

Estimating Pi Using Coconuts: Graphs and What Happened while Writing This

Isabella Phung

January 20, 2023

1 Introduction

This document details the specifics of each file and exactly what they use rather than the design thinking that leads to each one of these files. For more design related questions, check the DESIGN.pdf.

2 The Files

`monte_carlo.c` - This file was written by the instructors of Winter 2023 CSE 13S at UCSC. It generates a seed using the computer's time, then checks for any user inputted op codes using the `getopt()` function. the options are `-n jint numberi`, which indicates how many values this iteration of the `monte_carlo.c` program will generate; and the `-r jint numberi` option, when given an integer, this will be substituted as the seed rather than the time value. If it isn't given any valid opcodes, it will exit with an error. The program then proceeds with its algorithm. A random x and y value are generated between 0 and 1. The distance between the origin and the point is calculated via

$$distance = x^2 + y^2$$

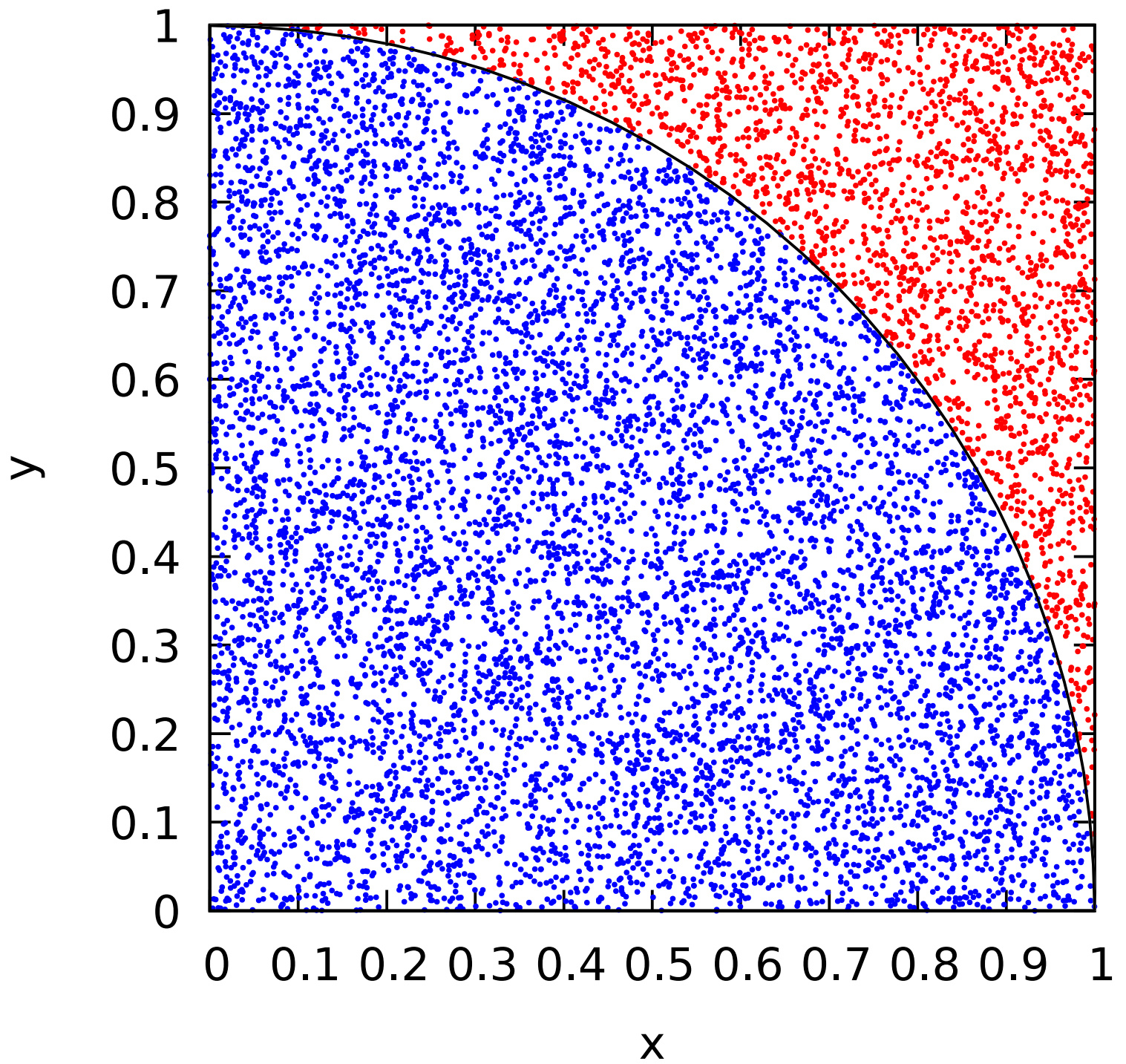
If this distance value is below 1, then it's inside of the circle, otherwise it's outside of the circle. The `monte_carlo.c` program keeps track of how many values are within the circle. That total is divided by the number of all points placed, multiplied by four, which should give an approximation of pi.

`Makefile` - this file was written by the instructors of Winter 2023 CSE 13S at UCSC. This file uses clang to compile `monte_carlo.c`. It also generates an object file, which helps with compilation. It has the following flags: `-Wall -Wpedantic -Werror -Wextra`, which will flag any errors that may cause any issues and prevent a program with errors in it from running. It accepts "make all", which will compile the `monte_carlo.c` program. "make clean" will remove any unnecessary files. "make format" will format `monte_carlo.c` with the proper c formatting style indicated by the CSE13S style guide.

plot.sh - this file will generate two graphs, showing 10000 random points within and outside of the circle. The second shows the error of multiple runs of the monte_carlo.c program of up to 10000 iterations.

