Monte Carlo aided dosimetry

Random variables

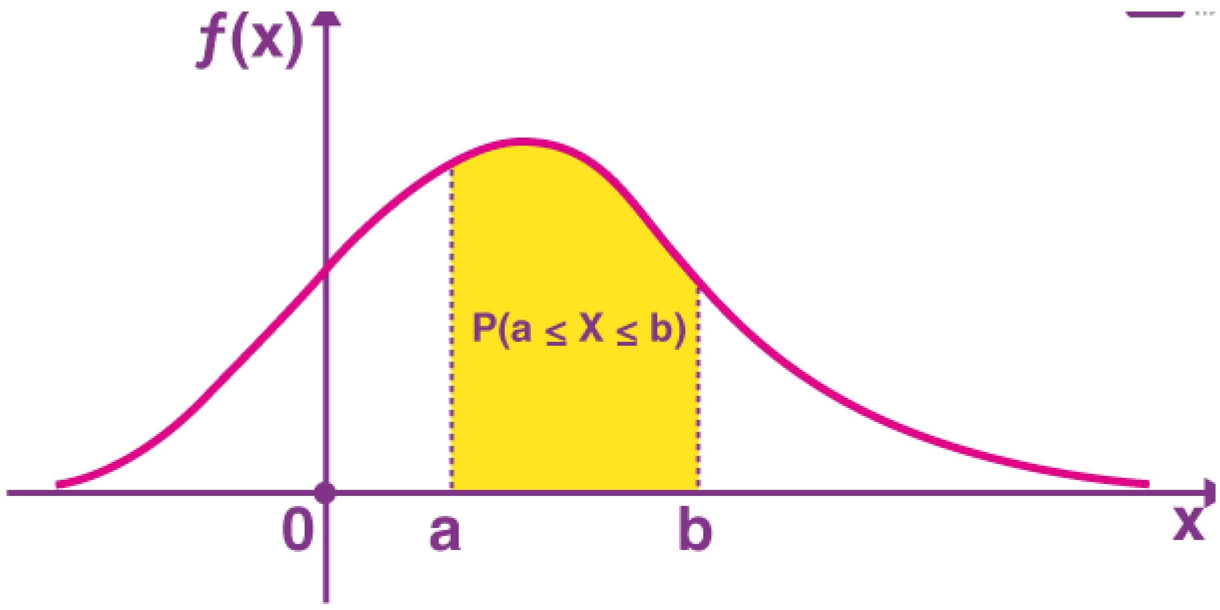
A typical example of a random variable is **the outcome of a coin toss**. Consider a probability distribution in which the outcomes of a random event are not equally likely to happen. If the random variable Y is the number of heads we get from tossing two coins, then Y could be 0, 1, or 2

Discrete random variables & Continuous random variables

A discrete variable is a variable whose value is **obtained by counting**. A continuous variable is a variable whose value is **obtained by measuring**. A random variable is a variable whose value is a numerical outcome of a random phenomenon. A discrete random variable X has a countable number of possible values.

