

# CSE 13S: Assignment 1 Design Doc

Johnny Li

January 20, 2023

## 1 Introduction

We'll be using gnuplot to create a bash script that plots tens of thousands of points according to the monte-carlo.dat output from the provided monte-carlo.c program in order to familiarize ourselves with Unix systems and fundamental C syntax. After each iteration with a random point, the aforementioned program outputs the Monte Carlo estimation for  $\pi$ . This will be done using a variety of programs, including the previously mentioned gnuplot and Bash (preinstalled), as well as a number of supplied files from the CSE 13S Section 1 Resources GitLab repository. (And various other pieces of software like "make" which are essential for the program to run but require manual installation because... Linux)

## 2 gnuplot

gnuplot, a powerful command-line and graphical user interface graphing utility, is widely utilized to plot an array of 2-dimensional and 3-dimensional functions across a plethora of platforms, including but not limited to Windows, MacOS, Linux, and even the antiquated OS/2 operating system. The versatile nature of this program allows for data to be plotted from an array of sources, including data files, mathematical functions, and equations, and further enables the output of the plots in a plethora of formats such as PNG, PDF, and Postscript. Furthermore, gnuplot's ability to seamlessly integrate with other programs, such as LaTeX, has contributed to its widespread usage in scientific papers and presentations.

Unlike Bash, gnuplot is not typically preinstalled on systems, requiring users to manually install the program. In our case, the program was installed through the terminal with the command "sudo apt install gnuplot", which is a quick and easy process. Once installed, the program can be run in the terminal by simply typing "gnuplot" into the command-line.

## 3 LaTeX

LaTeX is a powerful and widely utilized typesetting software that is employed in the composition of various scientific documents, such as journal articles, research papers, dissertations, and books. This is owing to the program's capability to seamlessly integrate mathematical and scientific notation with regular text, thereby allowing for the creation of documents that effectively convey complex ideas and information.

LaTeX is a markup language, meaning that in addition to plain text, it utilizes special commands to format and structure documents. In this regard, its syntax is akin to a combination of HTML and a traditional text editor, providing users with a robust set of tools for document creation.

## 4 Tasks

### 4.1 C Syntax

The C programming language, developed during the early 1970s, has cemented its position as one of the most widely-used and enduring programming languages in the annals of computer science. Its lasting popularity can

be attributed to a plethora of factors, chief among them being its simple yet powerful syntax. Despite the passage of time, the syntax of C has remained remarkably consistent, with only minimal modifications made to keep pace with the ever-evolving landscape of technology.

One of the defining characteristics of C's syntax is its adherence to a minimalistic and low-level approach. This approach allows for the crafting of highly efficient and performant code, as the syntax is designed to closely resemble the underlying machine architecture. This allows programmers to exert a high degree of control over low-level operations such as input/output and memory allocation, resulting in streamlined and optimized code.

C's syntax is also known for its support of structured programming, an approach to code organization that involves the use of reusable blocks of code known as functions. Functions are defined by a function header, which includes the function's name, return code, and parameters. This feature enables the creation of modular and maintainable code, making it easier to test and debug.

In addition to its support of structured programming, C's syntax also accommodates a wide range of data types, including integers, floating-point numbers, and characters. It also boasts a plethora of operators, such as mathematical operators and comparison operators, that allow for the manipulation of data and control structures such as if-else statements, for and while loops, and function calls.

Another seminal feature of C's syntax is the use of pointers. Pointers allow for the manipulation of memory addresses directly, and are a fundamental concept in C programming. They enable the creation of dynamic data structures such as linked lists and trees, further expanding the capabilities and flexibility of the language.

## 4.2 Monte Carlo Method

The Monte Carlo method is a statistical approach that uses random sampling to approximate  $\pi$ . To do this, the program plots random points on a 2D graph and determines the ratio of points that fall within the radius of the circle to the total number of points generated. A large amount of points are used as the more points tested, the more accurate the estimation. It scatters points uniformly across a square and measures the the number of points within the quadrant and the other points. In order to get random points, a pseudo-random number generator is incorporated since true randomness cannot be achieved solely with computer code. The provided C program prints it's estimation for  $\pi$  using the Monte Carlo method after each point it plots. gnuplot will be used to produce said plots and everything will be automated by the Bash script we'll create.

## 4.3 Bash Script

All the necessary C gnuplot programs and files have already been provided to us through the CSE 13S Resources GitLab repository. Our goal is to write a Bash script that automates the creation of two graphs. One graph will display random dots within a quarter circle, the dots within the circle will be blue while those outside will be red. The C program will calculate and output a monte-carlo.dat file which will contain five columns displaying the iteration, pi estimate, x-coordinate, y-coordinate, and a 0 or 1 to signify whether the respective dot is within or outside the circle.

To create the first graph, the script creates a file with only the third, fourth, and fifth columns without the first row. It then separates the points based on their distance from the circle with 0 going into one file and 1 going into another file. From there, the files are plotted with gnuplot into a single graph. The dots are colorized during their plotting.

As for the second error estimation graph, the script does the same thing again but with the initial two columns instead and subtracts  $\pi$  from the value of the second column which contains the C program's estimation for  $\pi$ . It does this four times to produce four independent results and file which are graphed into a line graph. The y-axis showing the absolute error with a range of  $[-1, 1]$  and the x-axis showing the respective iterations.