Monte Carlo

Björn Formgren

Spring 2023

Introduction

The sixth assignment in ID1019 consisted of approximating π by using the Monte Carlo technique. The technique is broadly used to get an estimate of something that is hard to calculate. In this assignment the calculation were done by throwing darts at a square which contains a circle sector. If the square has an area of r^2 the circle sector has an area of $\pi/4 * r^2$. The probability that a dart hits inside the circle is of course the ratio between the square and circle - $\pi/4$, therefore if we know all the darts hitting the circle and all darts in total we can simply estimate π by calculating $4 * dartsCircle/dartsSquare$

Code

```
defmodule Monte do
  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

  def round(0,_,hits) do hits end
  def round(n,r,hits) do
    if dart(r)
    do
      round(n-1, r, hits+1)
    else
      round(n-1, r, hits)
    end
  end

  def rounds(k,r) do
    a = round(1000,r,0)
```

```
    rounds(k,1000,r,a)
  end

  def rounds(0,n,_,a) do 4*a/n end
  def rounds(k,n,r,a) do
    a = round(n,r,a)
    n = n*2
    pi = 4*a/n
    #formating excluded
    rounds(k-1,n,r,a)
  end
end
```

Firstly, a function that throws a random (coordinate) dart was created returning true if it hit the circle and false if it did not. As per the instructions, the comparison $r^2 > x^2 + y^2$ was implemented to verify this.

Next, a function that returns the number of hits was implemented. This function was named round and took the number of darts, k, on a target with radius r, and adds the hits to *hits* accumulator.

Lastly. a test that could run a number of rounds were set up. The function was named *rounds* and displayed the approximated π after every round.