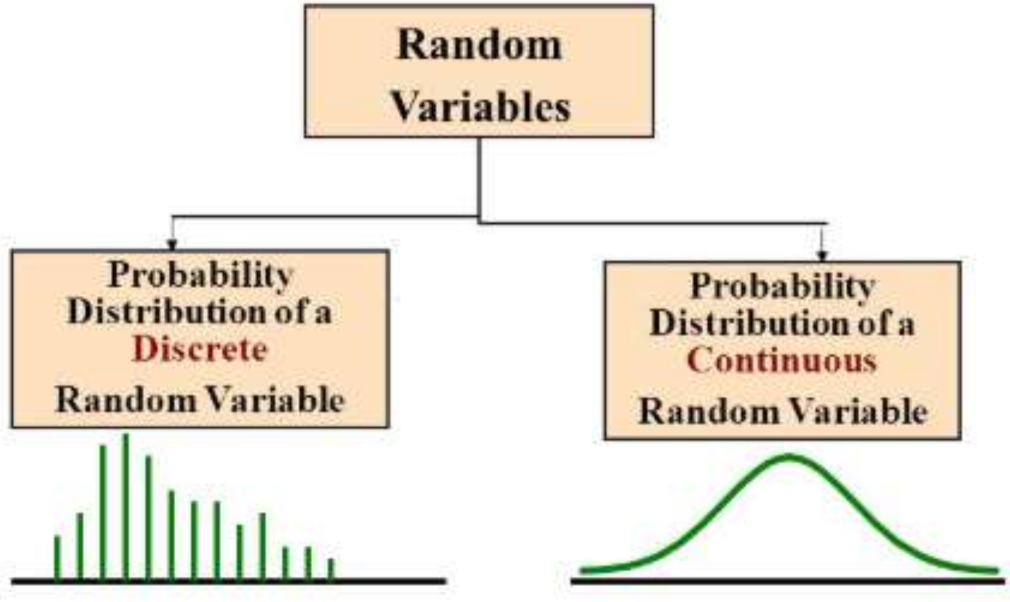
PROBABILITY DENSITY FUNCTION

a probability density function (PDF) is **used to define the random variable's probability coming within a distinct range of values, as opposed to taking on any one value**. The function explains the probability density function of normal distribution and how mean and deviation exists.



Discrete & continuous probability density function

If a variable can take on any value between two specified values, it is called a continuous variable; otherwise, it is called a discrete variable



CUMULATIVE DISTRIBUTION FUNCTION

The cumulative distribution function (CDF) of a random variable X is denoted by $F(x)$, and is defined as $F(x) = \Pr(X \leq x)$.

Using our identity for the probability of disjoint events, if X is a discrete random variable, we can write

$$F(x) = \sum_{k=1}^{n} \Pr(X = x_k)$$

where x_n is the largest possible value of X that is less than or equal to x.

Accuracy

The accuracy is calculated using the following equation :

$$Accuracy(\%) = \frac{Observed\,value - Theoritical\,value}{Theoritical\,value}$$

$$\text{Precision} = \frac{\text{True Positives}}{\text{True Positives} + \text{False Positives}}$$

Central Limit Theorem

The central limit theorem states that whenever a random sample of size n is taken from any distribution with mean and variance, then the sample mean will be approximately normally distributed with mean and variance. The larger the value of the sample size, the better the approximation to the normal.

$$\bar{X} \sim N(\mu, \frac{\sigma}{\sqrt{n}})$$

We can describe the sampling distribution of the mean using this notation:

Where:
- $\bar{X}$ is the sampling distribution of the sample means
- ~ means “follows the distribution”
- *N* is the normal distribution
- μ is the mean of the population
- σ is the standard deviation of the population
- *n* is the sample size

Random numbers and their generation

Generate one or more random numbers in your custom range from 0 to 10,000. Generate positive or negative random numbers with repeats or no repeats.

There are two main types of random number generators: **pseudo-random and true random.**

A pseudo-random number generator (PRNG) is typically programmed using a randomizing math function to select a "random" number within a set range. These random number generators are pseudo-random because the **computer program or algorithm may have unintended selection bias**. In other words, randomness from a computer program is not necessarily an organic, truly random event.

A true random number generator (TRNG) relies on randomness from a physical event that is external to the computer and its operating system. Examples of such events are blips in **atmospheric noise, or points at which a radioactive material decays**. A true random number generator receives information from these types of unpredictable events to produce a truly random number.

Tests for randomness

Example

The following sequence was observed when flipping a coin:

H, T, T, H, H, T, H, H, H, T, H, T, T, T, H, H

The coin was flipped 16 times with 9 heads and 7 tails.
There were 9 runs observed.

Values

$n = 16$

$n_1 = 9$

$n_2 = 7$

$r = 9$

Critical values from table (9,7) = 4, 14

Since $4 < r = 9 < 14$, then we Fail to reject and conclude that we don't have enough evidence to say that it is not random.

Runs Test (large case)

Large-Sample Case: If $n_1 > 20$ or $n_2 > 20$ the test statistic in the runs test for randomness is

$$z = \frac{r - \mu_r}{\sigma_r}$$

where

$$\mu_r = \frac{2n_1n_2}{n} + 1$$

$$\sigma_r = \sqrt{\frac{2n_1n_2(2n_1n_2 - n)}{n^2(n-1)}}$$

Let n represent the sample size of which there are two mutually exclusive types.
Let n_1 represent the number of observations of the first type.
Let n_2 represent the number of observations of the second type.
Let r represent the number of runs.

