Design Document - Assignment 1

Hsiang Yun Lu

January 23, 2023

1 Purpose of Program

The purpose of this program is to produce plots using `gnuplot` based on the `C` program provided in the class resources repository.

The provided `C` program first produces uniformly scattered points in a square with an inscribed quadrant and then measures the number of points that fall within the quadrant and out of the quadrant. The ratio of the number of points in the quadrant to the number of points in the square can be used to estimate π. After each random point, it tests, the `C` program will output the current iteration, the estimated value of π of the current point, its x and y coordinates, and a 1 or 0 value indicating if it is in the circle, where 1 means it is inside the circle.

The first plot (*Figure 2*) will demonstrate the uniformly scattered points in a square and inscribed quadrant. The points in the quadrant have a distance less than or equal to 1 and are blue. And the points that are a part of the square but not the circle will be red.

The second plot (*Figure 3*) will show the difference between the estimated π and "true" π as iteration increases under different seeds for the random number generator.

2 Files To Be Included in Directory *asgn1*

- `plot.sh`
  - Bash script that produces the Monte Carlo method plots
- `monte-carlo.c`
  - A `C` program that implements Monte Carlo method to estimate π
  - Used in the `plot.sh` bash script to produce plots
- `Makefile`

  - The file that directs the compilation process of the Monte Carlo program

- `README.md`
  - This file describes how to use the bash script and `Makefile`
- `DESIGN.pdf`
  - This file describes the design and design process for the bash script
  - Include pseudocode
- `WRITEUP.pdf`
  - Include the plots that are produced using the bash script
  - Include a discussion on the UNIX commands used to produce each plot and the reasons to use them

3 Design Process

I took inspiration from the example script producing the plot of $y = sin(x)$. The idea is first to generate files containing data required for both plots, and write the *here-document* sent to `gnuplot` to produce the plots based on the data files.

Therefore, there will be four steps in my design: (1) data generation of *Figure 2* (2) data generation of *Figure 3* (3) plot *Figure 2* using `gnuplot` (4) plot *Figure 3* using `gnuplot`.

The data needed to plot *Figure 2* will be stored in one file and separated into two files, while data needed to plot *Figure 3* will be in six files. Because The different lines with different colors in *Figure 3* represent the results of different seeds for the random number generator.

4 Design/Structure

REBUILD $monte\_carlo$

./$monte\_carlo$ SET number of points for estimation | trim first 2 rows > file(a)

IF column 5 in file(a) == 1
    PRINT column 3, column 4 > file(b)
ELSE
    PRINT column 3, column 4 > file(c)
END IF

SET `pi` to a constant

FOR $i = 0$ to 5
    SET `num` as a random seed
    SET ./*monte_carlo* SET number of points for estimation and SET random seed `num` | trim first 2 rows | OBTAIN column 1, (`pi` - column 2) > file(i)
END FOR

`gnuplot`
    SET plot as pdf
    SET output name for *Figure 2*
    SET plot size
    SET formula of the quadrant
    PLOT file(b) as blue points
    PLOT file(c) as red points
    PLOT the line of the quadrant
END `gnuplot`

`gnuplot`
    SET plot as pdf
    SET output name for *Figure 3*
    SET range of x axis
    SET range of y axis
    SET plot title
    SET y lable
    PLOT file(0)-(5) with different colors
END `gnuplot`

5 Credit

- The general concept and format of the design took reference from the `plot.sh` example in the `asgn1.pdf`
- Some ideas, like how to use `awk` and general idea of making *Figure 3*, came from reading people's discussion on Discord
- Official Gnuplot online documentation Version 5.4