Understanding Monte Carlo Simulation:Monty Hall Problem.

Mr.Shaunak Girish Khare.
Mr.Aniket Ghanshyam Desai

May 28, 2021

A PROJECT REPORT ON

**UNDERSTANDING MONTE CARLO SIMULATION:
Monty Hall Problem.**

Submitted To

UNIVERSITY OF MUMBAI, MUMBAI

In Partial Fullfillment Of

M.Sc. in Physics

by

Mr.Shaunak Girish Khare.

Mr.Aniket Ghanshyam Desai.

Under Guidance Of

Dr.M . M. Belekar. (Head Of The Department - Department Of Physics.)

Dr.V.V. Bhide.(Lecturer - Department Of Physics.)

Through

LOKAMANYA TILAK POST GRADUATE CENTER,RATNAGIRI.
2020 - 2021

DECLARATION

As required by the university, I wish to state that the work embodies in this project title “UNDERSTANDING MONTE CARLO SIMULATION:MONTY HALL PROBLEM” forms my own contribution to the project work carried out under the guidance of Dr. M. M. Belekar (HOD, Department Of Physics) and Dr. V. V. Bhide. (Lecturer, Department Of Physics), Gogate Jogalekar college, Ratnagiri. This work has not been submitted for any other degree of this or any other university.

Signature Of The Student.

Mr. Shaunak Girish Khare.

Mr. Aniket Ghanashyam Desai.

1 Acknowledgement

It is pleasure to acknowledge all who have provided help, inspiration and encouragement to complete this project. We must put on record our deepest gratitude and respect towards Dr. M. M. Belekar Sir,(Head of Physics Department) for his help and encouragement during the preparation of project.

We are very much thankful to respected Dr.M.M.Belekar Sir (Head of Physics Department) and Dr.Vivek Bhide Sir (Lecturer, Department Of Physics), for their unfailing guidance and constant encouragement. They solved clearly each and every doubt in our project during the project work.

Last but not the least we are thankful to all teachers and non-teaching staff the physics department who directly or indirectly help us to complete this project.

-Participants.

Contents

2 Introduction

Monte Carlo methods (Or Monte Carlo experiments/Monte Carlo simulation) 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.Monte Carlo simulations help to explain the impact of risk and uncertainty in prediction and forecasting models.A variety of fields utilize Monte Carlo simulations, including finance, engineering, supply chain, and science.

In order to understand this method,we're going to study in detail the classic example of 'Monty Hall Problem' by using a computer program,that uses Monte Carlo simulation,written in Python.This project report is divided into five chapters.

- **Chapter 1**is an overview of the Monte carlo method.This chapter entails the brief explanation of a concept,its historical background and its working.

- **Chapter 2** contains a brief discussions about the four classic examples of the problem which are solved by MC method.This examples are very useful in order to understand the MC algorithms.

- **Chapter3** discusses the central problem of the project (i.e. 'Monty Hall problem) in depth.This chapter contains discussions of the origin of the problem, an intuitive approach towards its solution,a MC simulation which is illustrated through a python program and the comparison between theoretical and simulation solution.

- **Chapter 4** emphasizes applicability of MC method in a plethora of fields such as physics,mathematics,engineering,finance and many more with some examples.

- **Chapter 5** concludes the complete discussion of this project report and it has list of references used while making this project.

Chapter 1

The Monte Carlo Method : An Overview

In this chapter.we are going to study the concept of Monte Carlo method,the story of the origin of the method and its working principles.

1.1 What is The Monte Carlo Method?

Monte Carlo Simulation, also known as the Monte Carlo Method or a multiple probability simulation, is a mathematical technique, which is used to estimate the possible outcomes of an uncertain event.The core idea of Monte Carlo is to learn about a system by simulating it with random sampling. That approach is powerful, flexible and very direct. It is often the simplest way to solve a problem, and sometimes the only feasible way. The Monte Carlo Method was invented by John von Neumann and Stanislaw Ulam during World War II to improve decision making under uncertain conditions. It was named after a well-known casino town, called Monaco, since the element of chance is core to the modeling approach, similar to a game of roulette.

Since its introduction, Monte Carlo Simulations have assessed the impact of risk in many real-life scenarios, such as in artificial intelligence, stock prices, sales forecasting, project management, and pricing.[1] They also provide a number of advantages over predictive models with fixed inputs, such as the ability to conduct sensitivity analysis or calculate the correlation of inputs. Sensitivity analysis allows decision-makers to see the impact of individual inputs on a given outcome and correlation allows them to understand relationships between any input variables. Monte Carlo methods vary, but tend to follow a particular pattern:

[1] Douglas A. Hubbard Douglas; Samuelson. "Modeling Without Measurements." In: *OR/MS Today.* (2009), pp. 28–33.

1. Define a domain of possible inputs
2. Generate inputs randomly from a probability distribution over the domain
3. Perform a deterministic computation on the inputs
4. Aggregate the results

1.2 A brief history of Monte Carlo method.

The simplicity and value of simulation can be summed up in its 1946 origin story. Physicist Stanislaw Ulam, who was working on the atomic bomb project at Los Alamos, was on leave recovering from an illness. To occupy his mind, he started trying to calculate the probability that a dealt solitaire hand would result in a win – all 52 cards being placed on the piles anchored by the four aces.