Critical Values for a Runs Test for Randomness

Use the standard normal table.

Hypothesis Tests for Randomness Use Runs Test

Step 1 Requirements:

1) sample is a sequence of observations recorded in order of their occurrence
2) observations have two mutually exclusive categories.

Step 2 Hypotheses:

H_0: The sequence of data is random.

H_1: The sequence of data is not random.

Step 3 Level of Significance:

Large-sample case: Determine a level of significance, based on the seriousness of making a Type I error.

Small-sample case: we must use the level of significance, $\alpha = 0.05$.

Step 4 Compute Test Statistic: Small-Sample: r Large Sample: $z_0 = \dfrac{r - \mu_r}{\sigma_r}$

Step 5 Critical Value Comparison: Reject H_0 if

Small-Sample Case: r outside Critical interval

Large-Sample Case: $z_0 < -z_{\alpha/2}$ or $z_0 > z_{\alpha/2}$

Step 6 Conclusion: Reject or Fail to Reject

Example

The following sequence was observed when flipping a coin:

H, T, T, H, H, T, H, H, H, T, H, T, T, T, H, H, H, T
T, T, T, H, H, T, T, T, H, T, T, H, H, T, T, H, T, T, T, T

The coin was flipped 38 times with 16 heads and 22 tails. There were 18 runs observed.

Values
$n = 38$
$n_1 = 16$
$n_2 = 22$
$r = 18$

$$z = \frac{r - \mu_r}{\sigma_r}$$

$$z = -0.515$$

$$\mu_r = \frac{2n_1n_2}{n} + 1 = 19.5263$$

$$\sigma_r = \sqrt{\frac{2n_1n_2(2n_1n_2 - n)}{n^2(n-1)}} = 2.96$$

Since z (0.515) < $Z\alpha/2$ (1.96) we fail to reject and conclude that we don't have enough evidence to say its not random.

Example, Using Confidence Intervals

Trey flipped a coin 100 times and got 54 heads and 46 tails, so

- $n = 100$
- $n_1 = 54$
- $n_2 = 46$

$$\mu_r = \frac{2 \cdot 54 \cdot 46}{100} + 1 = 50.68$$

$$\sigma_r = \sqrt{\frac{2 \cdot 54 \cdot 46\,(2 \cdot 54 \cdot 46 - 100)}{100^2 (100 - 1)}} = 4.94$$

$$z = \frac{r - \mu_r}{\sigma_r}$$

We transform this into a confidence interval, PE +/- MOE.

Using Confidence Intervals

- The z-value for $\alpha = 0.05$ level of significance is 1.96

$$\text{LB: } 50.68 - 1.96 \cdot 4.94 = 41.0$$

to

$$\text{UB: } 50.68 + 1.96 \cdot 4.94 = 60.4$$

- We reject the null hypothesis if there are 41 or fewer runs, or if there are 61 or more
- We do not reject the null hypothesis if there are 42 to 60 runs

Inversion random sampling technique

Inverse transform sampling is a method for generating random numbers from any probability distribution by using its inverse cumulative distribution $F-1(x)F-1(x)$.

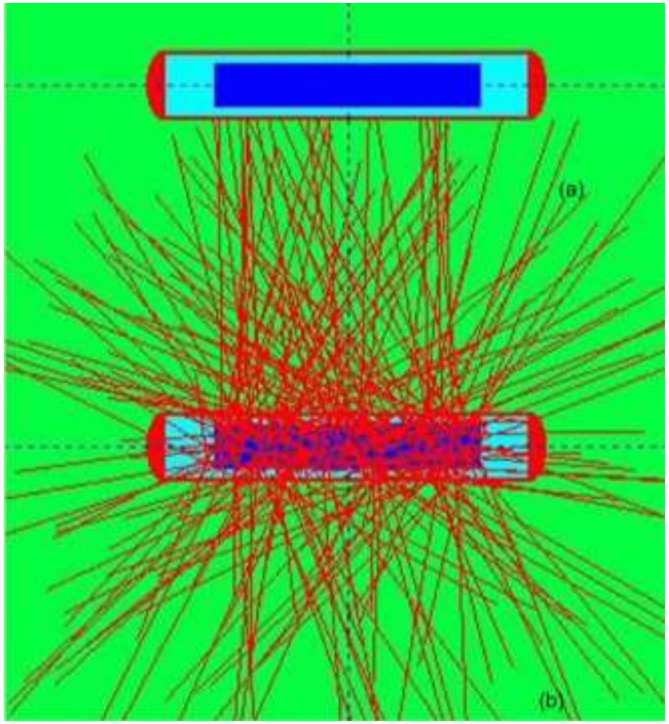
Recall that the cumulative distribution for a random variable XX is

