Report - Problem Set no 2

Yaghoub Shahmari

October 15, 2021

Problem 1

Basic description:

We want to simulate the Ballistic Deposition with relaxation. So, we will generate some random integers in the surface length range. Then, We put The random generated number as the index, and drop the paricle. but we have to check the neighbors of the chosen point of the surface. If there was a hole in the neighbors, Then the particle will move to that position. There is two way to store the data. We can have a 2D Matrix as our deposition layers or have a 1D vector that saves only the height of layers. I use the first method for Graphic view and the second to show the $W_{(t)}$ and $H_{(t)}$ in terms of time.

The results:

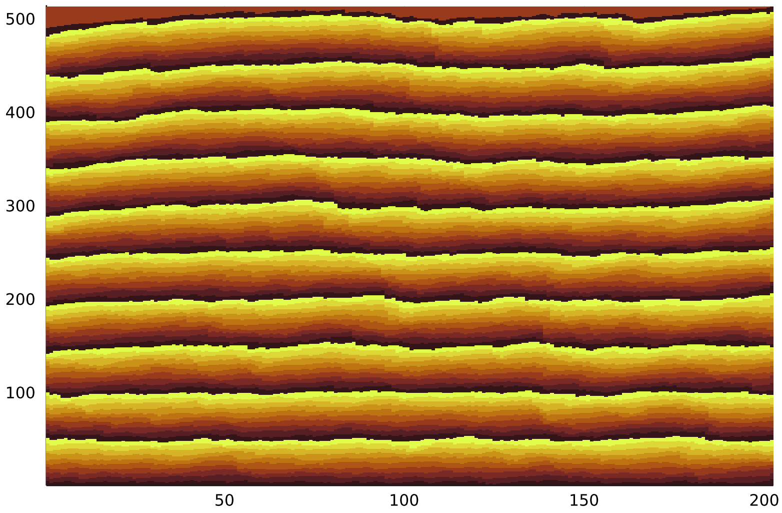


Figure 1: Ballistic Deposition with relaxation: For one hundred time steps, one thousand particles per each time step will drop on a surface of two hundred lengths. For every ten steps of time, The color of the visualization will change.

Now we're going to see the statistics of Deposition. To have a better view of the dynamic, we show our w plot on logarithmic scales. For each surface length, we split the time range into exponential periods. And for every considered condition, the drop rate of particles is one particle per time unit.

We created every plot from ensembles of 1000 runs, and all of them have error bars (some of them have very small errorbar because of accuracy).