The attractiveness of this brute force method was enhanced by the availability of computers, which were in their infancy. In fact, the need to run simulations in support of the Manhattan project was a major impetus for the rapid development of computers. Immediately after Ulam's breakthrough, John von Neumann understood its importance. Von Neumann programmed the ENIAC computer to perform Monte Carlo calculations. In 1946, nuclear weapons physicists at Los Alamos were investigating neutron diffusion in fissionable material. Despite having most of the necessary data, such as the average distance a neutron would travel in a substance before it collided with an atomic nucleus and how much energy the neutron was likely to give off following a collision, the Los Alamos physicists were unable to solve the problem using conventional, deterministic mathematical methods. Ulam proposed using random experiments. He recounts his inspiration as follows:[2]

> The first thoughts and attempts I made to practice [the Monte Carlo Method] were suggested by a question which occurred to me in 1946 as I was convalescing from an illness and playing solitaires. The question was what are the chances that a Canfield solitaire laid out with 52 cards will come out successfully? After spending a lot of time trying to estimate them by pure combinatorial calculations, I wondered whether a more practical method than "abstract thinking" might not be to lay it out say one hundred times and simply observe and count the number of successful plays. This was already possible to envisage with the beginning of the new era of fast computers, and I immediately thought

[2] Dobriyan M. Benov. "The Manhattan Project, the first electronic computer and the Monte Carlo method." In: *Monte Carlo Methods and Applications.* 22 (2016), pp. 73–79.

> of problems of neutron diffusion and other questions of mathematical physics, and more generally how to change processes described by certain differential equations into an equivalent form interpretable as a succession of random operations. Later [in 1946], I described the idea to John von Neumann, and we began to plan actual calculations.

This being secret government work, a code-name was required, and "Monte Carlo" was chosen, in a nod to the Monaco casino town (where Ulam's uncle gambled).

1.3 How does it work?

Unlike a normal forecasting model, Monte Carlo Simulation predicts a set of outcomes based on an estimated range of values versus a set of fixed input values. In other words, a Monte Carlo Simulation builds a model of possible results by leveraging a probability distribution, such as a uniform or normal distribution, for any variable that has inherent uncertainty. It, then, recalculates the results over and over, each time using a different set of random numbers between the minimum and maximum values. In a typical Monte Carlo experiment, this exercise can be repeated thousands of times to produce a large number of likely outcomes.

Monte Carlo Simulations are also utilized for long-term predictions due to their accuracy. As the number of inputs increase, the number of forecasts also grows, allowing you to project outcomes farther out in time with more accuracy. When a Monte Carlo Simulation is complete, it yields a range of possible outcomes with the probability of each result occurring.

One simple example of a Monte Carlo Simulation is to consider calculating the probability of rolling two standard dice. There are 36 combinations of dice rolls. Based on this, you can manually compute the probability of a particular outcome. Using a Monte Carlo Simulation, you can simulate rolling the dice 10,000 times (or more) to achieve more accurate predictions.

Monte Carlo methods use random numbers, so to implement a Monte Carlo method, it is necessary to have a source of random numbers. As we mentioned above, there are a number of good methods for generating random numbers.[3]

[3] J.E.Gentle. *Computational statistics.* 2010.

Regardless of what tool one use, Monte Carlo techniques involves three basic steps:

1. Set up the predictive model, identifying both the dependent variable to be predicted and the independent variables (also known as the input, risk or predictor variables) that will drive the prediction.

2. Specify probability distributions of the independent variables. Use historical data and/or the analyst's subjective judgment to define a range of likely values and assign probability weights for each.

3. Run simulations repeatedly, generating random values of the independent variables. Do this until enough results are gathered to make up a representative sample of the near infinite number of possible combinations.

One can run as many Monte Carlo Simulations as you wish by modifying the underlying parameters one use to simulate the data. However, he/she'll also want to compute the range of variation within a sample by calculating the variance and standard deviation, which are commonly used measures of spread. Variance of given variable is the expected value of the squared difference between the variable and its expected value. Standard deviation is the square root of variance. Typically, smaller variances are considered better.

In next chapter,in order to understand this method in great detail,four classic problem has been studied briefly.

Chapter 2

The Classic Examples of Monte Carlo Simulation.

In this chapter,we are going to study some classic examples where MC simulation is proved to be effective for solving them. In every problem,there is an underlying probabilistic nature which can be efficiently handled by such simulation.Due to the high computing power of modern computers,it is possible to simulate such problems to large numbers,which is the core principle of MC simulation.

We are going to see the following examples and will discuss them briefly:

- Coin Flip Example.
- Estimating PI using circle and square.
- Buffon's Needle Problem. and
- Monty Hall Problem.

2.1 Coin Flip Example:

This is a very preliminary example in probability theory.The probability of head for a fair coin is 1/2. However, is there any way we can prove it experimentally? Due to high computing capability,we are able to simulate such experiment repeatedly for more than thousand times which is much time consuming in a real scenario.If we repeat this coin flipping many, many more times, then we can achieve higher accuracy on an exact answer for our probability value.

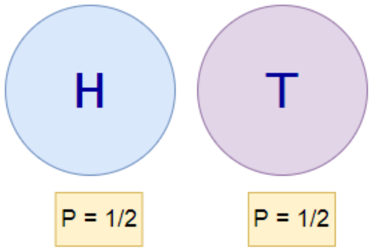


Figure 2.1: Heads and tails, mathematical representation.

While flipping a coin:

$$P(Head) = P(Tail) = 1/2$$

Now Monte Carlo algorithm for this problem cab be written as follows:

1. Simulation of real life coin flip i.e generating random numbers (in this case,0 & 1 for head and tail respectively.)
2. Define the empty list for storing the result of probability values.
3. Define variable for calculating the probability values for a event.(for Head or Tail)
4. append (extend) the list which is defined early.
5. Repeat the above procedure several times.(for 1000 or more than it for better accuracy).
6. Print the probability values or Plot them.

