Modelling and Simulation
Lab Assignment - 5

Rushi Vachhani
Student Id: 201601422
DAIICT - Gandhinagar
201601422@daiict.ac.in

Kathan Vaidya
Student Id: 201601465
DAIICT – Gandhinagar
201601465@daiict.ac.in

I. Abstract

Modeling with Randomness:
In this lab we will explore the techniques of Monte Carlo simulation to find the integral and estimating the value of pi. We will also touch upon the methods for random number generation.

II. Problem Statement

In this problem you are supposed to use Monte Carlo method for your calculation. You can use the inbuilt random number generator. You are supposed to do the problem in two ways

(i) Through a single run and increasing values of the number of random numbers
(ii) Run the simulation many times and then calculate the average.

In each of the cases you should show through a single figure how the estimate improves/converges as the length of the runs is increased or the number of runs increases.

- Using Monte Carlo Method calculate the following:

A. Calculate the value of integral:

We want to calculate the area between the curves $f(x) = x^2$ and x – axis from x = 0 to x = 2.
So, in Monte Carlo technique we generate uniform random numbers whose x range is [0,2] and y range from [0,4]. The we see whether the point lies under the curve or above it. Now the ratio of number of points lying under the curve to total number of points generated gives us the value of integral.

(i) Here firstly we will use the first method (i.e. for single run increase the value of random numbers to be generated.)

In this method we first generate a specified number of random numbers and calculate the integral in single run.

The following graph is obtained for various number of random numbers generated in a single run:

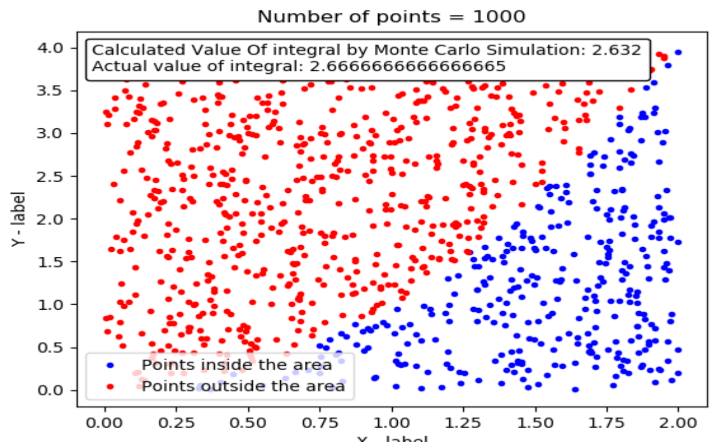

Figure 1.

The above graph is obtained by generating 1000 random numbers. As we can see that the random numbers are scattered in the region. The dots in the blue are the random numbers which are under the curve and the red ones are outside.
Now by taking the ratio of number of blue dots to total number of points generated (i.e. 1000), the calculated value of integral turns out to be 2.632 whereas the actual value of integral is 2.66665. So, we can see that with n = 1000 we have achieved precision of just 1 decimal point.

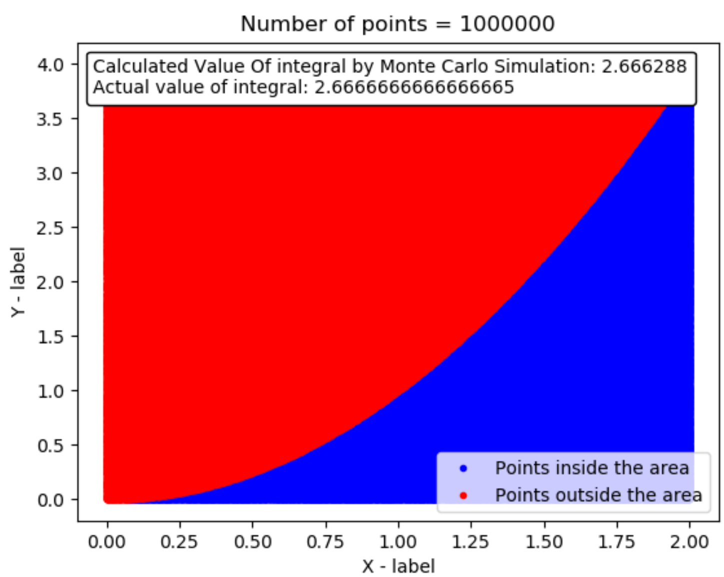

Figure 2.

Here the number of random points generated is 10,00,000 and by generating so many random variables we see that the calculated value of integral turns out to be 2.66288 which is close to 2.66667 and has the precision of just two decimal numbers.

