# Monte Carlo Simulation

Jorge Guzman Nader

March 5, 2018

**Abstract**

This project creates a simulation of the probability experiment proposed by Georges-Louis Leclerc,Comte de Buffon, which creates an scenario, where a person drops needles or sticks on a grid to see how many do and do not cross a line in the grid and based in the number of needles that were dropped and the quantity that hit the lines, an estimate for $\pi$ can be calculated.Then based in the estimate $\pi$ value and our empirical data from the physical experiment, we can conclude if the simulation was accurate.

## 1 Introduction

The Monte carlo simulation is based in a proof made by a French mathematician named Georges-Louis Leclerc,Comte de Buffon which proposed a derivation using calculus and probability to estimate the likehood that a random event will occur following many permutations.

In his experiment he used a known quantity of needles and drooped them in a grid, by using the length and angles of the landing needles and integrating inside the area between 2 lines of a grid from 0 to $\frac{\pi}{2}$ he derived a useful equation that we used in our simulation to estimate the value of $\pi$.

## 2 Materials and Methods

### 2.1 Physical experiment

For the Monte Carlo experiment, I used a white cardboard with drawn lines in parallel to simulate a grid were the toothpicks were to be dropped. I used 20 toothpick of an approximate 6 cm in length. To ensure a good random distribution in the data collected, I dropped the 20 toothpick in 4 different ways, repeating each mode of dropping 5 times.

The modes of dropping were: drop 5 toothpicks each time in different locations of the grid, drop all the 20 toothpick at one time on the grid, drop the toothpicks one by one on the grid and drop the toothpicks in pair on the grid.

The times that each toothpick crossed a grid line were measured and reported to further compare them to the simulation data.The physical experiment was done over the duration of a single studio period.

### 2.2 Mathematics

For deriving the formula used in the experiment Let the needles have length x1 and the parallel lines be drawn at a distance x2 (x2 > x1) apart. A 'hit' occurs when any part of a needle crosses a line.

We can think of the center of the needle being uniformly distributed between 0 and $\frac{x2}{2}$. The smaller angle between the direction of a needle and the parallel lines will be $\theta$, so that $\theta$ is uniformly distributed between 0 and $\frac{\pi}{2}$

Then we assign y to be the distance of the mid-point of the needle from the closest line, and interception or crossing occur if: $y < \frac{x1}{2} sin(\theta)$

We can now make 2 axes with y up the vertical axis varying from 0 to $\frac{x2}{2}$, and theta along the horizontal axis varying from 0 to $\frac{pi}{2}$. To get the probability for pi we can use the area under the sine curve $\int_0^{\frac{\pi}{2}} (\frac{x1}{2}) sin(\theta) = \frac{-x1}{2}[0-1] = \frac{x1}{2}$

This makes the Probability of an intersection $= \frac{\frac{x1}{2}}{\frac{\pi}{2}\frac{x2}{2}}$

We can then notice that probability $= \frac{NeedleCross}{nofneedles} = \frac{2x1}{\pi x2}$

If we keep the value of x1 and x2 at a constant value and solve for $\pi$ we can get the final expression. $\pi = \frac{2nofneedles}{NeedlesCross}$

### 2.3 Simulation Algorithm

The main script call 2 functions to create a visual simulation of the Monte Carlo experiment, additionally it approximate $\pi$ using the Buffon derivation discussed previously and displays a histogram of the distribution for the $\pi$ values generated, lastly it displays an standard deviation for the $\pi$ results.

The functions used by the script are:

`BuffonNeedle_plot(N)`

This function generates a grid of vertical lines equally space between them using a for loop, then uses a for loop to iterate through 4 points (x1,y1) and (x2,y2) that have a random position and angle assign to them, this points will create the lines that simulate the toothpicks and the plot them using an if-else statement, that evaluates if the toothpicks crosses or not the grid lines and based on that the toothpicks are colored red (hit line) or blue(not hit line) the function uses the number of toothpicks "N" as input 'parameter.Additionally the function display a message box showing the number of toothpicks that crossed and did not crossed the grid lines and the possibility of a toothpick crossing the line based in the performed simulation.

`BuffonNeedle_pi(N)`

This function estimate $\pi$, the function is fundamentally the same than the BuffonNeedle plot but it does not plot any graphics nor display any message, the estimates for $\pi$ are based in the derived equation $\pi = \frac{2N}{nCrossed}$ where N is the total number of toothpicks and nCross the number of toothpicks that crossed a line.

