Exercises

Numerical Estimation of $pi$

1. How do you calculate $\pi$ using the ratio of points that fall within the circle and square? Use a full circle for your computation.

Complete either the `Python` or `C++` Monte Carlo program to calculate $\pi$ using this integration technique . Include the entire code within your report or attach as the code files to your submission and comment upon the lines of code that you wrote in the report.

We can compute the value of pi by taking the following ratio between the area of a circle of diameter d and a square of length l.

From the picture, we have:

Area C = (d/2)^2\*pi and area S = l^2. Thus, pi can be computed as

Where the areas mean the number of points that fall inside the circle and those falling outside the circle. It is important to mention that the denominator is the total number of points given by: Ncircle + N square.

From the above formula, we can view the ratio Ncircle/(Ncircle + N square) as a probability of any given point to fall inside the circle. Additionally, the ratio l/d > = 1.

As a result, if we were to modify the diameter of the circle, the ratio Ncircle/(Ncircle + N square) will decrease and in the limit d = 0, pi is zero: a circle whose radio is zero.

However, changing the position of the circle -as long as it is completely contained in the square – will have no effect on the the computed value of pi. But the smaller d is, the worse our pi estimation will be. To overcome this problem, many points should be taken in such a manner the whole are of the circle is full of points.

2. Perform the $\pi$ estimation for 1000, 100,000 and 1,000,000 trials. Take a screenshot of these estimations and include them in your report (e.g of the python plots). What happens to the accuracy of the $\pi$ estimation when going from 1000 to 1,000,000 trials and why?

The more samples we take, the more accurate our result will be. This is given by the strong law of large numbers, which states that the more samples we take from an experiment, the more the mean of these numbers will approximate the true value; that is:

SLLN

The rate of the error is given by the Central Limit theorem. Then, the error in our Monte Carlo of pi decays at a rate 1/sqrt(N); hence we have the following plot:

3. What happens if you use the same seed for the PRNG?

If we set the seed for the PRNG, the sequence of pseudo-random numbers we obtain is always the same for each run our program. That is:

If we run three times our code with the same seed, we obtain the following numbers:

As observed, these are identical.

It is worth pointing out that setting or not the seed will have no effect on the pi estimation we compute since the random numbers are generated in such a manner that we can think of them as identical distributed and independent, iid for short.

4. What happens to the estimation of $\pi$ when the circle origin is

changed? Why?

As long as the whole area of the circle is inside the square, the center of the circle has no effect on the pi estimation.

This happens since we are counting the number of points inside of the circle, then its center has no importance. Besides, the used random numbers are iid and, as result, each point has the same probability of falling inside the circle. This means

To illustrate the latter, we have the following plots for distinct location of the center of the circle.

As we see, the pi estimation is not affected.

5. What happens to the accuracy of the estimation when you increase the

square size, or decrease the circle size? Is there an optimal ratio?

The optimal ratio is when l = d because this is the value for which the probability of finding a point inside the circle is the highest; we “cover” the largest area when l = d.

From Quantum to Classical Mechanics

1. Prove the Hellmann-Feynman theorem, equation (2).

2. \*\*Bonus:\*\* Explain the Born-Oppenheimer approximation in your own

words. You do not have to use any equations (but you may if you

wish).