$$FX(x)=P(X\leq x)FX(x)=P(X\leq x).$$

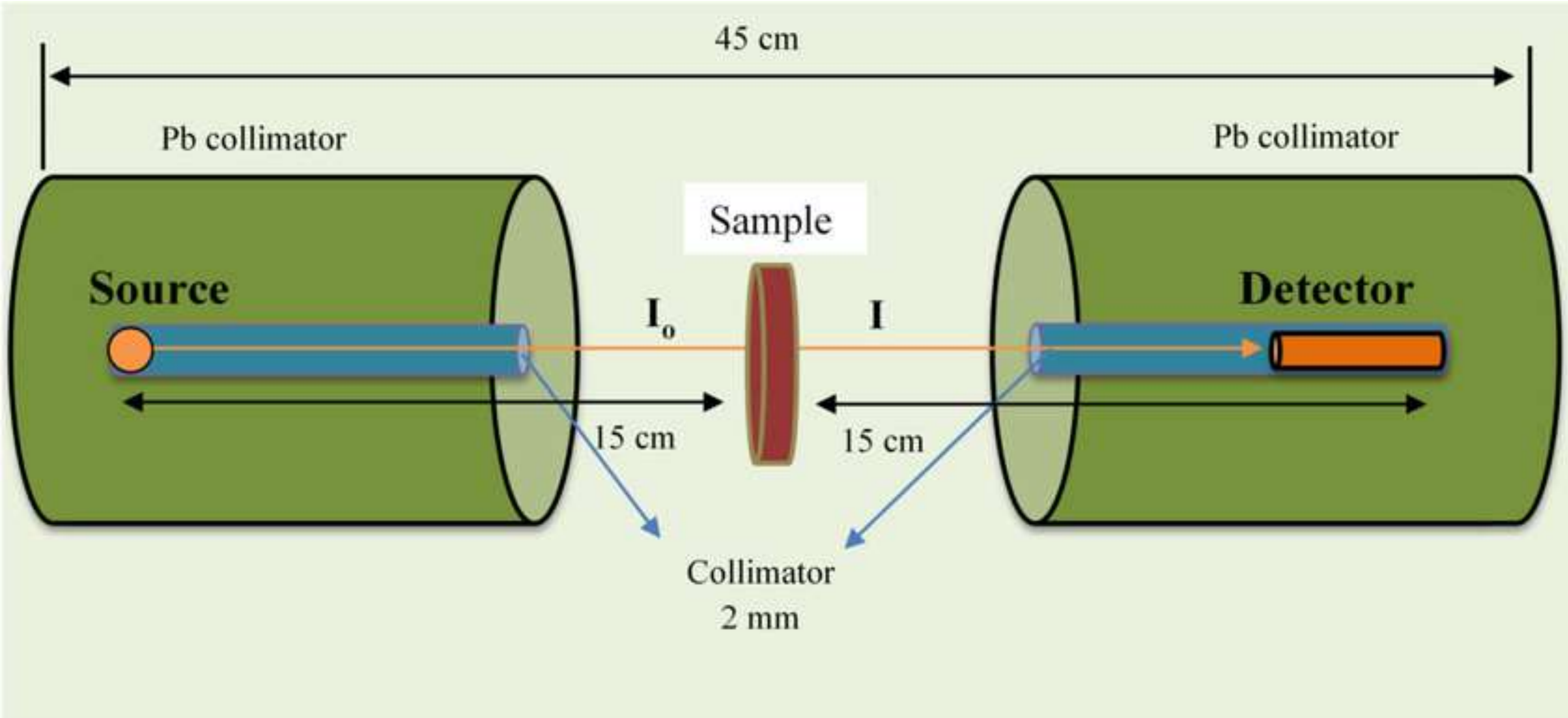
In what follows, we assume that our computer can, on demand, generate independent realizations of a random variable UU uniformly distributed on [0,1]

MCNP®, Monte Carlo N-Particle®, code can be used for general-purpose transport of many particles including neutrons, photons, electrons, ions, and many other elementary particles, up to 1 TeV/nucleon.