## 3 Results and Discussion

### 3.1 Physical experiment

In the physical experiment the number that the toothpicks crosses a grid line was recorded for each of the 20 trials performed.

From a total of 400 toothpicks that were dropped in the grid, 268 crossed the line and 132 did not. This is equivalent % 67 that crossed a line.

### 3.2 Simulation

I ran multiple scenarios for the simulation with samples of needles ranging from 10 to 100,000. The distribution for the values obtained for $\pi$ showed that when more toothpicks were dropped, a more accurate $\pi$ value was obtained, these values were reported in a bell-like curve for easy of analysis.

## 4 Conclusion

From the physical and simulated experiment, Two important trends were noticed, the first was that the value of $\pi$ got more accurate in direct proportion to the number of needles being tested, in other words the highest the number of needles used, the closes the value of $\pi$ was to its real value.

The other important trend was that the probability of a toothpick crossing a line was bigger than not crossing, this matches the physical experiment performed and validate the simulation.

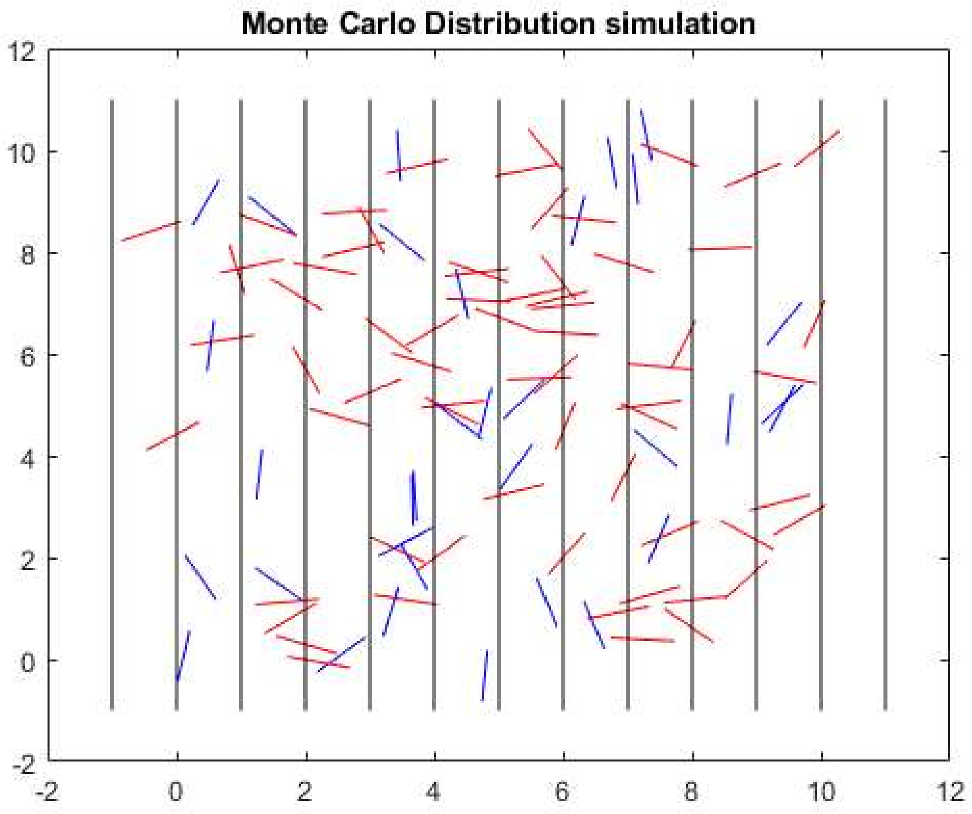
A potential application for this kind of statistical simulation can be in statistical mechanics while calculating micro and macro states for a given system, in flow dynamics to account for the randomness of a fluid model, for predict an estimate for a random set, to account for henry currents in an electrical system and others.

