nuclide was hit (if there are multiple, which there often are) and which reaction occurred. The steps involved are:

- Sample **target nuclide** for a mixture with J nuclides

$$\Sigma_t = \sum_{j=1}^{J} N_j \sigma_{t,j}$$

- *Discrete PDF* to determine which nuclide is hit

$$p_j = \frac{\Sigma_{t,j}}{\Sigma_t}$$

- Sample **reaction type** for an nuclide with R types of reactions

$$\Sigma_{t,j} = \sum_{x=1}^{R} \Sigma_{x,j}$$

- *Discrete PDF* to determine which reaction occurs

$$p_x = \frac{\Sigma_{x,j}}{\Sigma_{t,j}}$$

Reaction Outcomes

After the collision, there are a couple of possible outcomes for our neutral particle:

- Particle maybe absorbed
- Particle may continue its history in a *different direction* and/or with a *different energy*

If the particle is to continue, we need to sample the new energy-angle distribution. Each is tabulated in different formats:

- Scattering laws have analytic forms with parameters in data tables
  (Direct inversion or rejection sampling)
- Tabulated data that describes a piecewise analytic interpolation
  (Hybrid sampling; we skipped this)

Scattering Angle

Scattering angles are defined relative to the original direction (considered as the z-axis)

- Polar angle, ϕ, determined by sampling from data
- Azimuthal angle, θ, determined by sampling isotropically
- The new direction is $\big(\sin(\theta)\cos(\phi), \sin(\theta)\sin(\phi), \cos(\phi)\big)$

$$= \big(\sqrt{1-\mu^2}\cos(\theta), \sqrt{1-\mu^2}\sin(\theta), \mu\big)$$

This is shown graphically in Figure 4.

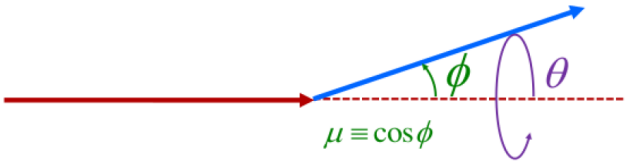


Figure 4: Scattering angle diagram.

Scoring

Now that we know what happens at the collision site, we need to formalize how we score (tally) the results for radiation transport.

We have shown how to generate a collection of histories.
Each history, i, is just a series of interaction sites.

Estimators convert each history into a score, x_i.
Each history has a different score (in general).
A *tally* accumulates a set of scores, $\{x_i\}$, to form a probability density function.

Usually we're interested in the *expected value* of the underlying PDF.
Tallies are *normalized* to the number of source particles. Thus

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

where N is the number of MC particles in the simulation source.
The absolute physical quantities require multiplication by the physical source strength (check for a given Monte Carlo code to ensure this is the way it works).

Typically, we can choose to subdivide phase space into domains or bins (energy bins for spectra, cosine bins for angular distributions, etc.) for the tallied response.

We have three main types of estimators for radiation transport:

1. point estimators: surface crossings (current tally, surface flux tally) and collisions (eigenvalue tally)
2. track length estimator: path length through a cell (volume flux tally)
3. energy balance estimator: energy loss in cell (pulse height tally)

We can now develop the estimator scores with a weight value that scales its contribution to the tally. For now the weight will always be 1, so you can functionally ignore it. However, these weights will become important when considering variance reduction techniques (only briefly described in this class).

For a surface or current tally, we count particle crossing a surface:

$$x_i = \sum_j w_{ij} \quad \text{summing over each crossing } j \text{ of history } i$$

The current [particles] is

$$\int_A dA \int_t dt \int_{\hat{\Omega}} d\hat{\Omega} \int_E dE\, \hat{n}\cdot\vec{J}(\vec{r},E,t) \approx \frac{1}{N}\sum_i^N x_i = \frac{1}{N}\sum_i\sum_j w_{ij} = \frac{W}{N}$$

And the surface energy current [MeV] is

$$\int_A dA \int_t dt \int_{\hat{\Omega}} d\hat{\Omega} \int_E dE\, E\hat{n} \cdot \vec{J}(\vec{r}, E) \approx \frac{1}{N} \sum_i \sum_j E_{ij} w_{ij} = \frac{E_T\, W}{N}$$

