UNIVERSITY OF TEXAS ARLINGTON

PHYS 5319-001: Math Methods in Physics III

Monte Carlo Application

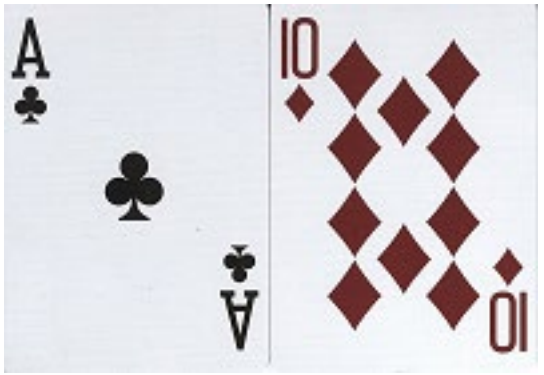
Instructor:	**Dr. Qiming Zhang**
Office:	**CPB 336**
Phone:	**817-272-2020**
Email:	**zhang@uta.edu**

Monte Carlo method

- Monte Carlo methods, or Monte Carlo experiments, are a broad class of computational algorithms that rely on repeated random sampling to obtain numerical results.
- The underlying concept is to use randomness to solve problems that might be deterministic in principle.

Simple Examples of How Monte Carlo Works

1. Odds for twenty-one



Aces are worth 11 points and the following cards are worth ten points: Jack, Queen, King.

You could write down all the possibilities:
Ten of clubs / Ace of clubs
Jack of clubs / Ace of clubs
Queen of clubs / Ace of clubs
King of clubs / Ace of clubs…

If you wrote down all of the possible combinations of cards (including all those combinations of two cards that don't add up to 21, you would find the probability of getting a Blackjack is about 1:21.

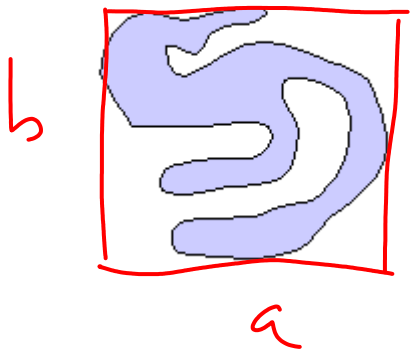
If you have a larger number of inputs — say, thousands of cards, then figuring out a sample space using a probabilistic method like this one becomes unwieldy.

Enter the Monte Carlo method: choose two cards a set number of times (say, one hundred times) and record the outcomes. The more times you take a sample of two cards, the closer you'll get to the "real" figure of 1:21.

Monte Carlo samples and locates the most likely outcome, creating a stochastic model.

The fact that Monte Carlo uses a very simple draw (in this example, two cards), and repeats it over and over again, is why the method is sometimes called **The Method of Statistical Trials.**

2. measure the area of a pond



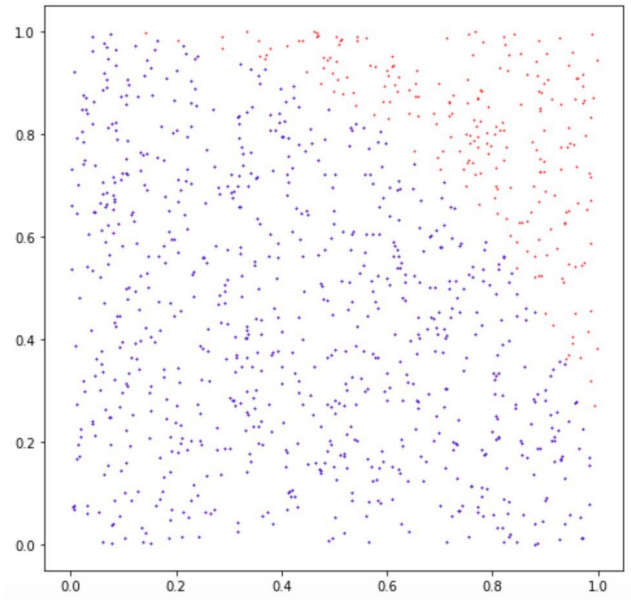
- The shape is irregular
- Draw a rectangle $a \times b$ to just enclose it
- The box area $A_{box}=ab$ *(we may rescale it to 1)*
- Generate a pair of random number within (0,1):