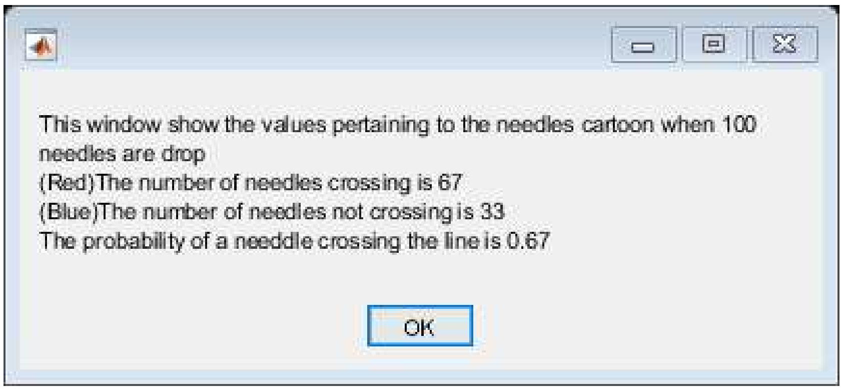
## 5 Appendix

In the next page all the published code is present for the script and the 2 used functions.

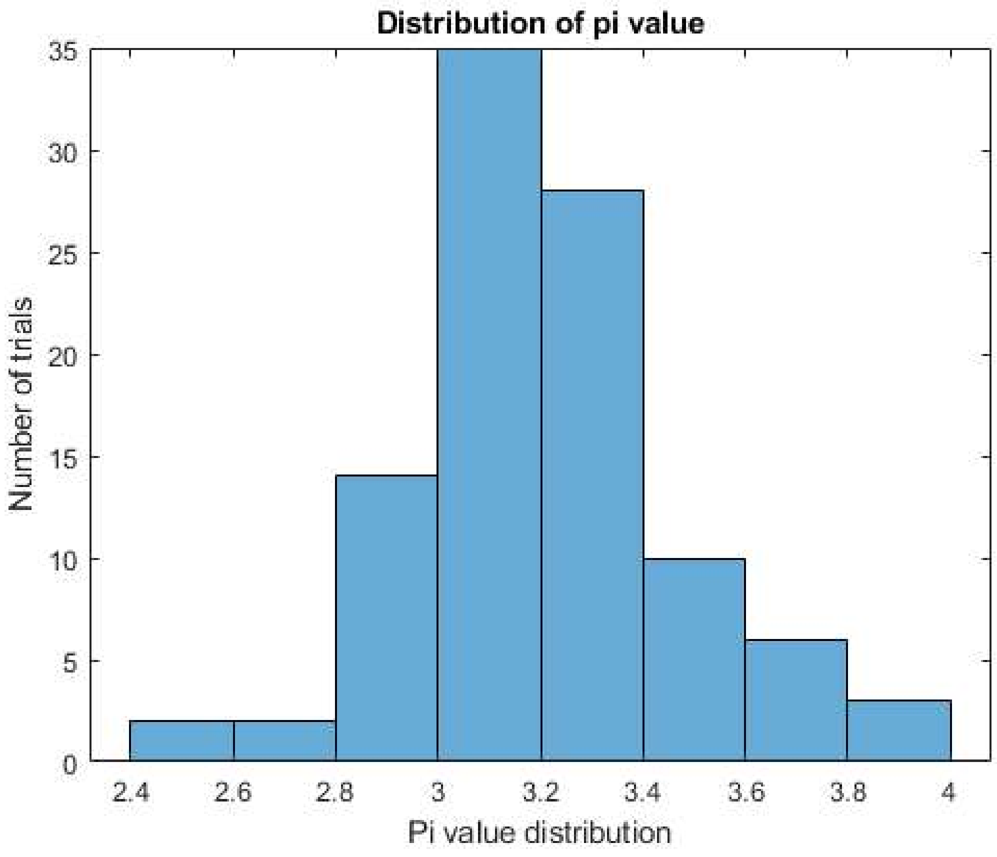
```matlab
%Georges-Louis Leclerc, Comte de Buffon, Monte Carlo Simulation Script
%Jorge Guzman Nader
%CBEE 102
%02/15/2018
%This scrip uses the next functions:
%BuffonNeedle_plot(N): Creates a cartoon of N amount of needles
%dropped in a grid of line, the needles have red color if have touch the line
%and blue color if not.
%BuffonNeedle_pi(N): Stimates the value of pi by a given number N of trials

clc

%tic,toc gives the runtime for the code
tic;

%This function plot the visual experiement of needle drop and color code it
BuffonNeedle_plot(100)

%This function calculate pi given different number of needle drops
N=100;
for i = 1:100
    pi(i)=BuffonNeedle_pi(N);
end
%Display the value of ans label as piVal
piVal = ans

%Display a histogram to show the distribution of the value of pi for every
%iteration of the for loop
histogram(pi)

%Title and labels for the histogram
title('Distribution of pi value')
ylabel('Number of trials')
xlabel('Pi value distribution')

%calculate the standar deviation for the stimated value of pi
std1 = std(pi)

toc;
```

```
h = 

  Figure (Msgbox_ ) with properties:

      Number: []
        Name: ' '
       Color: [0.9400 0.9400 0.9400]
    Position: [367.6250 311.6667 289.2500 108.5000]
       Units: 'points'

  Use GET to show all properties


ans =
    2.9851


piVal =

    2.9851


std1 =

    0.2740

Elapsed time is 0.519541 seconds.
```