Using above algorithm which iterates several times,it can be seen that the probability values for each event are very near to 1/2.The number of iteration should be in accordance with **Law of large numbers.**

2.2 Estimating PI using circle and square.

The another classic example of MC simulation is the 'estimating PI using circle and square.[1]In this problem,to estimate the value of PI, we need the area of the square and the area of the circle. To find these areas, we will randomly place dots on the surface and count the dots that fall inside the circle and dots that fall inside the square. Such will give us an estimated amount of their areas. Therefore instead of using the actual areas, we will use the count of dots to use as areas.This scenario can be depicted in Figure 2.2

We know that area of the square is 1 unit sq while that of circle is$\pi * (\frac{1}{2})^2 = \frac{\pi}{4}$. Now for a very large number of generated points,

$$\frac{area\ of\ the\ circle}{area\ of\ the\ square} = \frac{No.\ of\ points\ generated\ inside\ a\ circle}{No.\ of\ points\ generated\ in\ square\ or\ total\ no.\ of\ points\ generated}$$

That is,
$$\pi = 4 * \frac{No.\ of\ points\ in\ circle}{no.\ of\ points\ in\ circle}$$

[1] A.M.Johansen. "Monte Carlo Methods". In: *International Encyclopedia of education.* Third edition (2010), pp. 296–303.

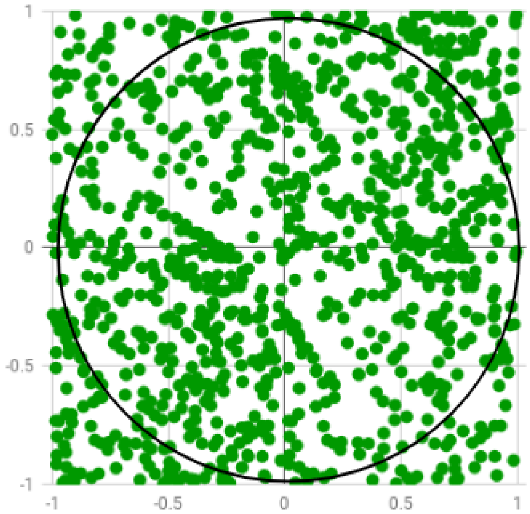


Figure 2.2: Random points are generated only few of which lie outside the imaginary circle.

Now,algorithm for this problem can be written as follows:

1. Initialize circle_points, square_points and interval to 0.
2. Generate random point x.
3. Generate random point y.
4. Calculate $d = x * x + y * y$.
5. If $d <= 1$, increment circle_points.
6. Increment square_points.
7. Increment interval.
8. If $increment < NO\_OF\_ITERATIONS$, repeat from 2.
9. Calculate $pi = 4 * (circle\_points/square\_points)$.
10. terminate

The beauty of this algorithm is that we don't need any graphics or simulation to display the generated points. We simply generate random (x, y) pairs and then check if $x^2 + y^2 <= 1$. If yes, we increment the number of points that appears inside the circle. In randomized and simulation algorithms like Monte Carlo, the more the number of iterations, the more accurate the result is. Thus, the title is "Estimating the value of Pi" and not "Calculating the value of Pi".

2.3 Buffon's Needle Problem:

A French nobleman Georges-Louis Leclerc, Comte de Buffon, posted the following problem in 1777.

> Suppose that we drop a short needle on a ruled paper — what would be the probability that the needle comes to lie in a position where it crosses one of the lines?

The probability depends on the distance (d) between the lines of the ruled paper, and it depends on the length (l) of the needle that we drop — or rather, it depends on the ratio $\frac{l}{d}$. For this example, we can interpret the needle as $l <= d$. In short, our purpose is that the needle cannot cross two different lines at the same time. Surprisingly, the answer to the Buffon's needle problem involves PI.[2]

Here we are going to use the solution of Buffon's needle problem to estimate the value of PI experimentally using the Monte Carlo Method. However, before going into that, we are going to show how the solution derives, making it more interesting.

Theorem:

> If a short needle, of length l, is dropped on a paper that is ruled with equally spaced lines of distance $d >= l$, then the probability that the needle comes to lie in a position where it crosses one of the lines is:
>
> $$P = \frac{2l}{\pi d}$$

[2] A.M.Johansen, "Monte Carlo Methods".

