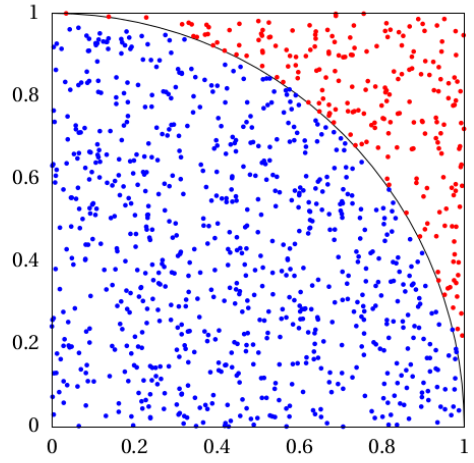
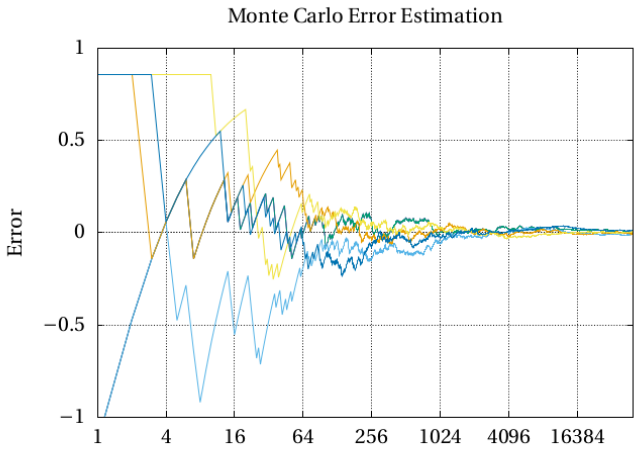
Assignment 1 DESIGN.pdf

Description of Program:

This program is a shell script that will take the output of a provided file, monte_carlo.c, and will use gnuplot to create two graphs that visually displays the Monte Carlo method of calculating pi. The first graph visually depicts the Monte Carlo method, with a quarter of a unit circle inscribed in a square. Points in both the circle and square are blue, while those just in the square are red. The second graph shows the error in the Monte Carlo calculation, with the x-axis being the number of iterations and the y-axis being the difference between the Monte Carlo calculation and the true value.





The plots shown above are what I am trying to recreate with my shell script. The code really is broken down into 3 parts. The first part is formatting, compiling, and running the monte_carlo.c program and placing the output in a .dat file. The second part is to write code that will recreate the first graph using gnuplot, creating a scatter plot of all the points generated by monte_carlo.c and then color coding them based on whether or not they fall into the unit circle or not. The third part would be recreating the second graph, where I have to plot the error of pi calculation from

monte_carlo.c against the number of iterations. This graph will be a line graph, and I have to do this for multiple data sets, in this case, 5.

Deliverables:

1. **plot.sh**
    - This is the shell file that I will be writing that, when given the output data from monte_carlo.c, should produce pdfs containing plots similar to the ones shown above.
2. **monte_carlo.c**
    - This file is provided and implements the Monte Carlo program, outputting data including the iteration number, current pi estimate, coordinates of the generated points, and whether or not the points are in or out of the circle.
3. **Makefile**
    - This file is provided and compiles the Monte Carlo program.
    - It also clears the files previously created by the compile process using “make clean”.
4. **README.md**
    - This file should be in Markdown syntax and should describe how to use your script and Makefile.
5. **WRITEUP.pdf**
    - This file is a writeup which should include the plots created by our bash script, as well as writing about the commands used in our shell file and why they were chosen and used.
6. **DESIGN.pdf**

- This file is what you are reading right now.
- It describes the design process for this assignment accompanied with pseudocode.
- It should be written with enough detail such that another programmer would be able to replicate my implementation.

Pseudocode:

```
format files
run monte_carlo and transfer output into .dat file
gnuplot << END
set output file as pdf
draw circle with radius 1 centered at origin
for point in monte_carlo.dat:
        plot point with coordinate data
        if inCircle == True:
                color = blue
        else:
                color = red
clear plot 1
run monte_carlo and transfer output into .dat file x5
set x-axis logarithmic scale
set color
error = predicted_pi - actual_pi
plot error vs iteration_#
```

set color_2

plot error vs iteration_#

set color_3

plot error vs iteration_#

set color_4

plot error vs iteration_#

set color_5

plot error vs iteration_#

END

Error Handling:

- The program will not run if monte_carlo.c is not formatted.
- This can be remedied by running the command ‘make format’ in the terminal

Credit:

- I attended John Yu’s discussion section on Jan. 19, 2023. It gave me some guidelines on how I should start my plot.sh file as well as how to do the coloring for the first plot using the palette command.
- I used the gnuplot manual as a reference to learn various features of gnuplot and how to implement them.
    - I consulted the gnuplot 5.0 manual for these following commands:
        - set xrange
        - set yrange
        - set xtics

        - set ytics
        - set size
        - set pointsize
        - set logscale
        - set grid
        - plot using
        - plot with
        - linestyle
- I used the demos on the gnuplot website to learn how to implement certain colors on my plots.
    - ‘pm3d colors’ demo was used to set the red/blue color palette for the first plot
    - ‘mixing styles’ demo was used to learn how to plot multiple lines on the same plot (for the second plot)