(ii) Now we will run the simulation many times and then take the average.

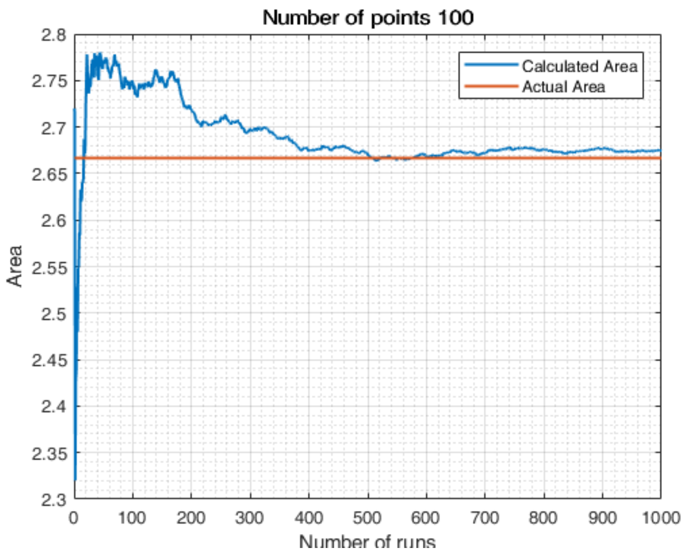

Figure 3.

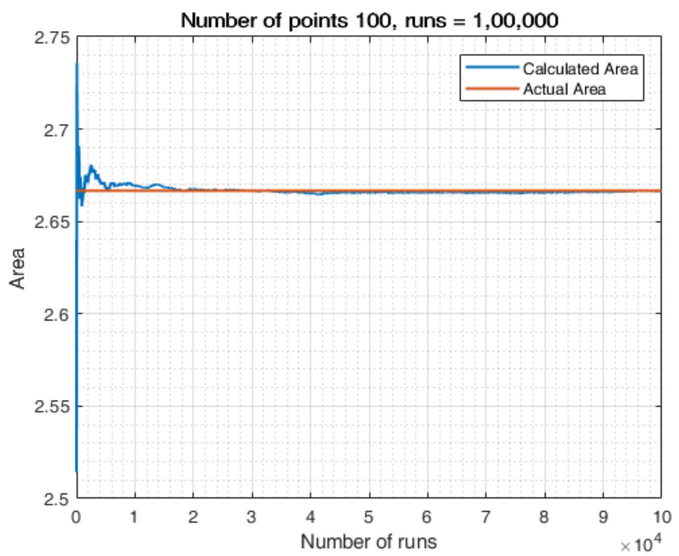

Figure 4.

As we can see from the figure3 and figure4, that the calculated value of the integral converges to the actual value if we increase the number of simulation but keeping the number of random numbers constant.

B. Calculate the value of pi:

Now we will use the Monte Carlo method to estimate the value of pi. To do this we firstly generate a certain number of points whose x and y values ranges from [0 1]. Now we will see that how many numbers of points are in the circle $x^2 + y^2 = 1$. Now by taking the ratio of number of points in the circle to total number of points generated give us the value of $\frac{\pi}{4}$.

(i) Here firstly we will use the first method (i.e. for single run increase the value of random numbers to be generated.)

The following graphs are obtained for different numbers of random numbers in a single run.

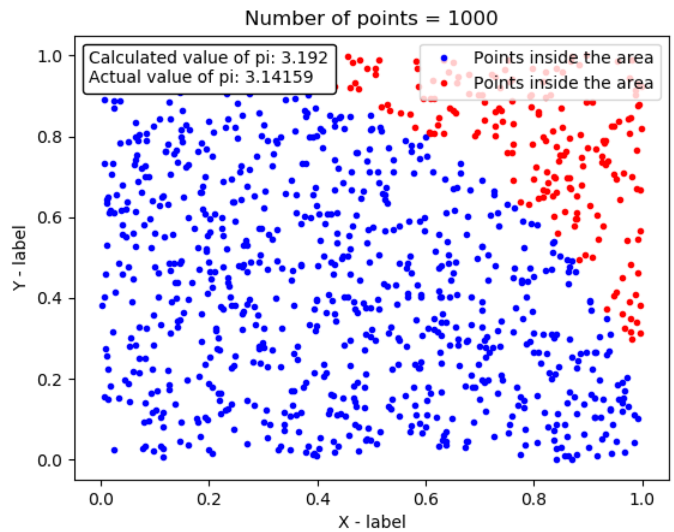

Figure 5.