To get the flux/fluence in a volume

$$\bar{\phi_V} = \frac{1}{V} \int_V dV \int_t dt \int_E dE\, \phi(\vec{r}, E, t) = \frac{1}{V} \int_V dV \int_s ds \int_E dE\, N(\vec{r}, E, t)$$

where we noted that $\phi \equiv vN$ and then $vdt = ds$.
We can now see that we can use either a collision or a track length estimator for flux.
Collision (score happens at collision of particle i):

$$\bar{\phi_V} \approx \frac{1}{V} \frac{1}{N} \sum_i x_i = \frac{1}{V} \frac{1}{N} \sum_i w_i = \frac{W}{V\, N}$$

Track length ($N(\vec{r}, E, t)ds$ is density of track lengths, T_l):

$$\bar{\phi_V} \approx \frac{1}{V} \frac{1}{N} \sum_i \sum_j w_{ij} T_{l,ij} = \frac{W\, T_l}{V\, N}$$

Surface flux can also be obtained with track length tallies by accounting for angle of crossing, θ and $\mu_{ij} = \cos(\theta_{ij})$. Assume we're thinking of a volume cell that becomes infinitely thin with thickness δ.

$$T_{l,ij} = \frac{\delta}{|\cos(\theta_{ij})|}$$
$$\bar{\phi_A} \approx \frac{1}{V} \frac{1}{N} \sum_{i=1} \sum_{i=j} w_{ij} \frac{\delta}{|\mu_{ij}|} = \frac{1}{A\delta} \frac{1}{N} \delta \sum_{i=1} \sum_{i=j} \frac{w_{ij}}{|\mu_{ij}|} = \frac{W}{|\mu|\, A\, N}$$

And so on for other items of interest. Chapter 2 of the MCNP manual has a great description of common tallies for MC methods.

Statistics

The "true" mean value, μ, of any PDF is the expected value, $E(x)$

$$\mu = E(x) = \int x f(x) dx$$

Because we can't usually do this, we use random samples and estimate the true mean from the "sample" mean, $\bar{x}$

$$\bar{x} = \frac{1}{N} \sum_{i=1}^{N} x_i \qquad \lim_{N \to \infty} \bar{x} \to \mu \, .$$

The variance of a PDF is the measure of spread in that PDF

$$\begin{aligned} \sigma^2 &= E[(x-\mu)^2] = \int (x-\mu)^2 f(x) dx \\ &= \int x^2 f(x) dx - 2\mu \int x f(x) dx + \mu^2 \int f(x) dx \\ &= E(x^2) - \mu^2 \end{aligned}$$

However, we don't know the PDF so we use the samples to get the sample variances

$$\begin{aligned} S_x^2 &= \frac{1}{N-1} \sum_{i=1}^{N} (x_i - \bar{x})^2 \\ &= \frac{1}{N-1} \left[\sum_{i=1}^{N} x_i^2 - 2\bar{x} \sum_{i=1}^{N} x_i + \bar{x}^2 \sum_{i=1}^{N} 1 \right] \end{aligned}$$

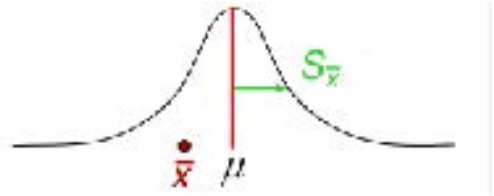
The Central Limit Theorem states that For N *independent* random variables, x_i, sampled from *identical distributions*, their mean follows a Normal (Gaussian) distribution.
We can use this information to define *confidence intervals*

$$\begin{aligned} \bar{x} - S_{\bar{x}} < E(x) < \bar{x} + S_{\bar{x}} \quad &\text{about 68\% of the time} \\ \bar{x} - 2S_{\bar{x}} < E(x) < \bar{x} + 2S_{\bar{x}} \quad &\text{about 95\% of the time} \end{aligned}$$

