

LAB 3

OBJECTIVE

To estimate the value of PI using Monte Carlo Simulation method

THEORY

Monte Carlo methods are a broad class of computational algorithms that rely on repeated random sampling to obtain numerical results. One of the basic examples of the Monte Carlo algorithm is the estimation of Pi.

The idea is to simulate random (x, y) points in a 2-D plane with domain as a square of side 1 unit. Imagine a circle inside the same domain with same diameter and inscribed into the square. We then calculate the ratio of number points that lie inside the circle and total number of generated points.

We know that area of the square is 1 unit sq, while that of circle is :

$$\pi \times \left(\frac{1}{2}\right)^2 = \frac{\pi}{4}$$

Now for a very large number of generated points,

$$\frac{\text{area of the circle}}{\text{area of the square}} = \frac{\text{no. of points generated inside the circle}}{\text{total no. of points generated}}$$

That is,

$$\pi = 4 \times \frac{\text{no. of points generated inside the circle}}{\text{no. of points generated inside the square}}$$

CODE

```
# Estimation of PI using Monte Carlo Simulation in python.
```

```
import numpy
```

```
import matplotlib.pyplot as plt
```

```
no_of_points=[10000,100000,1000000]
```

```
for N in no_of_points:
```

```
    x=numpy.random.uniform(low=-1,high=1, size=[N,1])
```

```
    y=numpy.random.uniform(low=-1,high=1, size=[N,1])
```

```
    inside_circle = x**2+y**2<1
```

```
    print("Out of {}, {} points inside the circle."  
          .format(N,numpy.sum(inside_circle)))
```

```
    approx_pi=4*numpy.sum(inside_circle)/N
```

```
    print('Approximated Value : {} Actual Value : {}\n'  
          .format(approx_pi,numpy.pi))
```

```
    x_in=x[inside_circle]
```

```
    y_in=y[inside_circle]
```

```
    plt.figure(figsize=[4,4])
```

```
    plt.scatter(x,y,s=1)
```

```
    plt.scatter(x_in,y_in,color='r',s=1)
```

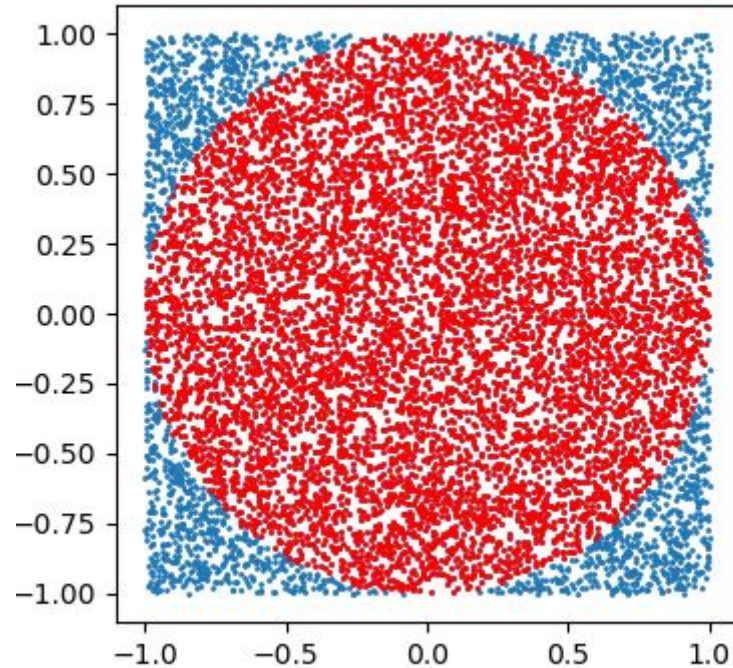
```
    plt.show()
```

```
    plt.close()
```

OUTPUT

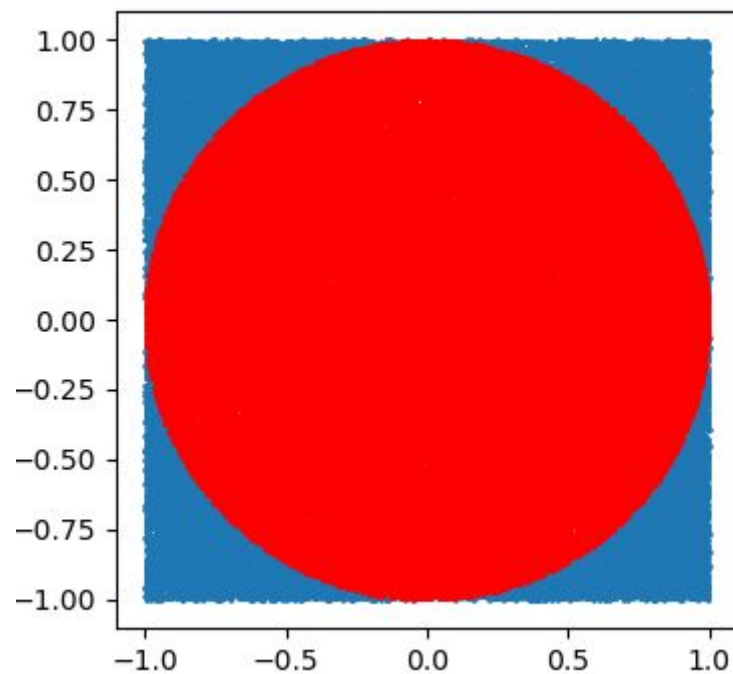
Out of 10000, 7881 points inside the circle.

Approximated Value : 3.1524 Actual Value : 3.141592653589793



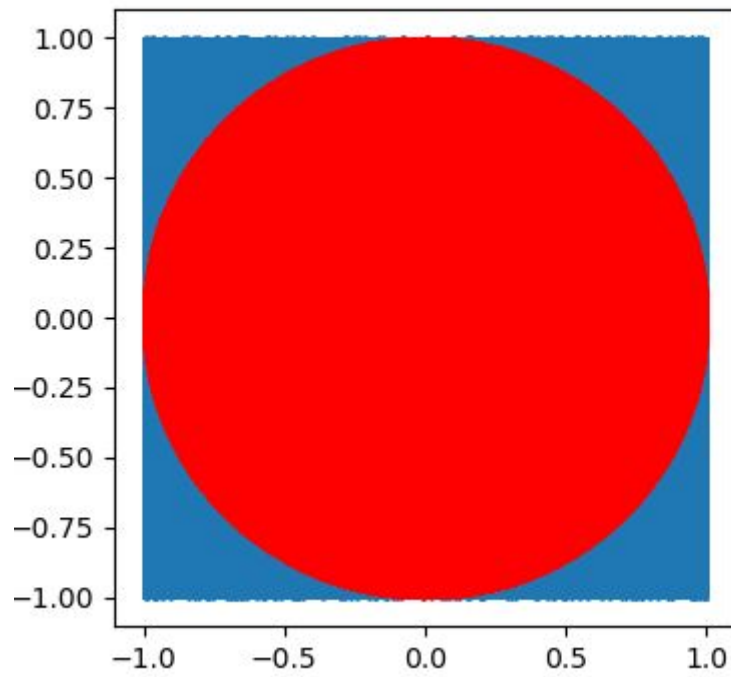
Out of 100000, 78798 points inside the circle.

Approximated Value : 3.15192 Actual Value : 3.141592653589793



Out of 1000000, 785610 points inside the circle.

Approximated Value : 3.14244 Actual Value : 3.141592653589793



CONCLUSION

In this lab, I got familiar with Monte Carlo estimation methods and estimated the value of pi in Python programming language.