Here as you can see that the obtained value of pi by generating 1000 random numbers is 3.192 but the actual value is 3.14159.

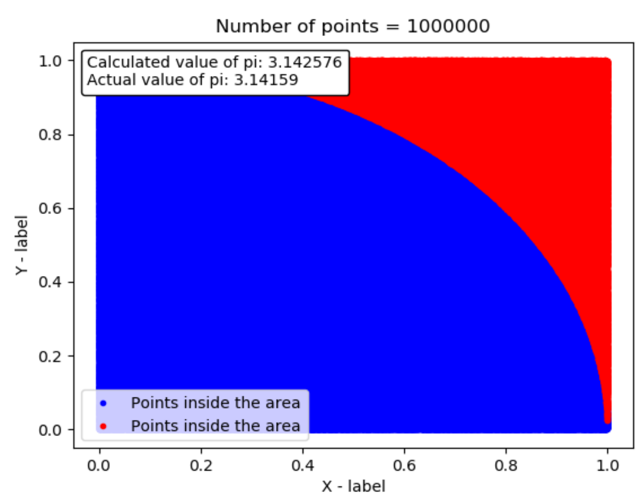

Figure 6.

Here the obtained value is improved a bit. The number of points generated is 10,00,000 and the value obtained is 3.142576 which has the precision up to 2 decimal places.

(ii) Now we will run the simulation many times and then take the average.

Figure 7.

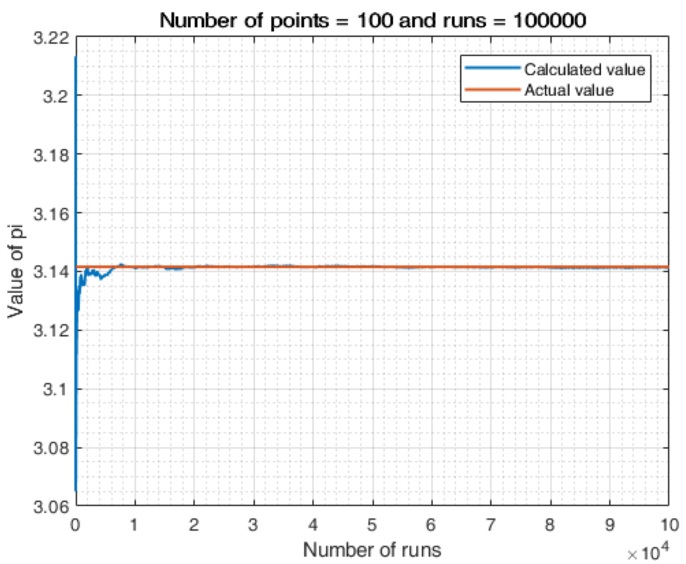

Figure 8.

As we can see from the figures 7 and 8 that as the number of simulations increases the calculated value of pi comes closer and closer to the actual value.

III. Problem Statement

Starting from uniformly distributed random numbers between 0 and 1, generate random numbers which are distributed as following:

A. Normal Distribution:

We will use Box-Muller algorithm to generate the normal distribution from uniformly distributed random numbers.

In this method we first generate two random variables which are uniformly distributed namely, y_1 *and* y_2. Now, the normal random variables x_1 *and* x_2 are generated as following:

$$x_1 = \cos(2\pi y_2)\sqrt{-2log(y_1)}$$

$$x_2 = \sin(2\pi y_2)\sqrt{-2log(y_1)}$$

So, the generated random variables x_1*and* x_2 are uniformly distributed.

The following figure is generated for number of points = 1,00,000 and bin size of 30:

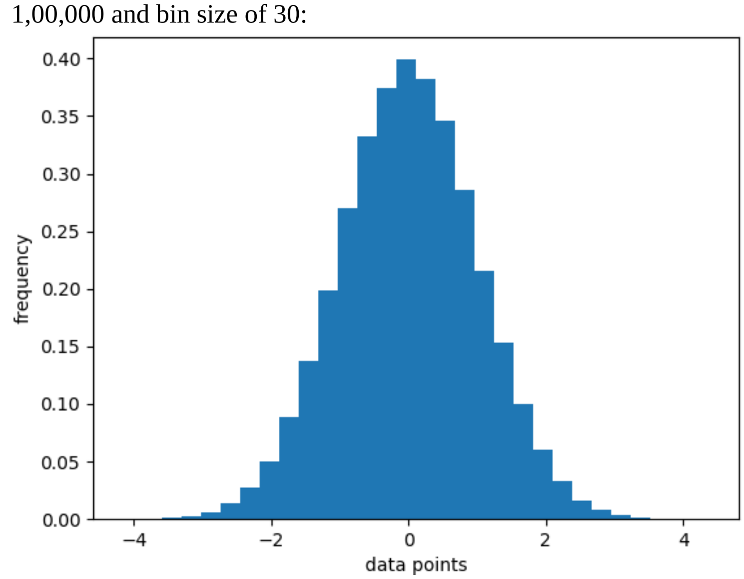

Figure 9.

