Assignment 1 WRITEUP.pdf

Vincent Liu

January 23, 2023

1 Introduction

Assignment 1, titled "Getting Acquainted with UNIX and C", introduces students to GNUplot, UNIX and C. GNUplot is a command line tool used to plot graphs of functions or data. It can be used to display all sorts of data, whether it relates to science, business, or any other fields where statistics is involved. With the use of provided source files, students are able to produce plots/graphs of the Monte Carlo method of estimation. UNIX is a computer operating system that was initially developed in 1970 and has been in use since. C is a general purpose programming language developed in the 1970s and has also been in use since, mostly due to its influence and wide variety of applications.

2 Monte Carlo Method of Estimation

The Monte Carlo method is a computer algorithm that utilizes random sampling to estimate a numerical value. In our case, we make use of the Monte Carlo method to approximate the value of pi. We do so by running the program and iterating 1000 times, which gives us a good amount of samples. Additionally, we use the command line as well as GNUplot to generate a graph in order to visualize how well the Monte Carlo method works when it comes to approximating a value.

3 What I learned

What I learned from this assignment was that I am able to make use of the UNIX command line and GNUplot to create interesting and useful graphs. These graphs can help visualize certain things such as large data files and/or the efficiency of a program. I was able to learn how to plot lines and points, as well as how to color them. Furthermore, I learned to understand the basics of Makefiles, their purpose, and how to use them when compiling other files.

3.1 UNIX commands used

- ./plot.sh: Used to execute/run plot.sh and generate a graph.
- vi: Vi is a text editor and I utilized this tool for assignment 1 when modifying/reading monte_carlo.c, Makefile, and plot.sh.
- ls: lists all files in current directory
- cd: changes directory from whichever folder user is in
- '>': Used to push the output of a program into another program. Was used to push the output of monte_carlo executable to GNUplot

4 Generating plots

Generating plots with GNUplot takes multiple lines of code and cannot be done with a single line. Parameters and preferences must be set before running 'plot' and giving data to GNUplot to generate a graph. For example, before running 'plot ...', one must set the name of the output file for the graph, as well as the file type (I used .pdf). Additionally, one can also set the preferred size of points on the graph, labels of of the x and y axis, and whether or not the 'key' should be shown on the top right corner (key: A reference which reflects the data displayed).

5 Files

5.1 `Makefile`

Makefile contains instructions for the terminal to create and delete certain files. The first line on this file is 'make clean', which creates a clean environment to work from. What this means is that it clears all objects and executable files from the directory so that the compiler can correctly compile files. It also has make all,

5.2 `plot.sh`

Plot.sh makes use of the Makefile in the same directory to first clears the directory of all executable files, then create an executable file called 'monte_carlo' as well as a file called 'monte_carlo.o'. Plot.sh takes the output of this script and places it into a temporary data file (.dat) with the use of the '¿' symbol. A data file is a file in which values are separated by whitespaces and organized into columns. In our case, the execuatble creates data that is separated into 5 separate columns, consisting of iteration, pi estimate, x and y coordinates, and a 0 or 1 if the coordinates fell within a circle. Plot.sh then runs gnuplot, setting up the graph according to the specified 'set' commands. It then uses the 'plot' command to hand over information from the data file to gnuplot, as well as set options for the graph such as lines or points, and which colors to use. The generated graph is then exported as a pdf file into the same directory.

5.3 `monte_carlo.c`

Monte_carlo.c is a C language file that contains a program that replicates the Monte Carlo method of estimation. It also contains information on how to use certain functions within it from the plot.sh file.

