

Estimating Pi Using Coconuts: Design

Isabella Phung

January 20, 2023

1 Introduction

The irrational number pi is available in programming and C in a variety of libraries. Utilizing the digit in most applications requires less than 10 digits. There are a wide variety of ways to estimate pi, one of which is illustrated in the following program known as monte_carlo. The monte_carlo program was written by the instructors of the CSE13S Winter 2023 class and estimates pi by drawing a unit square, inscribing a circle within it and dropping random points within the square. By finding the ratio between the number of points within the circle in comparison to the number of points outside of the circle, we should get approximately

$$\frac{\pi}{4}$$

. If this value is multiplied by 4, we get our approximated pi value! This concept is mentioned in the sitcom series, Gilligan's Island, which describes the idea thusly: If you're stuck on an island and need to calculate pi, what do you do? You can draw a square around a palm tree and inscribe a circle inside of it. If you shake the coconuts down, they should fall randomly within the square or circle. You can then use the algorithm described as above to find pi! But about how many coconuts do you have to drop to get a relatively accurate value of pi?

2 Method

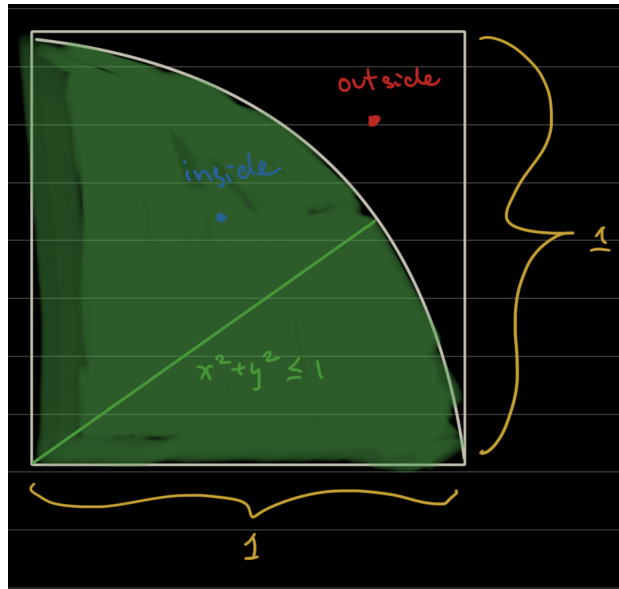
The monte_carlo program follows the following approximate steps to perform an approximation: generate random x and y values within the unit square, so somewhere between 0 and 1. using the formula for a unit circle,

$$x^2 + y^2 = 1$$

The distance from the origin of the circle is found which can also be described as the radius of the inscribed circle. If this origin is less than 1, then it's within the circle, greater than 1, it's outside the circle. This was not described in a previous version of the DESIGN.pdf document, but in order to retrieve the values of the monte_carlo program, the awk library would be used. The extracted values will then be fed into gnuplot to create the diagrams.

3 Diagrams

For clarity and visual simplicity, the graph shown in this paper will not feature a full circle but rather a quarter circle instead. The general principle still remains the same.



Points outside of the circle are distinguished in red in comparison to the blue points illustrated within the circle.

4 Error

In order to illustrate the increase of accuracy as the monte_carlo is run more times, the following graph was generated illustrating the difference between the approximated value of pi that monte_carlo spits out in comparison to an accepted value of pi. The accepted value of pi was 3.1415926535, as recommended by Dev, a TA in the CSE13S class. It was initially thought to use the percent error to calculate the error between the generated and accepted values of pi, but after looking over the spec, simply taking the difference between the two would suffice.