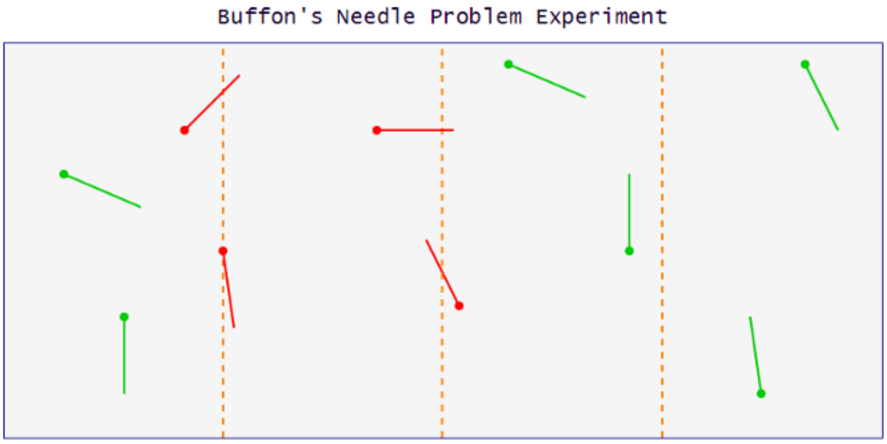
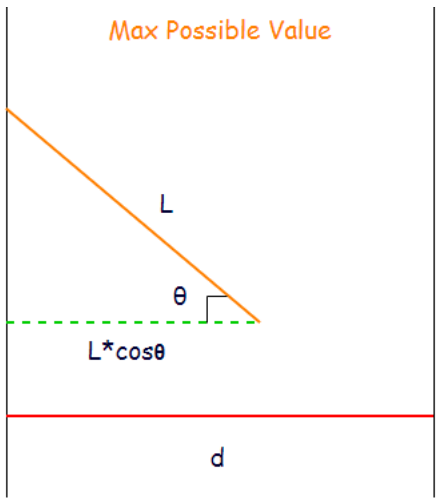


Figure 2.3: Visualizing Buffon's needle problem.

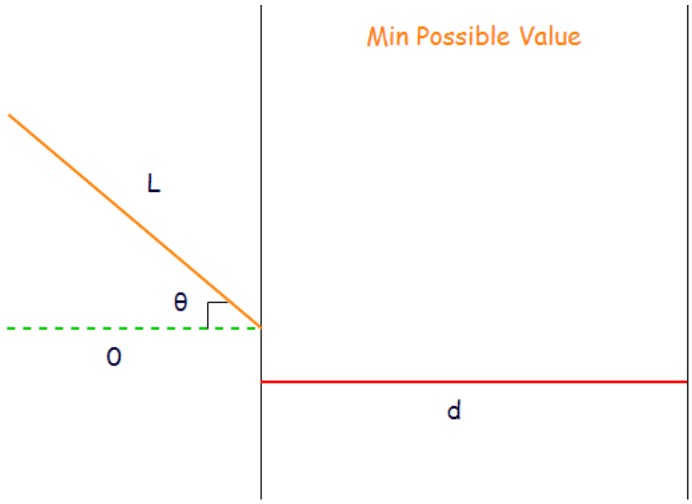
Proof:

Next, we need to count the number of needles that crosses any of the vertical lines. For a needle to intersect with one of the lines, for a specific value of theta, the following are the maximum and minimum possible values for which a needle can intersect with a vertical line.

1.Maximum Possible Value.



2.Minimum possible Value.



Therefore, for a specific value of theta, the probability for a needle to lie on a vertical line is:

$$P(land\, on\, line\, for\, given\, \theta) = \frac{lcos\theta}{d}$$

The above probability formula is only limited to one value of theta; in our experiment, the value of theta ranges from 0 to pi/2. Next, we are going to find the actual probability by integrating it concerning all the values of theta.

$$P = \frac{\#\ of\ favourable\ outcomes}{\#\ of\ total\ possible\ outcomes}$$

$$P = \frac{\int_0^{\frac{\pi}{2}} l\, cos\theta\, d\theta}{\int_0^{\frac{\pi}{2}} d\ \ d\theta}$$

$$P = \frac{l\ \int_0^{\frac{\pi}{2}}\ cos\theta\, d\theta}{d\ \int_0^{\frac{\pi}{2}}\ d\theta}$$

$$P = \frac{l}{d} \frac{[sin\theta]_0^{\frac{\pi}{2}}}{[\theta]_0^{\frac{\pi}{2}}}$$

$$P = \frac{l}{d} \frac{\sin\left(\frac{\pi}{2}\right) - \ sin(0)}{\left(\frac{\pi}{2}\right) - 0}$$

$$P = \frac{l}{d} \frac{1 - 0}{\left(\frac{\pi}{2}\right) - 0}$$

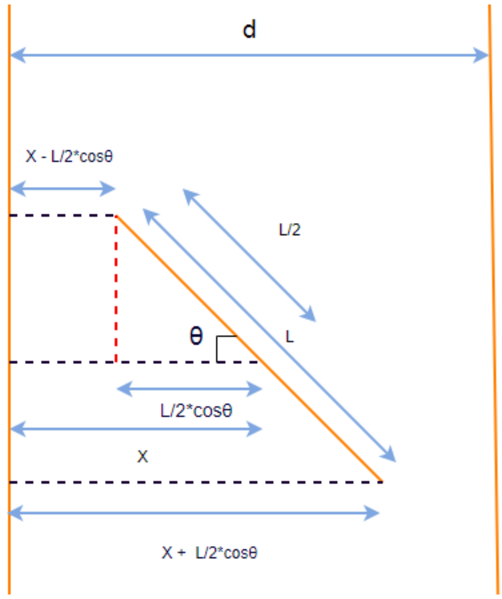
$$P = \frac{2\, l}{\pi\, d}$$

Estimating PI using Buffon's Needle Problem:

Next, we are going to use the above formula to find out the value of PI experimentally.

$$\pi = \frac{2l}{Pd}$$

Now, notice that we have the values for l and d. Our goal is to find the value of P first so that we can get the value of PI. To find the probability P, we must need the count of hit needles and total needles. Since we already have the count of total needles, the only thing we require now is the count of hit needles.
Below is the visual representation of how we are going to calculate the count of hit needles.