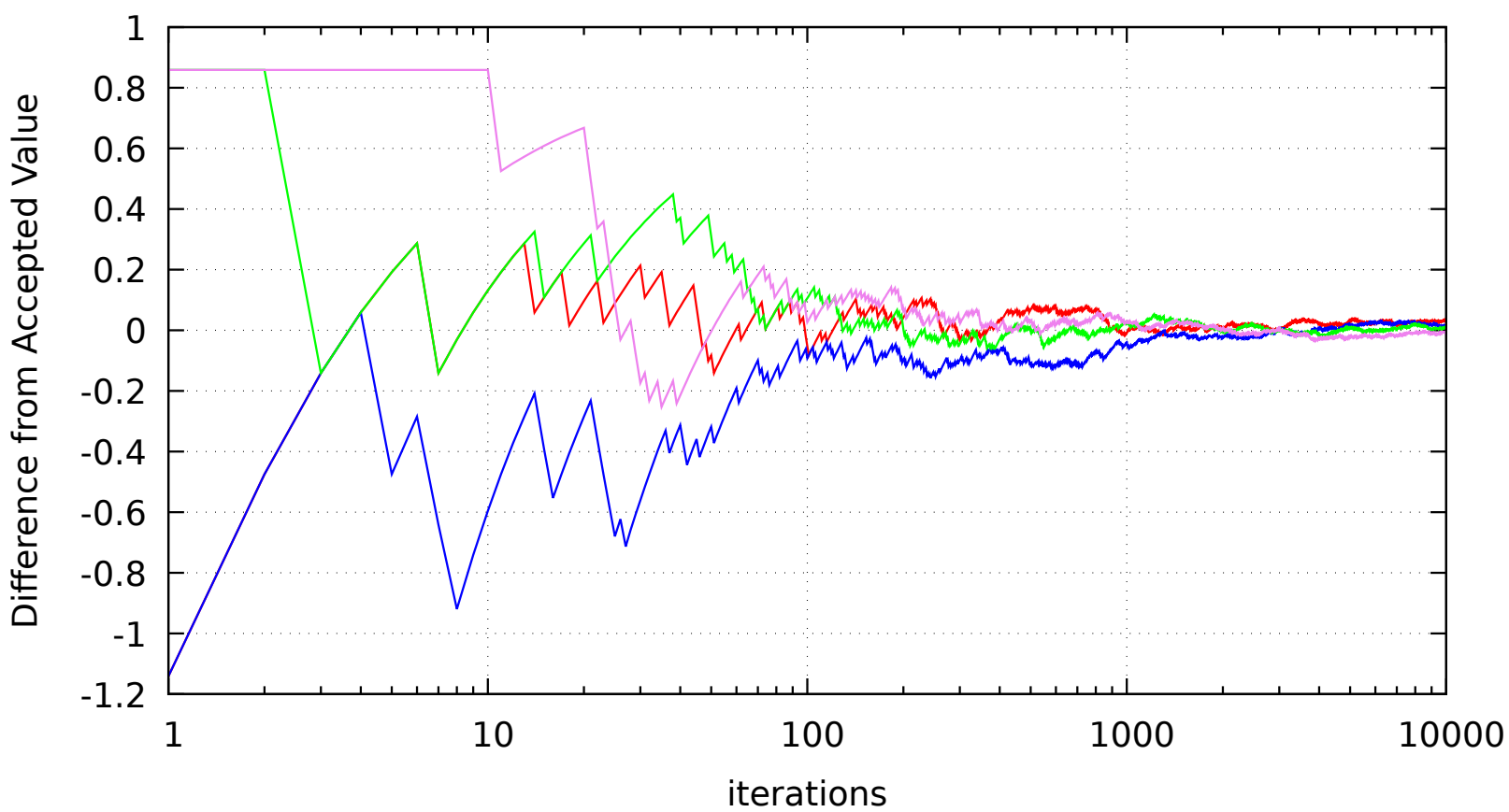
3 Graphs

Dropped Coconuts



Shown above is the circle diagram showing the randomly placed points within and outside of the circle.

Pi Error over multiple iterations



Shown above is the error graph showing the multiple run iterations and how the error diminishes around 9000 points.

4 Utilized Resources

Dev helped elaborate upon the method so as to not spend hours parsing strings: Awk. With it, it was significantly easier to quickly analyze the necessary data for creating our graphs. I initially tried my hand at the problem before I had my section and ended up spending hours creating a horrific script that involved weird string concatenation and it was a terrible mess. I didn't realize we were allowed to use a utility like awk to help our data analysis. I still had some difficulty working with awk by myself so I used the following online resources to help: How to print columns using Awk: <https://www.cyberciti.biz/faq/bash-scripting-using-awk/> How to skip the first line in Awk: https://linuxhint.com/skip_line_awk/ How to create an if else in Awk: <https://linuxhandbook.com/awk-if-else/> How to perform arithmetic in Awk: https://www.gnu.org/software/gawk/manual/html_node/Arithmetic-Ops.html The official GNUpot documentation located here (<https://docs.w3cub.com/gnuplot/>), was critical in understanding how to get gnuplot to cooperate. I used to understand how to change canvas size, use lines or dots in my graphs, create objects to make an inscribed circle, as well as create the log scale x axis. Despite the documentation provided for GNUpot, I used these resources to help with creating my graphs: How to create an arc using the circle object: <https://subscription.packtpub.com/book/big-data-and-business-intelligence/9781849517249/3/ch03lvl1sec43/plotting-with-objects-new> How to set grid lines: <https://stackoverflow.com/questions/20069093/how-do-i-set-grid-spacing-in-gnuplot>