The graphs of probability density function and cumulative density function, for n = 1,00,000 are as following:

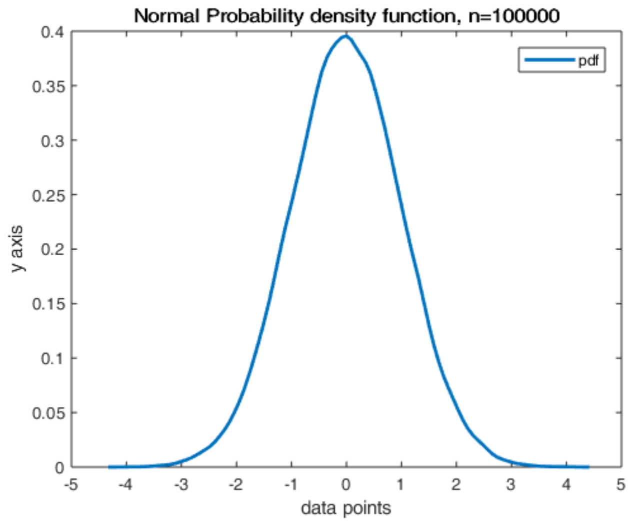

Figure 10.

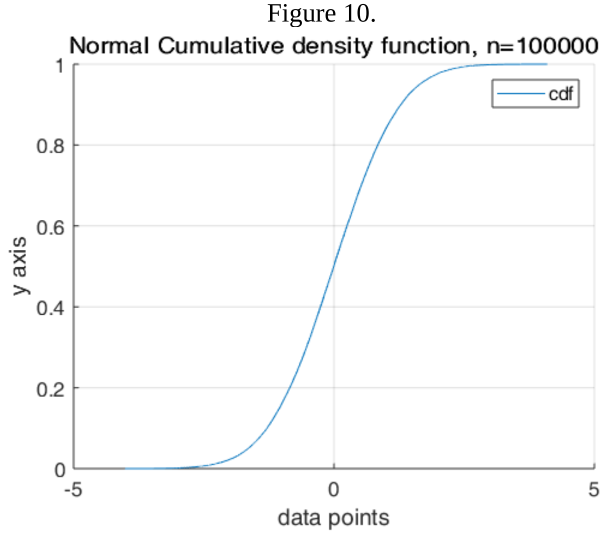

Figure 11.

B. Exponential Distribution:

We will use the method of inverse transform to generate the exponential distribution.

Here we have a function $F(x) = 1 - e^{-\lambda x}, x \geq 0$ then we have $X = -\frac{1}{\lambda}\ln(1-U)$, where U is the uniformly distributed random variable.

For $\lambda = 10$, n = 1,00,000 and bin size = 30, we have:

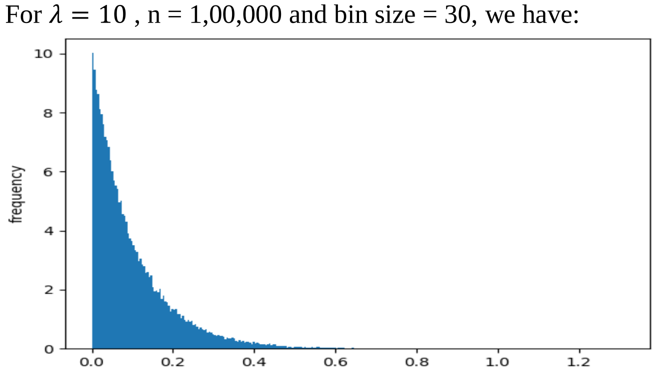

Figure 12.