Algorithm The algorithm for this problem can be written as follows:

1. Initialize No. of hits as nhits=0.
2. for all needles,Generate random distances from nearest vertical line.(i.e min=0 to max=d/2)
3. Generate random values ofθ uniform over 0 to $\pi/2$.
4. check if needle crosses a line or not by calculating $xtip = x - \frac{lcos\theta}{2}$
5. if $xtip < 0$,then increment the nhits.

6. calculate the value of PI using given formula.
7. itrerate it for many times
8. terminate

2.4 Monty Hall Problem:

The Monty Hall problem is a famous, seemingly paradoxical problem[3] in conditional probability and reasoning using Bayes' theorem. Information affects your decision that at first glance seems as though it shouldn't.

In the problem, you are on a game show, being asked to choose between three doors. Behind each door, there is either a car or a goat. You choose a door. The host, Monty Hall, picks one of the other doors, which he knows has a goat behind it, and opens it, showing you the goat. (You know, by the rules of the game, that Monty will always reveal a goat.) Monty then asks whether you would like to switch your choice of door to the other remaining door. Assuming you prefer having a car more than having a goat, do you choose to switch or not to switch?

The solution is that switching will let you win twice as often as sticking with the original choice, a result that seems counterintuitive to many. The Monty Hall problem famously embarrassed a large number of mathematicians with doctorate degrees when they attempted to "correct" Marilyn vos Savant's solution in a column in Parade Magazine.

[3] Cecil. Adams. "On Let's Make a Deal, you pick door 1. Monty opens door 2 – no prize." In: *The straight dope.* (1990).

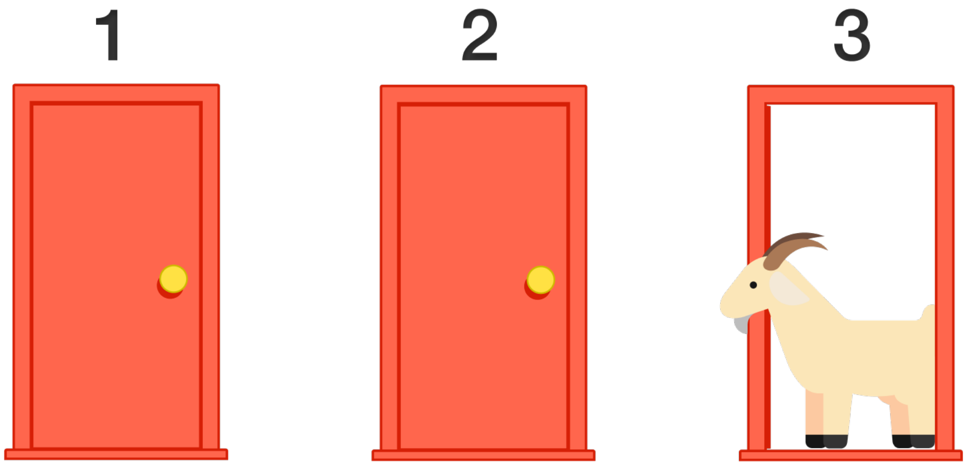


Figure 2.4: Monty hall reveals the goat.

In next chapter,we will see the intuitive approach to this problem and the solution by Monte Carlo simulation using python program in detail.

Chapter 3

Monty Hall Problem: a detailed study.

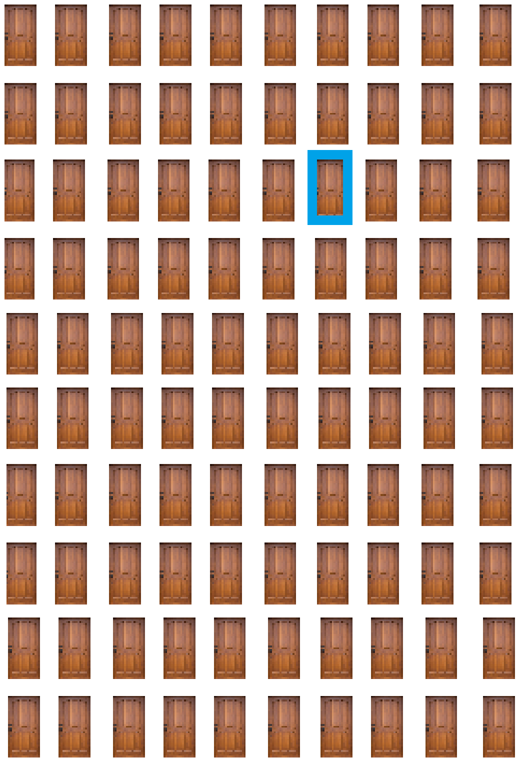
This Chapter is the core part of this project report.In last chapter,we have discussed the Monty Hall problem briefly.we have seen that switching will let you win twice as often as sticking with the original choice,which seems counter intuitive to many.In next section,We are going to see the more intuitive approach to the problem.

3.1 Why does "Switching" work? : an intuitive approach.

A lot of people have trouble with the better odds of switching doors. But they don't realize a simple fact: the odds are better if you switch because Monty curates the remaining choices.[1] Let's say you played the game where Monty doesn't know the location of the car. It wouldn't make any difference if you switch or not (your odds would be 50% no matter what). But this isn't what happens. The Monty Hall problem has a very specific clause: Monty knows where the car is. He never chooses the door with the car. And by curating the remaining doors for you, he raises the odds that switching is always a good bet.

Another of the reasons some people can't wrap their head around the Monty Hall problem is the small numbers. Let's look at the exact same problem with 100 doors instead of 3. Someone pick a random door.

