Abstract

This lab aims to illustrate the implementation of the monte carlo problem for calculating pi and giving the estimated time for it to execute each time we change the interval value.

1 Introduction

The lab was requiring an implementation of monte carlo algorithim and to test for each certain value (1000,10000..) the amount of time the program took to execute and calculate pi.

2 Implementation

First we declared the i for iterations and count to track number of points and interval. We tried to prompt the user to enter a certain value for interval but it was giving bugs so we ended up initilizing the interval in order to run the code and get an output, count is increamented each time the equation z ¡=1 which means i have a random point generated hitting the circle part.

```
#include <stdio.h>
#include <math.h>
#include <stdlib.h>
#include <time.h>
#define SEED time(NULL)

int main() {

 srand( SEED );
 int i, count, interval = 1000000;
 double x_circle,y_square,z,pi;

 //printf("interval = ");
 //scanf("%d", &interval);

 count = 0;

 for(i = 0; i < interval; ++i) {

      x_circle = (double)rand() / RAND_MAX;

      y_square = (double)rand() / RAND_MAX;

      z = x_circle * x_circle + y_square * y_square;

      if( z <= 1 ) count++;
 }

 pi = (double) count / interval * 4;

 printf("Approximate PI = %g", pi);

 return(0);
}
```

3 Experimental Platform

We used Visusal Studio to write and execute the serial code.

4 Results

For interval = 1000, Approximate PI = 3.052, execution time = 1.072 secs
For interval = 10000 ,Approximate PI = 3.1416, execution time = 1.216 seconds
For interval = 100000, ,Approximate PI = 3.13464, execution time = 1.216 seconds
For interval = 10000000 ,Approximate PI = 3.14144, execution time = 9.637 seconds

5 Conclusion

To sum up, the monte carlo problem increases in time to be executed when the interval increases, it is a problem where the area of the square and circle are combined and pi is calculated depending on the number of points generated inside the circle and square. Thus, the problem is not scalable.
Repository link: https://github.com/JicksonHD/$CSC_447_LAB1 - Parallele-$