Buffon's needle problem

061800825 Shimura Koki

June 30, 2021

1 Preface

Random processes is sometimes useful to gain the approximate value of pi. The famous one is Monte Carlo method; we draw a quarter circle in a square, plot points randomly, and search the rate of the inside points against the outside one. Here I'll introduce the other method. This is called "Buffon's needle". Here, we set d and l as the distance between two lines and the length of a needle($l \leq d$). The reference is [GW],P.77-78.

2 Theory

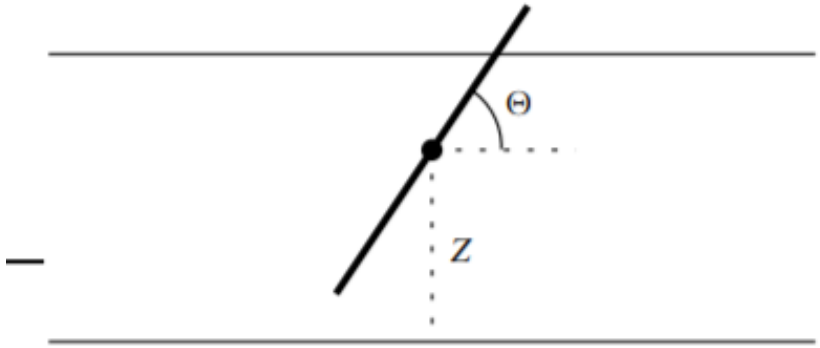

Fig. 5.5 The needle and the ruled plane in Buffon's problem.

There are two parameters to decide the position of needle. We define Z as the distance between the midpoint of a needle and a line below it, and Θ as its inclination to horizonal.
Z is uniformly distributed on [0,d], Θ on$[0,\pi]$, and these two parameters are independent. These conditions implies that

$$f_{Z,\Theta}(z,\theta) = \frac{1}{\pi d} \tag{1}$$

and intersection occurs if and only if$(z, \theta) \in B$, here

$$B = \left\{ (z, \theta) : either \quad z \leq \frac{l}{2} \sin\theta \quad or \quad d - z \leq \frac{l}{2} \sin\theta \right\} \tag{2}$$

Thus the probability is given by

$$\iint_B f_{Z,\Theta}(z, \theta) dz d\theta = \frac{2l}{\pi d} \tag{3}$$

3 Practice

When we throw needles on the lines, they have intersections along the probability that we derive above. This shows that

$$\frac{S}{N} = \frac{2l}{\pi d} \tag{4}$$

here S is the number of needles that have intersections, and N is the total number of thrown needles. By using this fomula, we can get the approximate value of π. I constructed the program to calculate it, using Python 3.6.9, GCC 8.4.0 on Ubuntu.

Listing 1: generate Pi from Buffon's needle problem

```
import random
import math

N = int(input())
d = 5
l = 3 # We have to set l that satisfy l<=d.
i = 0
S = 0
while i < N:
    X = random.uniform(0, 1)
    Z = random.uniform(0, 1)
    r = math.sqrt(X**2 + Z**2)
    if r < 1:
        sine_theta = Z / r
        i += 1
        z = random.uniform(0, d/2)
        if z <= l / 2 * sine_theta:
           S += 1

print(2 * l * N / (S * d))
```

You can play this by choosing numbers as you like. Through some tests, I got the values below:

$$\begin{aligned} 2.791 \quad & (N = 100) \\ 3.333 \quad & (N = 1000) \\ 3.117 \quad & (N = 10000) \\ 3.137 \quad & (N = 100000) \end{aligned}$$