[1] Edward. Barbeau. "Fallacies, Flaws, and Flimflam: The Problem of the Car and Goats". In: *The College Mathematics Journal.* 24 (1993), pp. 149–154.



Instead of one door, Monty eliminates 98 doors.These are doors that he knows do not have the prize! This leaves two doors. The one which is picked, and one that was left after Monty eliminated the others.

Now,it is wise to switch over the original choice. When the first door is picked, There is a 1/100 chance of getting the right door. Furthermore, it was sheer guesswork. Now the contestant being presented with a filtered choice, curated by Monty Hall himself. It should be clear that now odds are much better if contestant switches.

This solution, given in Parade Magazine, shows all of the possible results of staying or switching.[2]

BEHIND DOOR 1	BEHIND DOOR 2	BEHIND DOOR 3	RESULT IF STAYING AT DOOR #1	RESULT IF SWITCHING TO THE DOOR OFFERED
Car	Goat	Goat	Car	Goat
Goat	Car	Goat	Goat	Car
Goat	Goat	Car	Goat	Car

[2] Marilyn vos Savant. "Ask Marilyn". In: *Parade*. 26 (1991).

STAYING: Contestant picks door 1. Monty opens a "goat door." contestant stays. For scenario 1, contestant would win. And for the other two scenarios he would lose. Giving you a 1/3 chance of winning for all scenarios.

SWITCHING: Contestant picks door 1. Monty opens a "goat door." he switches. For scenario 1 he would lose. And this time, for the other two scenarios he would win. This gives him 2/3 odds of winning.

3.2 Monte Carlo simulation of Monty hall problem using a python problem.

Monte Carlo simulation is very effective computing method for this problem.In this section,we are going to simulate whole problem using a python program.

Flowchart: Below is given a flowchart for MC simulation of the Monty hall problem.