The **standard deviation** of the mean is a measure of the error in the result

$$
\begin{aligned}
S_{\bar{x}}^2 &= E[(\bar{x}-\mu)^2] = E\left[\left(\frac{1}{N}\sum_{i=1}^{N} x_i - \mu\right)^2\right] = E\left[\left(\frac{1}{N^2}\sum_{i=1}^{N}(x_i-\mu)\right)^2\right] \\
&= \frac{1}{N^2}E\left[\sum_{i=1}^{N}(x_i-\mu)\sum_{j=1}^{N}(x_j-\mu)\right] = \frac{1}{N^2}E\left[\sum_{i=1}^{N}\sum_{j=1}^{N}(x_i-\mu)(x_j-\mu)\right] \\
&= \frac{1}{N^2}\sum_{i=1}^{N}\sum_{j=1}^{N}E\left[(x_i-\mu)(x_j-\mu)\right] = \frac{1}{N^2}\sum_{i=1}^{N}\sum_{j=1}^{N}S_x^2\delta_{ij} = \frac{1}{N^2}\sum_{i=1}^{N}S_x^2 \\
&= \frac{NS_x^2}{N^2} = \boxed{\frac{S_x^2}{N}}
\end{aligned}
$$

The error in the results decreases with the square root of increasing the number of histories.



Relative Error is

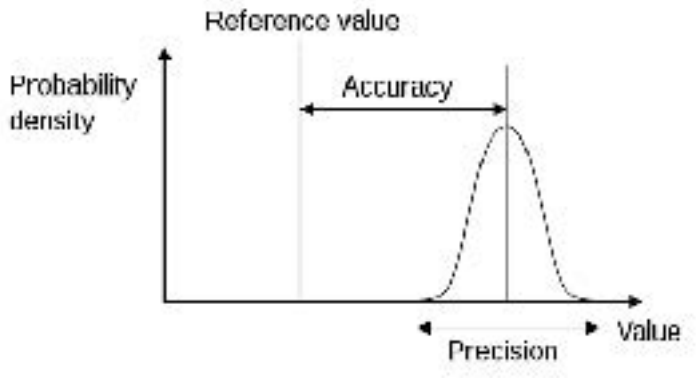
$$R = \frac{S_{\bar{x}}}{\bar{x}} = \sqrt{\frac{\sum_{i=1}^{N} x_i^2}{\left(\sum_{i=1}^{N} x_i\right)^2} - \frac{1}{N}}$$

If x_i are equal and non-zero, R=0.
Thus, we can reduce the error by reducing the spread in x_i.

Accuracy vs. Precision

The distinction between accuracy and precision can be seen in Fig. .
Accuracy is the degree of closeness of measurements of a quantity to that quantity's true value.
The *precision* of a measurement system, related to reproducibility and repeatability, is the degree to which repeated measurements under unchanged conditions show the same results.

Accuracy can be affected by systematic errors in simulation: physical and mathematical models; errors in geometry or source model; incorrect code use by user.
Usually unknown.



Conversely, precision can usually be improved: run more histories; use variance reduction; adjust your measurement (fewer scoring bins).

Variance Reduction

What we have talked about so far is *Analog* Monte Carlo:

- Natural laws are **preserved**
- The game is the "analog" of the physical problem of interest (the history of each particle is simulated exactly)
- No alteration of PDFs
- At collision, particle is killed if absorption
- Particle is born with weight 1
- weight unchanged throughout history
- Score when tallying events is 1

We often, instead, want to do *Non-Analog* Monte Carlo:

- To reduce computation time, the strict analog simulation of particles is abandoned
- Variance Reduction techniques: Absorption suppression, Russian Roulette (history termination), Splitting (history propagation), Forced collisions, Source biasing, Hybrid methods
- Alter PDFs to favor events of interest

- Particle can have different birth weight
- Weight is altered if biased PDF is used
- Particle survives "absorption" and weight is changed
- Splitting and RR can change weight
- Score current weight when tallying

We'll talk about implicit capture (a.k.a. survival biasing), roulette, splitting, and weight window maps.

The first thing to think about is how to measure success. How do we know if a calculation is "better"?
We use the figure of merit

$$FOM = \frac{1}{R^2 t} ,$$

where R is the relative error and t is the particle tracking time.
What we really want is to reduce both of these.
Why are they related to one another this way? Recall that $S_{\bar{x}} \propto \sqrt{\frac{1}{N}}$.
It's clear that without variance reduction techniques to reduce error by a factor of two you need to increase particle count (and hence time) by a factor of four.
FOM measures if we're really winning.

The idea of VR is to track particles that will contribute meaningfully to the desired results and to avoid tracking those that will not while maintaining a fair game.

The implementation of VR techniques is beyond the scope of this class, but simple methods will be described for MCNP.