

Assignment 1 Design.pdf

Vincent Liu

Purpose

This program graphs a Monte Carlo estimation of π onto a graph with the use of gnuplot and shell. The output of the Monte Carlo C program (`monte_carlo.c`) is given as a data (`.dat`) file to the `.shell (.sh)` script. The shell script then takes the values of certain columns from within the data file to be used as x and y values on a graph via the gnuplot program. The graph created by the gnuplot program via `plot.sh` contains the C program's estimated value of π over a certain number of iterations. There are two graphs created by `plot.sh`, both of which are in pdf format. One involves random coordinates in which certain points are colored depending on whether or not they fall within the range of a circle. The other graph involves an error estimation which allows for visualization of how the Monte Carlo C program gradually gets closer to π over hundreds of iterations. All points get their x and y coordinates from columns 3 and 4 of the temporary data file, respectively. If a point falls within the square, but not the circle, it will be colored red. If a point falls within the square and the circle, it will be colored blue. All points on graph 2 get their x and y values from columns 1 and 2 of the temporary data file, respectively.

Files to be included in directory "asgn1":

1. `monte_carlo.c`
 - Source file that contains `main()` and the majority of the content that allows the code to run
2. `plot.sh`
 - Passes output data from `monte_carlo.c` towards gnuplot in order to create a graph as a pdf
3. `Makefile`
 - Describes how to properly build the program
4. `README.md`
 - Text file in Markdown format that describes how to build and run the program.
5. `DESIGN.pdf`
 - Describes the design for the program thoroughly with pseudocode and visualizations
6. `WRITEUP.pdf`
 - Describes what the program does, and gives insight on the results found about the method's efficiency as well as other details.

Pseudocode/Structure:

1. Creates a “clean” state to work from and creates the executable program “monte_carlo”
2. Runs the monte_carlo program 1000 iterations with seed number 1
3. Runs the monte_carlo program 1000 iterations with seed number 2
4. Runs the monte_carlo program 1000 iterations with seed number 3
5. Runs the monte_carlo program 1000 iterations with seed number 4
6. Runs the monte_carlo program 1000 iterations with seed number 5

7. Calls gnuplot
8. Sets terminal output as pdf
9. Sets name of output file as “monte_carlo_.pdf”
10. Sets title for first graph as “First graph”
11. Sets x axis title “x”
12. Sets y axis title “y”
13. Sets zeroaxis
14. Sets size of points to 1
15. Turns of key/legend (usually present in top right corner)
16. Sets a palette so that colors can be correlated to certain values. 0 is set to red and 1 is set to blue
17. Unset colorbox. This removes the color heatmap usually added to the right of a graphic that utilizes ‘palette’
18. Plots monte_carlo_0.dat file using columns 3,4,5 with points and palette to be applied on values of column 5

19. Sets title of second graph to “Monte Carlo Error Estimation”
20. Sets x axis label to "Samples"
21. Sets y axis label to "Error"
22. Plots monte_carlo_0.dat, monte_carlo_1.dat, monte_carlo_2.dat, monte_carlo_3.dat, monte_carlo_4.dat with lines. Each line is set to a different color. Each lines uses column 1 as x values and column 2 (observed value) subtracted from the actual value of pi as y values.
23. END

Credit/References/Citation:

http://www.gnuplot.info/docs_4.2/node215.html

-Gave insight on how to use 'palette' for both of my graphs. Allowed for coloring of points on 'First graph'

<https://livebook.manning.com/book/gnuplot-in-action-second-edition/chapter-9/36>

-Used to learn how to implement 'rgb' for the second graph. Used to color line for the Monte Carlo Error Estimation.

<https://stackoverflow.com/questions/24421947/how-to-remove-palette-colors-heatmap>

-Used to remove the heatmap on the right hand side of a graphic that utilizes 'palette' to color

-Attended Section on 1/19 (Thursday). Got general understanding on how the first few lines of the plot.sh worked.