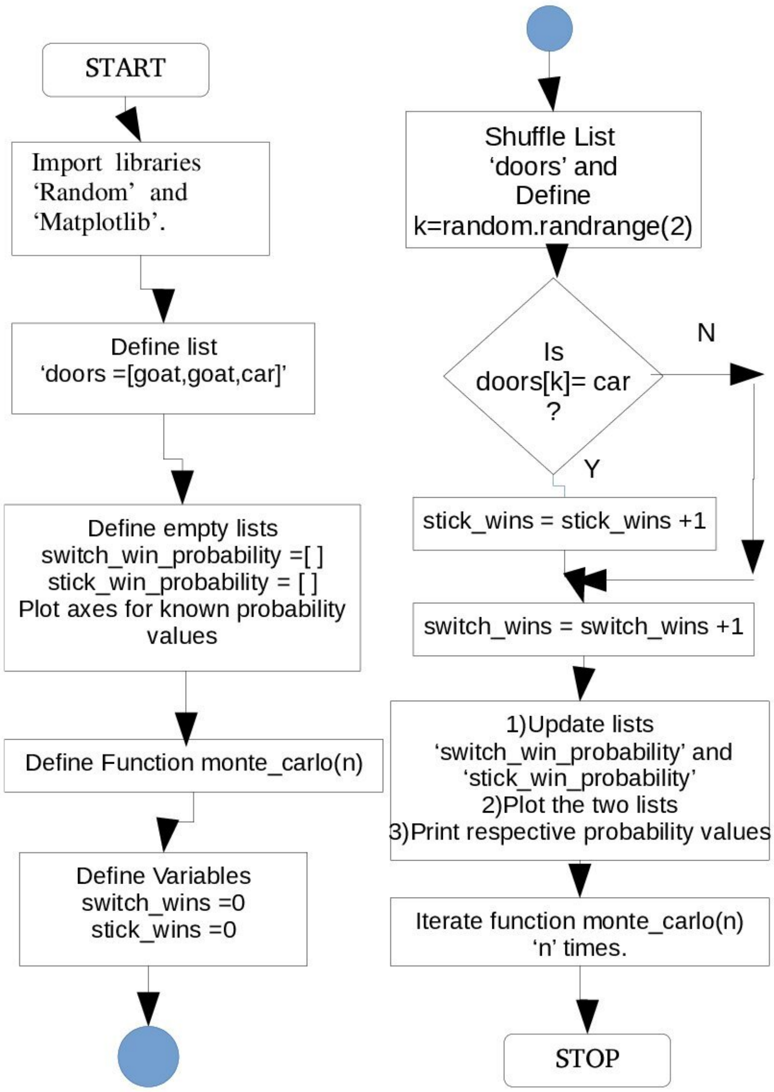


Figure 3.1: Flowchart

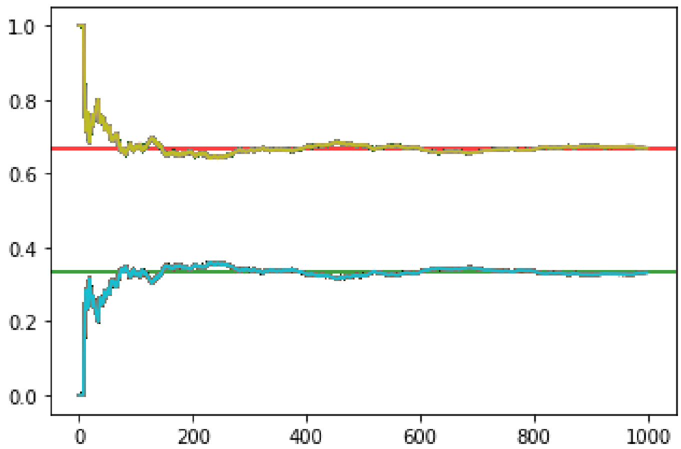
3.3 Python program.

```
#Imorting required libraries.
import random
import matplotlib.pyplot as plt


#We are going with 3 doors:
#1-goats
#2-car
doors=["goat","goat","car"]

#Empty lists to store probability values:
switch_win_probability=[]
stick_win_probability=[]

plt.axhline(y=0.66666 , color='r',linestyle='-')
plt.axhline(y=0.33333 , color='g',linestyle='-')

#Monte Carlo Simulation.
def monte_carlo(n)

  #calculating switch and stick wins.
  switch_wins=0
  stick_wins=0

  for i in range(n)
    #Randomly placing goats and car behind three doors.
    random.shuffle(doors)

    #Contestant's choice
    k=random.randrange(2)

     #if contestant doesn't get the car.
     if doors[k]!='car':
     switch_wins += 1

     #if contestant does get the car.
     else:
     stick_wins += 1
```

```
    #Updating the lists values:
    switch_win_probability.append(switch_wins/(i+1))
    stick_win_probability.append(stick_wins/(i+1))

    #Plotting the data:
    plt.plot(switch_win_probability)
    plt.plot(stick_win_probability)



#Printing the probability values:
print('Winning probability if you always switch'
,switch_win_probability[-1])
print('Winning probability if you stick to your original choice'
,stick_win_probability[-1])


    #Calling the function:
    monte_carlo(1000)
```

Output:

```
Winning probability if you always switch: 0.669
Winning probability if you always stick to your original choice: 0.331
```



As we can see from the output,that both probability values are in good agreement with theoretical values.So,the monte carlo simulation suggests that,in this case,the switching is always better option which is has probability value twice than sticking to the original choice.
In the next chapter,we will see the various applications of the monte carlo method.

Chapter 4

Applications of The Monte Carlo Physics.

Since it's emergence,Monte carlo method is developing and had found plethora of applications in physics,Artificial intelligence,finance and predominantly in risk analysis.

Risk analysis is part of almost every decision we make, as we constantly face uncertainty, ambiguity, and variability in our lives. Moreover, even though we have unprecedented access to information, we cannot accurately predict the future.
The Monte Carlo simulation allows us to see all the possible outcomes of our decisions and assess risk impact, in consequence allowing better decision making under uncertainty.
We will see various applications of MC simulation in various fields in next section.

4.1 Applications

Monte Carlo methods are especially useful for simulating phenomena with significant uncertainty in inputs and systems with many coupled degrees of freedom. Areas of application include:

1. **Physical sciences:**

   Monte Carlo methods are very important in computational physics, physical chemistry, and related applied fields, and have diverse applications from complicated quantum chromodynamics calculations to designing heat shields and aerodynamic forms as well as in modeling radiation transport for radiation dosimetry calculations.[1]In statistical physics Monte Carlo molecular modeling is an alternative to computational molecular dynamics, and Monte Carlo methods are used to compute statistical field theories of simple particle and polymer systems.[2] Quantum Monte Carlo methods solve the many-body problem for quantum systems. In radiation materials science, the binary collision approximation for simulating ion implantation is usually based on a Monte Carlo approach to select the next colliding atom. In experimental particle physics, Monte Carlo methods are used for designing detectors, understanding their behavior and comparing experimental data to theory. In astrophysics, they are used in such diverse manners as to model both galaxy evolution and microwave radiation transmission through a rough planetary surface. Monte Carlo methods are also used in the ensemble models that form the basis of modern weather forecasting.

2. **Engineering**

Monte Carlo methods are widely used in engineering for sensitivity analysis and quantitative probabilistic analysis in process design. The need arises from the interactive, co-linear and non-linear behavior of typical process simulations.For example:

- In microelectronics engineering, Monte Carlo methods are applied to analyze correlated and uncorrelated variations in analog and digital integrated circuits.
- In geostatistics and geometallurgy, Monte Carlo methods underpin the design of mineral processing flowsheets and contribute to quantitative risk analysis.
- In wind energy yield analysis, the predicted energy output of a wind farm during its lifetime is calculated giving different levels of uncertainty (P90, P50, etc.)
- impacts of pollution are simulated and diesel compared with petrol.
- In telecommunications, when planning a wireless network, design must be proved to work for a wide variety of scenarios that depend mainly on the number of users, their locations and the services they want to use. Monte Carlo methods are typically used to generate these users and

[1] Peter; Jiang Steve B Jia Xun; Ziegenhein. "GPU-based high-performance computing for radiation therapy". In: *Physics in Medicine and Biology.* 59 (2014), R151–R182.

[2] Stephan A. Baeurle. "Multiscale modeling of polymer materials using field-theoretic methodologies: A survey about recent developments." In: *Journal of Mathematical Chemistry.* 46 (2009), pp. 363–426.

their states. The network performance is then evaluated and, if results are not satisfactory, the network design goes through an optimization process.

- In signal processing and Bayesian inference, particle filters and sequential Monte Carlo techniques are a class of mean field particle methods for sampling and computing the posterior distribution of a signal process given some noisy and partial observations using interacting empirical measures.

3. **Climate change and radiative forcing:**

   The Intergovernmental Panel on Climate Change relies on Monte Carlo methods in probability density function analysis of radiative forcing.