  (x_i, y_i) (similar to "throw" a stone)
- Check if the point (stone) is in the pond or missed

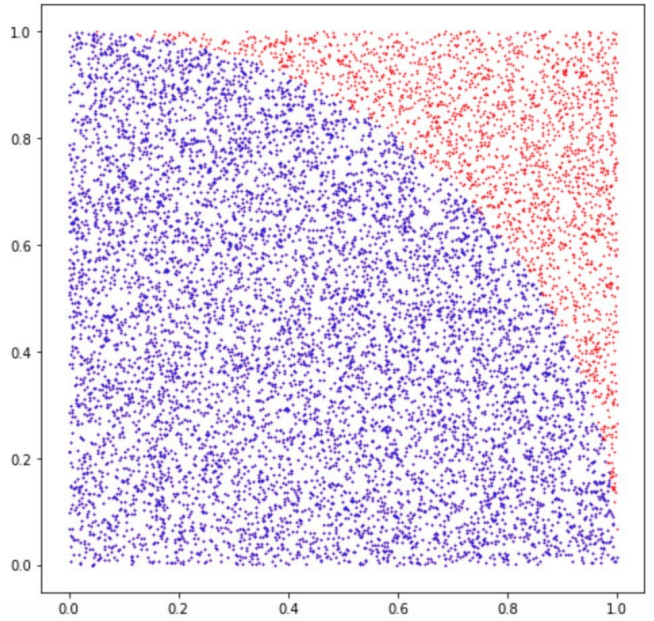
  keep counting N_{in} & N_{miss}
- Repeat this process until the sampling is large enough
- Finally, $A_{pond} = \frac{N_{in}}{N_{in}+N_{miss}} A_{box}$