- '-r' : Sets a seed from which the Monte Carlo executable file begins estimating the value of pi from
- '-n' : Set the number of iterations the file would run. Essentially sets the number of points or guesses the program would make
- 'help' : Displayed program help and usage

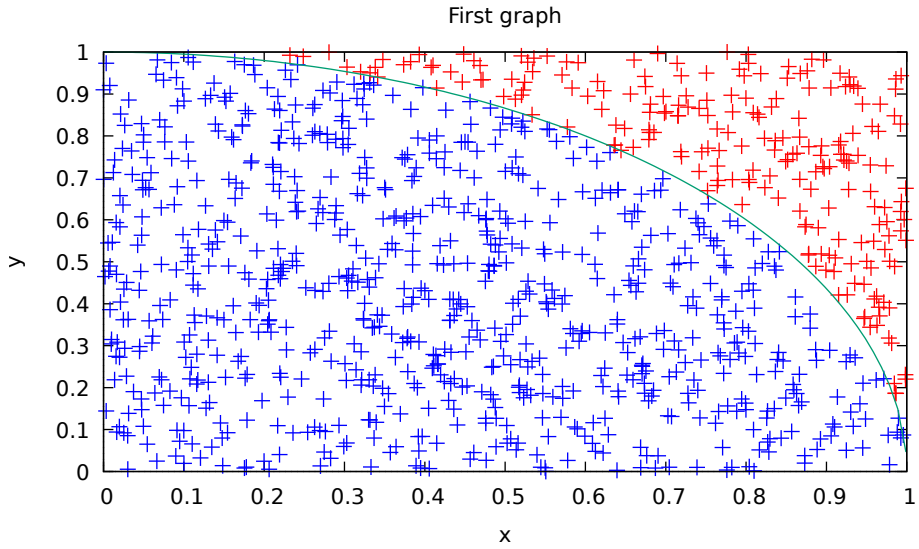

Figure 1: Points are plotted at random and colored according to distance

In the visual above we are able to see how the Monte Carlo method is used to plot points within a circle. There are 1000 randomly plotted points, most of which fall within the a circle and are colored blue. Those that fall within the circle have a distance that is less than or equal to 1 unit, as can be seen on the graph. Of course, this means that the red points on the visual which are outside the circle have a distance greater than 1.

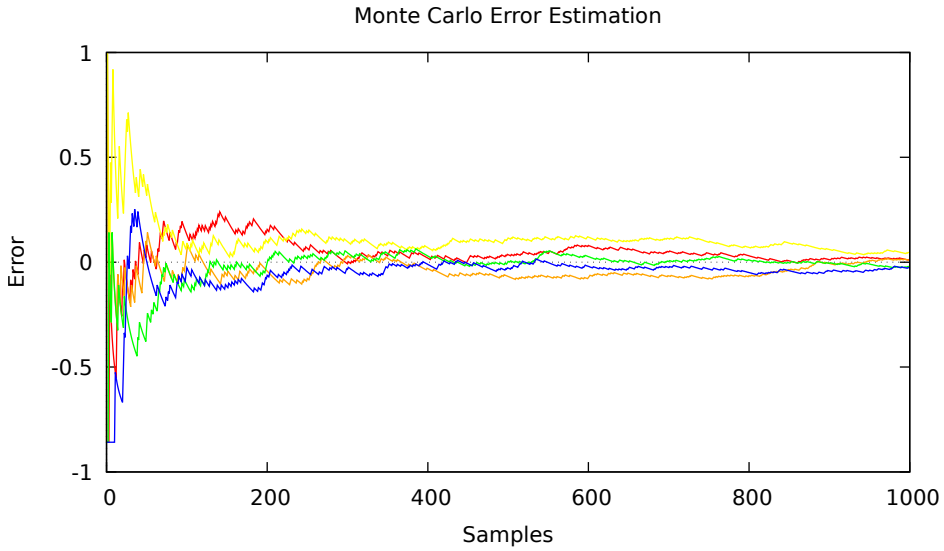

Figure 2: Monte Carlo method of estimation

In figure 2 we are able to see how accurate the Monte Carlo Method is to guessing the value of pi when it produces 1000 sample points. This graph exhibits the error difference between our value of pi and the program's estimated value of pi over the course of its 1000 iterations. Each color on the graph represents a different seed, or 'origin point', from which the Monte Carlo program begin its estimating process. All data was retrieved using seeds: 1, 2, 3, 4, 5 on monte_carlo.c. It should be noted that this estimation process is grueling and inefficient. Additionally, it can be argued that the process will take an infinite amount of time to accurately guess the value of pi, as pi itself is an