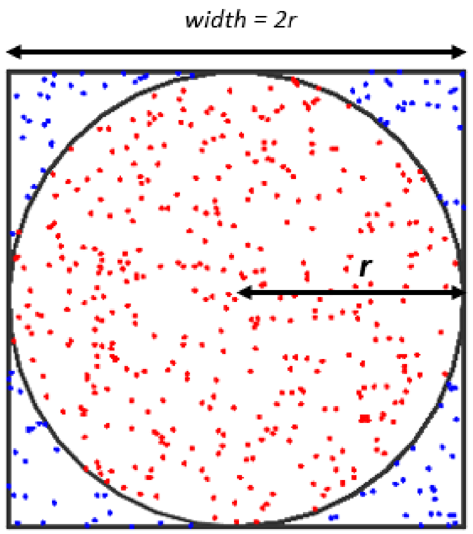
Important Note

You must write your code independently. Do not copy the code from any source. To receive full credit, your code must be well-documented with comments. Please submit your code to GitHub and upload a report summarizing your results to Brightspace.

Background



The value of π can be estimated using a Monte Carlo algorithm by leveraging the relationship between the probability that randomly chosen points in the $x - y$ plane fall within a specific region and the area of that region. Specifically, if we randomly select $(x, y) \in [0, 1] \times [0, 1]$, these points will always lie within a unit square. Some of these points may also lie within a quarter of a circle with radius 1, centered at the origin. The area of this quarter circle is $\pi/4$, which represents the probability that a given random point will fall inside the circle. We can use this approach to compute π by generating a large number of random points and counting how many fall inside the circle compared to the total number of points generated (i.e., those that fall within the square). The ratio of these two quantities will approximate $\pi/4$. As the number of random points increases, the precision of our approximation improves.

Submission Instructions

Due Date

Sunday, 11:59 PM EST, September 14th, 2025

For this project you will need to submit the following:

- Upload your completed MATLAB code to your GitHub repository, ensuring it is well-commented for clarity and readability. Please add a `README` file to your repository to explains how they can use it.
- Submit a short report as a PDF file to Brightspace. This report should provide a concise summary of your algorithm, discuss the performance of your code (including any relevant plots), and include the content of your “user guide” as well as a screenshot of your final product.

Tasks

Here you will write a MATLAB program that computes π using the approach described above via two DIFFERENT flow control mechanisms. In particular, you are asked to do the following:

1. **30 Points.** Write a MATLAB program that uses a `for` loop to compute π with a fixed number of random points. Plot the computed value of π and its deviation from the true value as the number of points increases. Measure the execution time of your code for different point counts, and create a plot comparing the precision to the computational cost.
2. **60 Points.** Write a modified version of your program that uses a `while` loop to compute π to a specified level of precision (e.g., 2 significant figures, 3 significant figures, etc.). Record the number of iterations required to achieve each level of precision. Importantly, your method for determining precision should **NOT** rely on the true value of π.
3. **30 Points.** Modify the code in (2) to be a **function** that includes all the following features:
    - The function should take a user-defined level of precision as input.

- A graphical display should plot the random points as they are generated, with points inside the circle plotted in a different color than those outside the circle.
- The final computed value of π, written out to the user-specified precision, should be both displayed in the command window and printed on the plot.
- The computed value of π should be returned by the function.

Grading Rubric

- 120 points for correctness of code.
- 30 points for documentation, report, coding practices (commenting, ease of reading, etc.), correct use of Git.
  - 1 report in a human readable format like PDF or DOC/DOCX.
  - 1 combined plain text code file that executes.