AMS 595
FALL 2024

PROJECT-1

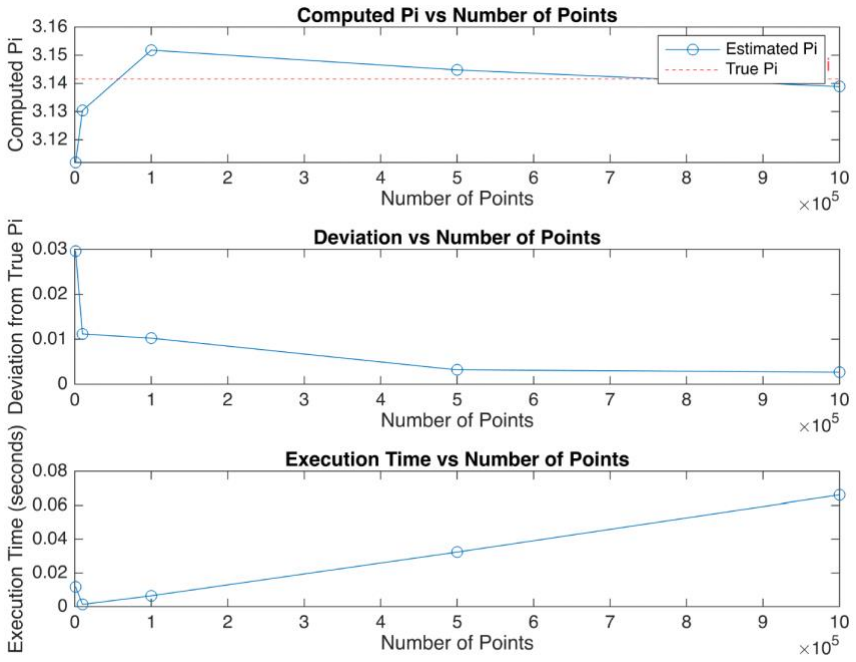
KULDEEPSINH RAJ
(SBU ID: 114980843)

In this project, we use the Monte Carlo method to estimate the value of π. To compute π using this method, we generate random points within a unit square [0,1] × [0,1] and check how many of those points fall inside a quarter circle with a radius of 1.
The Monte Carlo method can be summarized in the following steps:

1. Generate random points (x, y) within the unit square.
2. Determine if the point (x, y) lies inside the quarter circle by checking if $x^2+y^2 \leq 1$
3. The value of π is estimated using the formula:
    - $\pi \approx 4\times$ (Points Inside the Circle/Total Points inside the square)

Task 1: Compute π Using a For Loop
In this first task, the value of π is estimated using a for loop with a fixed number of random points. The number of points varies from 1,000 to 1,000,000, and the code counts how many of these points fall inside a quarter circle within a unit square. The ratio of points inside the circle to the total number of points is multiplied by 4 to estimate.
Key Results:

- A plot is generated showing:
    1. The computed value of π as the number of points increases.
    2. The deviation from the true value of π.
    3. The computational cost as the number of points increases.



- For 1,000 points, the estimated value of π is 3.112, with a deviation of 0.0296 from the true value.
- As the number of points increases, the precision improves. For 1,000,000 points, the estimated value is 3.1389, with a deviation of 0.0027.
- As expected, increasing the number of points improves the accuracy of the estimate.
- The execution time also increases with more points, reflecting the computational cost of generating and checking random points.

Task 2: Compute π\piπ Using a While Loop

In this task, the computation of π is modified to use a while loop that runs until a desired level of precision is achieved. The precision is specified in terms of significant figures (e.g., 2, 3, or 4). The number of iterations increases with higher precision, showing the relationship between precision and computational cost.

Result:

```
Precision level: 2 significant figures
Computed Pi = 3.1200000000
Number of iterations = 5001
Execution time = 0.00279 seconds
Precision level: 3 significant figures
Computed Pi = 3.1430000000
Number of iterations = 5001
Execution time = 0.00206 seconds
Precision level: 4 significant figures
Computed Pi = 3.1466000000
Number of iterations = 5006
Execution time = 0.00235 seconds
```

Task 3: User-Defined Precision

In this task, the algorithm from Task 2 is modified into a function where the user can input the desired precision. Additionally, the function generates a graphical visualization of the Monte Carlo simulation. Random points are plotted, with points inside the quarter circle colored green and points outside colored red. The graphical outputs illustrate that as the required precision increases, the estimate of π becomes more accurate.