Nicolas Corfmat

CSE13S, Winter 2023

ncorfmat@ucsc.edu

Exploring π Using The Monte Carlo Method

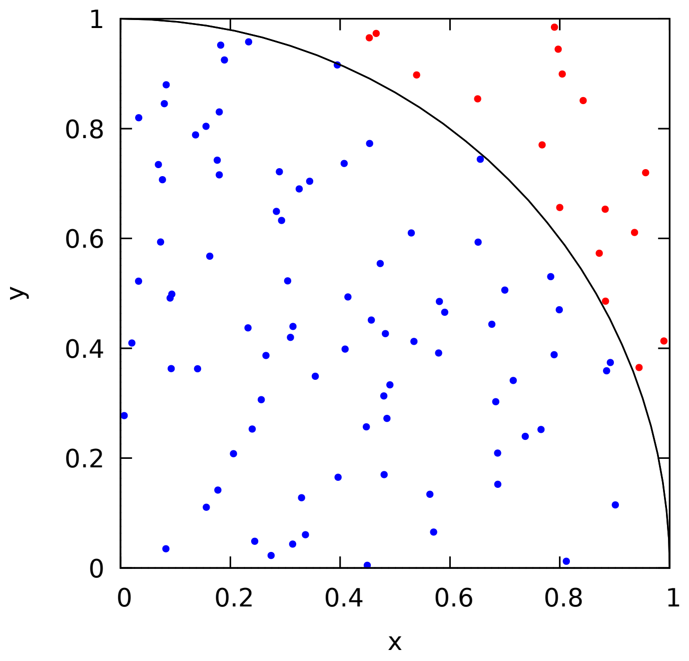
Getting Acquainted with UNIX and C

Description of Program

This bash script produces plots of the given C program, which generates a predetermined amount of random two-dimensional points and graphs them. One plot demonstrates the number of points that fall within a quarter circle inscribed within a square and denotes them with the color blue, and red otherwise. The other plot displays the error values of the estimated π value as the number of iterations increases.

Visualizing the Monte Carlo Method

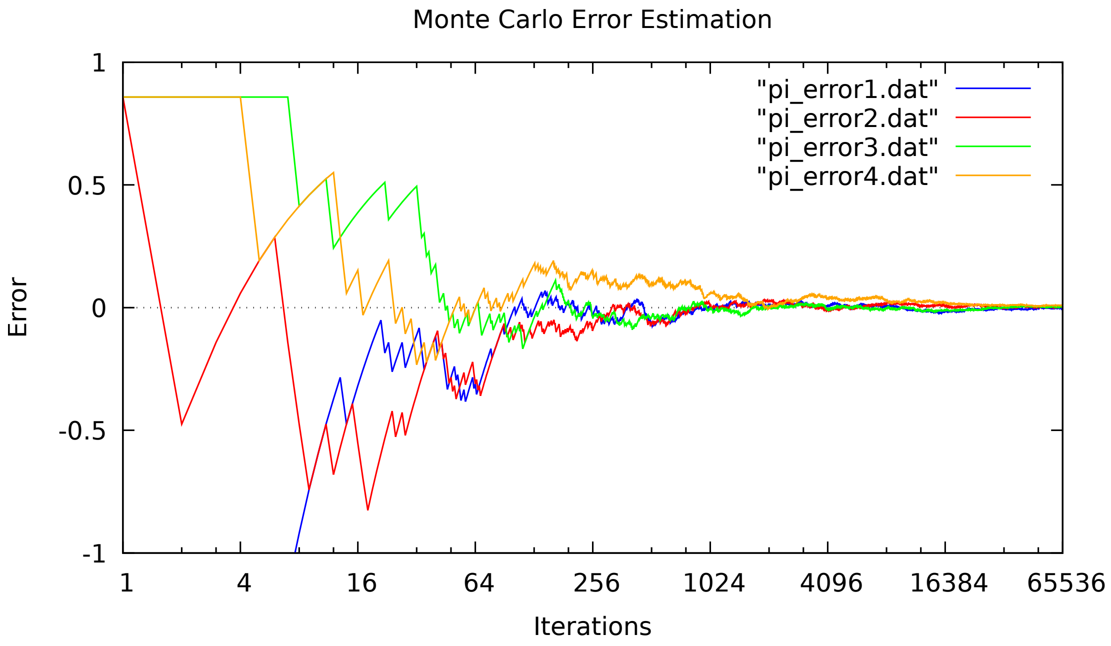
For those unfamiliar with the Monte Carlo method, IBM defines it as "a mathematical technique, which is used to estimate the possible outcomes of an uncertain event." (IBM) Essentially, it relies on utilizing a large number of randomly generated data points and predicting an outcome based on long-term pattern(s). In this case, we'll be estimating the value of π by plotting a predetermined sample of random, two-dimensional points, ranging from 0 to 1 (not including 1), and graphing them on a 1 by 1 square grid containing an inscribed quarter circle of radius 1). The program produces the first plot, highlighting points inside the circle as blue and red otherwise.