The graphs of probability density function and cumulative density function, for n = 1,00,000 are as following:

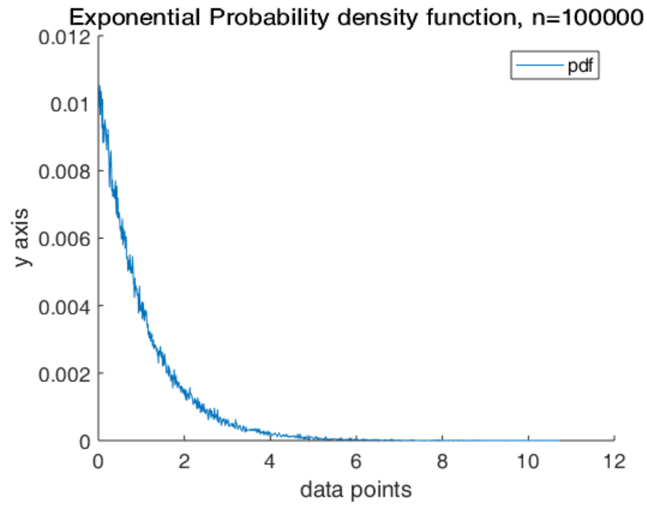

Figure 13.

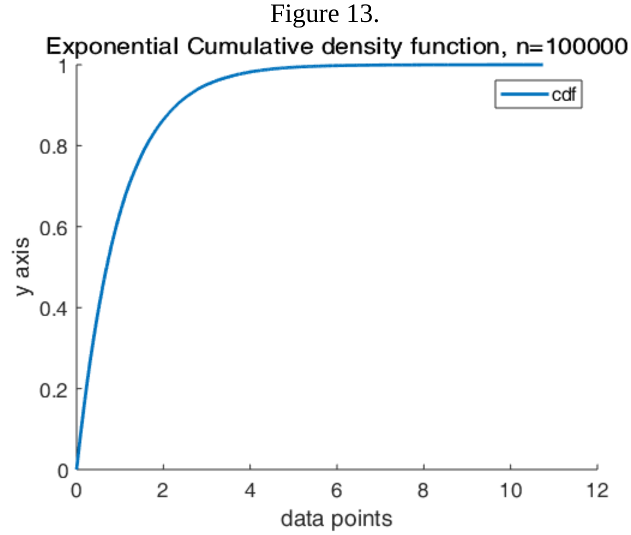

Figure 14.

IV. PROBLEM STATEMENT

Using Acceptance-Rejection Method obtain random numbers whose probability density function is given by $f(x) = 2\pi\sin(4\pi x)$ in the range 0 to 0.25. Generating sufficient number of values plot the histogram.

In the acceptance rejection method, we assume that $\rho(x) = 0$ outside the interval [a,b] and furthermore that $\rho(x)$ Is bounded above by c.

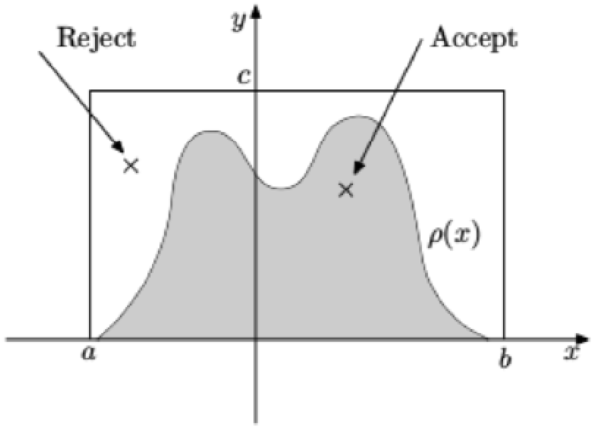

Figure 15.

Then we generate points (x_i, y_i) with x_i uniformly distributed in [a,b] and y_i uniformly distributed in [0,c]. If $y_i \leq \rho(x_i)$ then we accept the values of x_i. Now, the accepted values of x_i will have the probability density function $\rho(x)$.

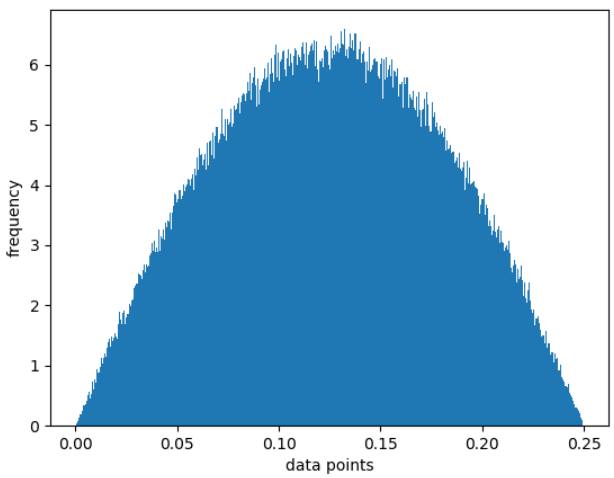

Figure 16.