3. The value of Pi

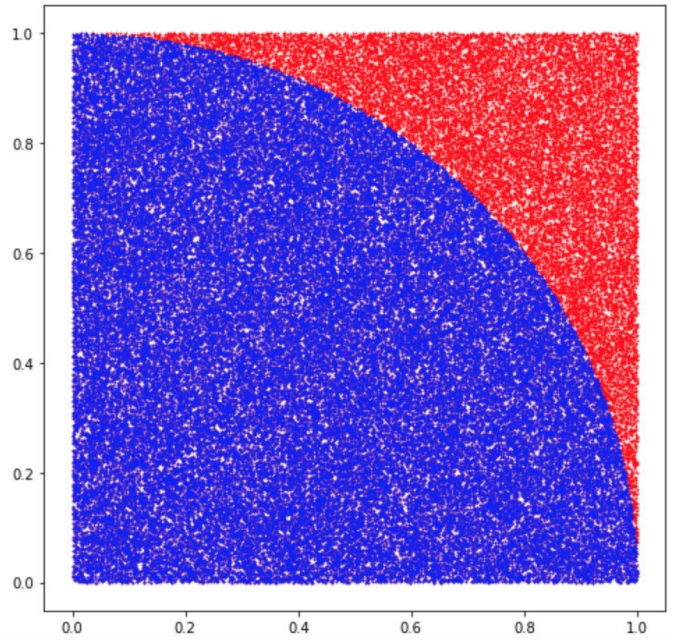
Estimate of pi: 3.116



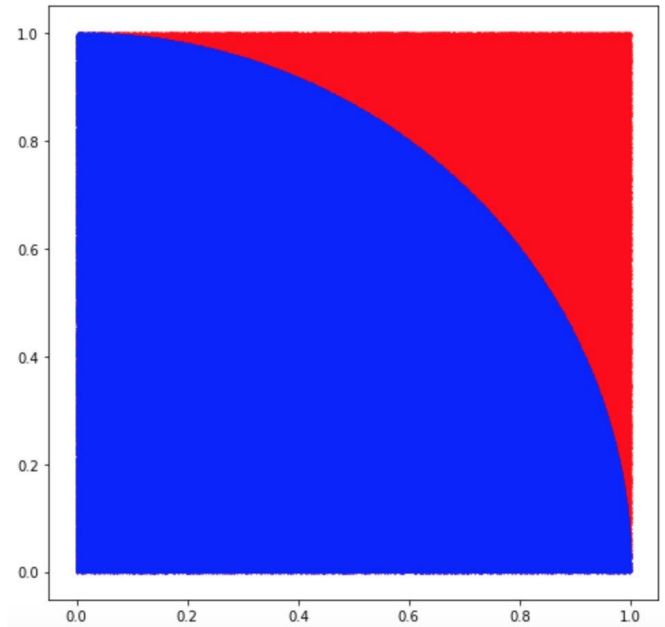
Estimate of pi: 3.142



Estimate of pi: 3.13872



Estimate of pi: 3.141932



4. Finance

MC is extensively used in financial engineering for stock market forecasting.

When picking a portfolio of stocks, you may be willing to take on different levels of risk depending on your goals. But regardless of your willingness to accept risk, you can maximize your returns per volatility of the portfolio by using Monte Carlo to find the optimal combinations and proportions of stocks. Using historical data one can generate hundreds of thousands of different combinations of stocks in different ratios, to see how each would perform relative to each other during that time period. Then one can choose the optimal configuration using a metric called a Sharpe ratio (a measure of the performance of an investment's returns given its risk.)

Monte Carlo integration

- Integral $$\int_a^b f(x)dx \approx \sum_{i=1}^{N} f(x_i)w_i$$

different algorithms to choose abscissas/weights

- Monte Carlo integration

$$I \equiv \int_a^b f(x)dx \approx \frac{b-a}{N}\sum_{i=1}^{N} f(x_i)$$

$(\vec{r}_1, \vec{r}_2, \cdots, \vec{r}_i)$

where $\{x_i\}$ are the *N* points chosen randomly from the interval *(a, b)* with uniform probability.

- For 1-d integral, a very large N is needed even for a 3 digits accuracy. Not very efficient.
- But it is more efficient for high-dimensional integration

Trapezoid rule:

$$\int_a^b \boldsymbol{f(x)dx} = \frac{\boldsymbol{h}}{2}\boldsymbol{f_1} + \boldsymbol{hf_2} + \boldsymbol{hf_3} + \cdots + \boldsymbol{hf_{N-1}} + \frac{\boldsymbol{h}}{2}\boldsymbol{f_N} + \boldsymbol{O(h^2 f^{''})}$$

Pseudo code for MC integration

- Initialize a random number, say, rand(), which often involves setting an initial seed.
- Initialize sum=0.0
- loop: *i=1:N*
  - X=rand()
  - X=X*(b-a)+a
  - sum=sum+func(X)

  end loop
- sum=sum*(b-a)/N

recall rand generate a random number within (0,1)

rescale the interval, and shift the starting point

$$I = \int_a^b f(x)\,dx$$

$$\sum f(x_i)$$

parallel

- The integral result is in 'sum'.
- If it's m-dimensional? Easy to update:

  -each point is an *m-d* random "vector" $\{\vec{x}_i\}$

  $-I = \frac{L^m}{N}\sum_{i=1}^{N} f(\{\vec{x}_i\})$

Assuming each dimension has an interval L.

- The truncation error is $O(\frac{1}{\sqrt{N}})$

0 1 2 3

Current homework #3

3. Use Monte Carlo method to calculate $I = \int_0^1 dx_1 \int_0^1 dx_2 \ldots \int_0^1 dx_{10} \, (x_1 + x_2 + \ldots + x_{10})^2$,

up to 5 significant figures. The analytic value is 155/6.

This is a 10-dimensional integral

check $O(\frac{1}{\sqrt{N}})$