   Probability density function (PDF) of ERF due to total GHG, aerosol forcing and total anthropogenic forcing. The GHG consists of WMGHG, ozone and stratospheric water vapour. The PDFs are generated based on uncertainties provided in Table 8.6. The combination of the individual RF agents to derive total forcing over the Industrial Era are done by Monte Carlo simulations and based on the method in Boucher and Haywood (2001). PDF of the ERF from surface albedo changes and combined contrails and contrail-induced cirrus are included in the total anthropogenic forcing, but not shown as a separate PDF. We currently do not have ERF estimates for some forcing mechanisms: ozone, land use, solar, etc.

4. Applied statistics:

   The standards for Monte Carlo experiments in statistics were set by Sawilowsky. In applied statistics, Monte Carlo methods may be used for at least four purposes:

   - To compare competing statistics for small samples under realistic data conditions. Although type I error and power properties of statistics can be calculated for data drawn from classical theoretical distributions (e.g., normal curve, Cauchy distribution) for asymptotic conditions (i. e, infinite sample size and infinitesimally small treatment effect), real data often do not have such distributions.
   - To provide implementations of hypothesis tests that are more efficient than exact tests such as permutation tests (which are often impossible to

compute) while being more accurate than critical values for asymptotic distributions.

- To provide a random sample from the posterior distribution in Bayesian inference. This sample then approximates and summarizes all the essential features of the posterior.
- To provide efficient random estimates of the Hessian matrix of the negative log-likelihood function that may be averaged to form an estimate of the Fisher information matrix

Monte Carlo methods are also a compromise between approximate randomization and permutation tests. An approximate randomization test is based on a specified subset of all permutations (which entails potentially enormous housekeeping of which permutations have been considered). The Monte Carlo approach is based on a specified number of randomly drawn permutations (exchanging a minor loss in precision if a permutation is drawn twice—or more frequently—for the efficiency of not having to track which permutations have already been selected).

5. **Artificial intelligence for games:**

Monte Carlo methods have been developed into a technique called Monte-Carlo tree search that is useful for searching for the best move in a game. Possible moves are organized in a search tree and many random simulations are used to estimate the long-term potential of each move. A black box simulator represents the opponent's moves.
The Monte Carlo tree search (MCTS) method has four steps:

(a) Starting at root node of the tree, select optimal child nodes until a leaf node is reached.
(b) Expand the leaf node and choose one of its children.
(c) Play a simulated game starting with that node.
(d) Use the results of that simulated game to update the node and its ancestors.

The net effect, over the course of many simulated games, is that the value of a node representing a move will go up or down, hopefully corresponding to whether or not that node represents a good move.
Monte Carlo Tree Search has been used successfully to play games such as Go,[81] Tantrix,Battleship, Havannah, and Arimaa.

6. **Finance and business.:**

Monte Carlo simulation is commonly used to evaluate the risk and uncertainty that would affect the outcome of different decision options. Monte Carlo simulation allows the business risk analyst to incorporate the total effects of uncertainty in variables like sales volume, commodity and labour prices, interest and exchange rates, as well as the effect of distinct risk events like the cancellation of a contract or the change of a tax law.
Monte Carlo methods in finance are often used to evaluate investments in projects at a business unit or corporate level, or other financial valuations. They can be used to model project schedules, where simulations aggregate estimates for worst-case, best-case, and most likely duration for each task to determine outcomes for the overall project. Monte Carlo methods are also used in option pricing, default risk analysis. Additionally, they can be used to estimate the financial impact of medical interventions.[3]

[3] Savvakis C. Savvides. "Risk Analysis in Investment Appraisal". In: *Project Appraisal Journal.* 9 (1994), pp. 1–34.

Conclusion

In this project,We have studied the concept of simple monte carlo method,its historical background and real life applications in science,engineering,AI and finance.The classic problems are really helpful to understand this technique.This technique is essentially deals with problems which are probabilistic or having a underlying uncertainty. we have studied the Monty Hall problem,which is a counter intuitive brain teaser based on conditional probability.We have seen its computational simulation using a python program,which is in good agreement with theory. In last section,we have studied the applications of MC simulation in various fields in great detail.

Bibliography

Adams, Cecil. “On Let’s Make a Deal, you pick door 1. Monty opens door 2 – no prize.” In: *The straight dope.* (1990).

A.M.Johansen. “Monte Carlo Methods”. In: *International Encyclopedia of education.* Third edition (2010), pp. 296–303.

Baeurle, Stephan A. “Multiscale modeling of polymer materials using field-theoretic methodologies: A survey about recent developments.” In: *Journal of Mathematical Chemistry.* 46 (2009), pp. 363–426.

Barbeau, Edward. “Fallacies, Flaws, and Flimflam: The Problem of the Car and Goats”. In: *The College Mathematics Journal.* 24 (1993), pp. 149–154.

Benov, Dobriyan M. “The Manhattan Project, the first electronic computer and the Monte Carlo method.” In: *Monte Carlo Methods and Applications.* 22 (2016), pp. 73–79.

Hubbard Douglas; Samuelson, Douglas A. “Modeling Without Measurements.” In: *OR/MS Today.* (2009), pp. 28–33.

J.E.Gentle. *Computational statistics.* 2010.

Jia Xun; Ziegenhein, Peter; Jiang Steve B. “GPU-based high-performance computing for radiation therapy”. In: *Physics in Medicine and Biology.* 59 (2014), R151–R182.

Savant, Marilyn vos. “Ask Marilyn”. In: *Parade.* 26 (1991).

Savvides, Savvakis C. “Risk Analysis in Investment Appraisal”. In: *Project Appraisal Journal.* 9 (1994), pp. 1–34.