AMS326HW4

william.zhang

April 2019

1 Problem Introduction

We wish to calculate the amount of times a needle crosses a line on a plane by random throws of a needle. The probability of sucess is the the number of needles thrown divided by the number of number of throws

2 Algorithm /Psuedo Code

The following is throwing the random needle.We return a random angle.Also a random posistion where the needle is thrown in between the spaces

Algorithm 1: ThrowNeedle(Length,SpaceBetween)

```
return length, uniform(0,spacebetween),random.uniform(0,np.pi)
```

This is the procedure to check if a needle is intersecting with a line

Algorithm 2: Checkintersections(center,length,angle,linedistance)

```
if $center + length/2 * np.sin(angle) >= linedistance$ then
    return True
else if $center + length/2 * np.sin(angle) <= 0$ then
    return True
else
    return False
```

Now we put it all together in one method

Algorithm 3: MonteCarlo(linedistance,pinlength)

```
numberofiteration = 300000000
counter = 0
for $x = 0$ tonumberofiterations do
    length,center,angle = throwneedle(pinlength,linedistance)
    if checkintersections(center,length,angle,linedistance then
        counter += 1
return counter/numberofiteration
```

3 Results

Our algorithm resulted in the following output

```
Approximation with a pinlength 1: 0.015894333333333333
Approximation with a pinlength 0.5: 0.007977
Approximation with a pinlength 0.333:0.005238666666666667
Approximation with a pinlength 0.25: 0.003986666666666667
```

4 Analysis

The problem is performed with 300 million throws on a random board. The distance between each line varies but I did limit it 40. And it pretty accurate as the ping size gets bigger does the probability of landing with a line on the board it self.