

# Pi Estimator via Monte Carlo Method

Adrian Lu

Tuesday November 11th, 2014

## 1 Summary

This is the second time ever I have coded in C++, and I decided to try making one of the assignments assigned by Dr. Dorin in CMSI 186. I ended up deciding to make a PiSolver via the Monte Carlo Method of estimation. For me the hardest part was definitely familiarizing myself with the classes that need to be imported to do even very basic functions. This surprised me after having so much experience with Java, where many of the basic mathematical functionalities are built in. Besides this, the overall syntax of the language was very foreign, which I expected, but still surprised me after having the transition from JavaScript to Java feel so smooth. In the end, I was able to use the same mathematical reasoning as I did in my Java version of the program, and I really was interested seeing the final programs and comparing the differences.

## 2 Code Snapshot

Listing 1: C++ code using listings

---

```
1 #include <iostream>
2 #include <math.h>
3
4 double randomNumber() {
5     return (double)rand()/(double)RANDMAX;
6 }
7
8 double getRandDistance() {
9     double x = randomNumber();
10    double y = randomNumber();
11    return sqrt(pow(x,2) + pow(y,2));
12 }
13
14 int main(void){
15     int numIter = 10000000;
16     int numHits = 0;
17     for (int i = 0; i < numIter; i++) {
18         if (getRandDistance() <= 1) {
19             numHits++;
20         }
```

```
21     }
22     double pi = 4.0 * ((double)numHits / (double)numIter);
23     std::cout << pi;
24     std::cout << "\n";
25     return 0;
26 }
```

---