The resulting graph reveals exactly what we had outlined—a plot of randomly generated and color-coded points. In order to create a data file consisting of solely pair coordinates, I used the UNIX *tail* command to preserve the last *n* number of lines. Essentially, I wanted to extract every line except the first, which were just labels indicating the kind of information listed in each column. Now that I had all the numerical data under one file, I utilized the *awk* command to copy only the columns containing the X and Y coordinates and writing them into two separate files, depending on whether they fell within the circle or not. This required checking if the point was within the circle using an if-statement.

Monte Carlo Error Estimation

The data produced from running "monte_carlo.c" consists of an estimated π values for each iteration. For the graph of the inscribed circle, we used a sample size of 100 points as this is what is considered statistically significant. However, in order to generate a precise value of π, we're going to need to increase our sample size substantially. By plotting the error value of π (difference between estimated π and actual π value), we will be able to observe changes to our estimates.



As shown from this graph, the error value fluctuates around 0. It appears though, that as the number of iterations increases, the lines for all data sets gets increasingly closer to 0, implying that the estimated π values do seem to approach its true value of π. If we look at the blue line, it might appear as if the data is incomplete for roughly the first 10 iterations. Perhaps the data file lacks these iterations? No. In reality, when the difference between the estimated π and real π values was calculated, the magnitude was large enough that it went beyond the given vertical range. This further solidifies the observation that the beginning iterations yield largely inaccurate π estimations—a trend seen across all lines on the graph.

Regarding the UNIX commands used to produce this plot, I again used *awk* to retrieve the estimated π values from the data file, and then subtracting the value of π from it (accurate to 38 decimal places). Additionally, the *tail* command ensured that I was ignoring the first line which only holds titles for each column.

Undoubtedly the most important thing to implement in our “Monte Carlo Error Estimation” graph, is utilizing *$RANDOM* to generate four data files containing random seeds. Forgetting this step would yield four identical data files, rendering our trials useless. After this, all that was left was to save this data into different files and plot them.

Conclusion

From this assignment, I learned that computer programs are helpful, if not imperative, in applying the Monte Carlo method to predict long-term mathematical trends—in this case, estimating the value of π. In regards to coding this with UNIX, I learned the importance of seeds during my experience coding the *Monte Carlo Error Estimation*. This will prove essential in future coding projects that demand the use of generating randomness across different trials.

Credits

- In order to subtract π from the estimated π values in the Monte Carlo output data file, I used a version of Professor Long's code which he posted into the CSE 13S Discord server. I personally implemented the code that would perform this arithmetic on 4 different data files using different seeds.

Citations

- *IBM. (n.d.).*, What is Monte Carlo Simulation?, *IBM*, Retrieved January 22, 2023, from https://www.ibm.com/topics/monte-carlo-simulation