This window show the values pertaining to the needles cartoon when 100 needles are drop
(Red)The number of needles crossing is 67
(Blue)The number of needles not crossing is 33
The probability of a needdle crossing the line is 0.67


Distribution of pi value
Number of trials
Pi value distribution
0
5
10
15
20
25
30
35
2.4
2.6
2.8
3
3.2
3.4
3.6
3.8
4

```
%Monte Carlo simulation (Plot and cartoon)
%Jorge Guzman Nader
%CBEE 102
%02/15/2018
%This function display a needle drop simulation and output a window showing
%the quantity of needles that touched the line in red and the ones that did
%not in blue, it also display the numerical probability that a needle
%crosses the line


function piValue = BuffonNeedle_plot(N)


    % Run the simulations

    %Set the counter for times that the needles cross the line
    numCross = 0;

    %loop dropping needles
    for idx = 1:N

        %Create first point in the line
        x1 = 10* rand(1);
        y1 = 10* rand(1);

        % Set the angle
        theta = 2*pi * rand(1);

        % Create the second point
        x2 = x1 + cos(theta);
        y2 = y1 + sin(theta);

        % Check if a needle intersects a line
        cross = floor(x1) ~= floor(x2);


        %Color needle red when needle intersects line
        if floor(x1) ~= floor(x2)
            plot([x1, x2], [y1, y2], 'red')
            hold on
        %Color needle blue when needle do not intersects line
        else
            plot([x1, x2], [y1, y2], 'blue')
            hold on
        end


        % Count how many needles intersect a line and how many do not
        numCross = cross + numCross;
        noCross = N - numCross ;

        % Return the probability that a needle intersects a line
        probability = numCross / N;

    end
```

```
     %message box indicating the number of needles that hit or not the line
     h = msgbox(sprintf('This window show the values pertaining to the needles cartoon when 1
00 needles are drop\n(Red)The number of needles crossing is %2.3g\n(Blue)The number of needle
s not crossing is %2.3g\nThe probability of a needdle crossing the line is %2.3g\n',numCross,
 noCross, probability))

    %Calculates PI = 2* needle length/(distance between grids * probobility)
    piValue = 2*N/numCross;


    % Plot the vertical lines grid that will serve as boundaries
    % for the needles
    for line_idx = -1:11
        plot( [line_idx,line_idx], [-1,11], 'black')
        hold on
    end

    %Plot title and labels
    title('Monte Carlo Distribution simulation')
    %legend('Hit ','red', 'No hit','blue')

    %When the script is executed 'figure' make the 2 windows output from
    %the functions be able to display, wihout overlap eachother
    figure

end
```

```
Not enough input arguments.

Error in BuffonNeedle_plot (line 20)
    for idx = 1:N
```

*Published with MATLAB® R2017b*

```
%Monte Carlo simulation (pi stimation)
%Jorge Guzman Nader
%CBEE 102
%02/15/2018
%This function calculate an stimate for pi, based in the Monte Carlo
%derivation.



function piValue = BuffonNeedle_pi(N)


    % Run the simulations

    %Set the counter for times that the needles cross the line
    numCross = 0;

    %loop dropping needles
    for idx = 1:N

        %Create first point in the line
        x1 = 10* rand(1);
        y1 = 10* rand(1);

        % Set the angle
        theta = 2*pi * rand(1);

        % Create the second point
        x2 = x1 + cos(theta);
        y2 = y1 + sin(theta);

        % Check if a needle intersects a line
        cross = floor(x1) ~= floor(x2);


        % Count how many needles intersect a line and how many do not
        numCross = cross + numCross;
        noCross = N - numCross;


        % Return the probability that a needle intersects a line
        probability = numCross / N;

    end

     %Calculates PI = 2* needle length/(distance between grids * probobility)
     piValue = 2*N/numCross;


end
```

```
Not enough input arguments.

Error in BuffonNeedle_pi (line 19)
```

```
      for idx = 1:N
```