Monte Carlo and π

Nariman Haidar

Spring Term 2023

Introduction

In this task, we'll utilize a Monte Carlo simulation to estimate the value of π, and by constantly estimating the value of π, we'll be able to increase the precision of the estimated value of π.

Monty Carlo method:

Monte Carlo simulation can be used to describe any technique that approximates solution to quantitative problems through statistical sampling. Monte Carlo analysis involves determining the impact of identify the range of possible out comes for a number of scenarios. The basic principle of the Monte Carlo approach is to create points at random inside of a square and count how many of those points fall inside an arch within the square.

One way to approximate the ratio of the circle's area to the square's area is to look at the number of points that fall inside the circle to the total number of points produced.

If the square has an area $4r^2$, and the the circular's area is $\pi\ r^2$. Since the probability that a dart would land inside the circle is $\pi\ r^2/4r^2$,so the pi is

$$\pi = 4 \cdot \frac{\text{number of darts inside circle}}{\text{number of darts inside square}}$$

The challenge:

To implement the code we have three methods. The first one is a function dart that simulates one dart being fired at a target with a given radius using Pythagoras' Theorem. It returns true if the dart lands inside the circle and false otherwise.

```
def dart(r) do
  x = :rand.uniform(r)
  y = :rand.uniform(r)
  if :math.pow(r,2) > :math.pow(x,2) + :math.pow(y,2)
  do true
    else false
    end
    end
```

The second function is `rounds/3` and with this method, you may count how many darts (k) land inside the circle and then add those darts to the total value (a). This method is going to call dart to determine whether or not to increase the accumulator. Since (a) is the number of hits, and (k) is the number of darts and (r) is the radius.

```
def round(0, _, a) do a
end
def round(k, r, a) do
  if dart(r) do
    round(k-1, r, a+1)
  else
    round(k-1, r, a)
  end
end
```

Finally, the third function is to I construct a program that estimates Pi based on the total darts fired and the number of hits.

Then we can see the results in the this table: That we can see in the results that when we have 10000 darts and the radius was 10000 I didn't get closer to the 3.14159 but when I increased the darts to 1000000 and the radius to the 1000000 so I get little closer to the 3.14159 with 4 4 decimal places.

Number of Rounds(n)	**Number of Darts(K)**	**Radius(r)**	π
5	10000	10000	3.144114
5	1000000	1000000	3.141510