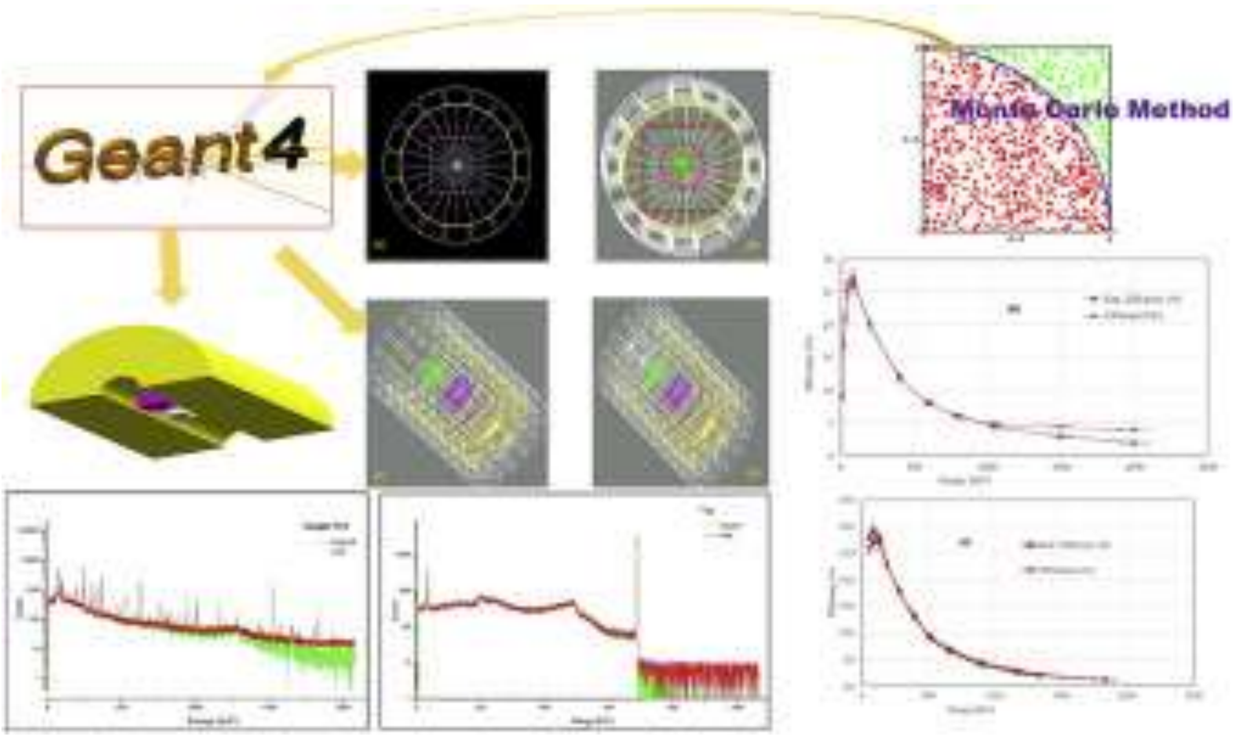
The transport of these particles is through a three-dimensional representation of materials defined in a constructive solid geometry, bounded by first-, second-, and fourth-degree user-defined surfaces.



FLUKA is a fully integrated particle physics MonteCarlo simulation package. It has many applications in high energy experimental physics and engineering, shielding, detector and telescope design, cosmic ray studies, dosimetry, medical physics and radio-biology.



Geant4 (for **GEometry ANd Tracking**) is a platform for "the simulation of the passage of particles through matter" using Monte Carlo methods. It is the successor of the GEANT series of software toolkits developed by The Geant4 collaboration, and the first to use object oriented programming (in C++). Its development, maintenance and user support are taken care by the international Geant4 Collaboration. Application areas include high energy physics and nuclear experiments, medical, accelerator and space physics studies



BEAMnrc retains compatibility with older BEAM input files that do not include EGSnrc inputs. In this case, the EGSnrc parameters simply revert to the default values used in BEAMnrc which differ from the EGSnrc defaults