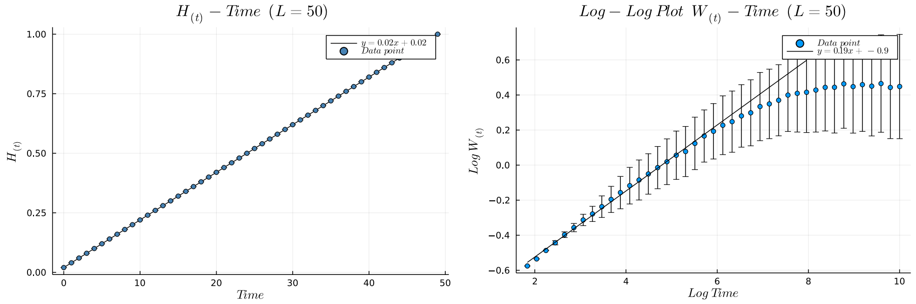


Figure 2: Ballistic Deposition with relaxation for $L = 50$

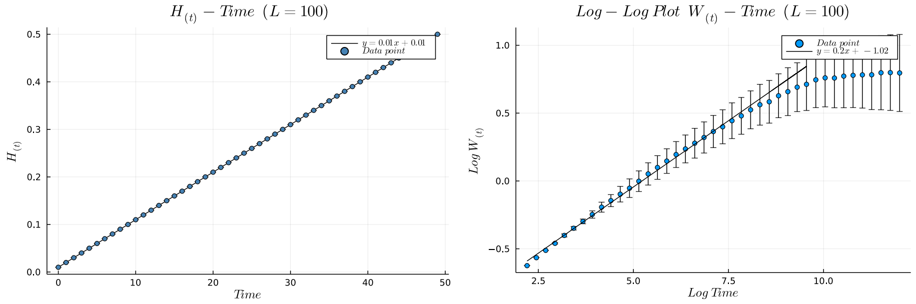


Figure 3: Ballistic Deposition with relaxation for $L = 100$

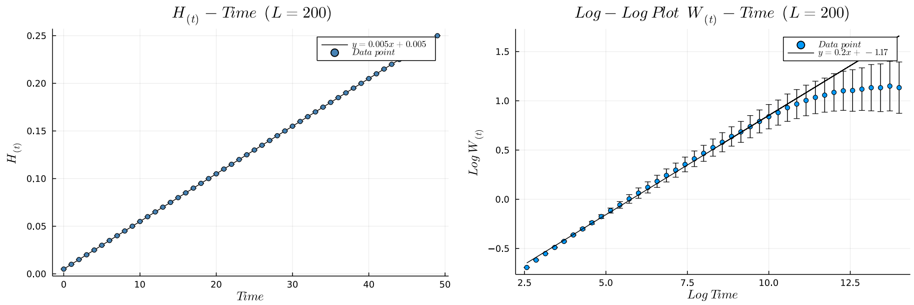


Figure 4: Ballistic Deposition with relaxation for $L = 200$

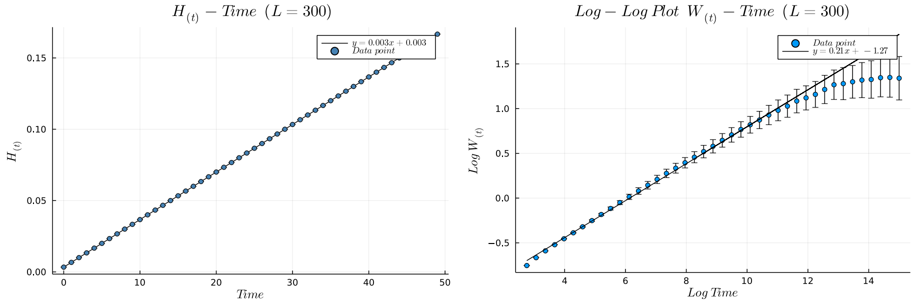


Figure 5: Ballistic Deposition with relaxation for $L = 300$

The results satisfied our expectations. As we expected, the amount of $W_{(t)}$ grows linearly (in log-log plot), and after a while, becomes a horizontal line. The β is equal to the initial line slope. So, for the following lengths, we have:

$$L = 50, t_s \approx 6, t_s = L^z \Rightarrow z \approx 0.458,\ \beta \approx 0.19, L^{\beta \dot{z}} = L^{\alpha} \Rightarrow \alpha \approx 0.09 \tag{1}$$

$$L = 100, t_s \approx 8, t_s = L^z \Rightarrow z \approx 0.452,\ \beta \approx 0.2, L^{\beta \dot{z}} = L^{\alpha} \Rightarrow \alpha \approx 0.09 \tag{2}$$

$$L = 200, t_s \approx 11, t_s = L^z \Rightarrow z \approx 0.435,\ \beta \approx 0.2, L^{\beta \dot{z}} = L^{\alpha} \Rightarrow \alpha \approx 0.09 \tag{3}$$

$$L = 300, t_s \approx 12, t_s = L^z \Rightarrow z \approx 0.437,\ \beta \approx 0.21, L^{\beta \dot{z}} = L^{\alpha} \Rightarrow \alpha \approx 0.09 \tag{4}$$

The H(t) is also as we wanted to be, it has a linear behavior, and the amount of the line slope follows:

$$line\ slope = drop\ rate \div L \tag{5}$$

Problem 2

Basic description:

In this question, we want to do the same things we have done in the previous question, but we have a little change. The particle will drop on the chosen spot, not in the neighbors, but also the particle is not going to place in the deepest place. The height will be the maximum height of neighbors and the midpoint. The run numbers and other parameters are the same as the previous question parameters.

The results:

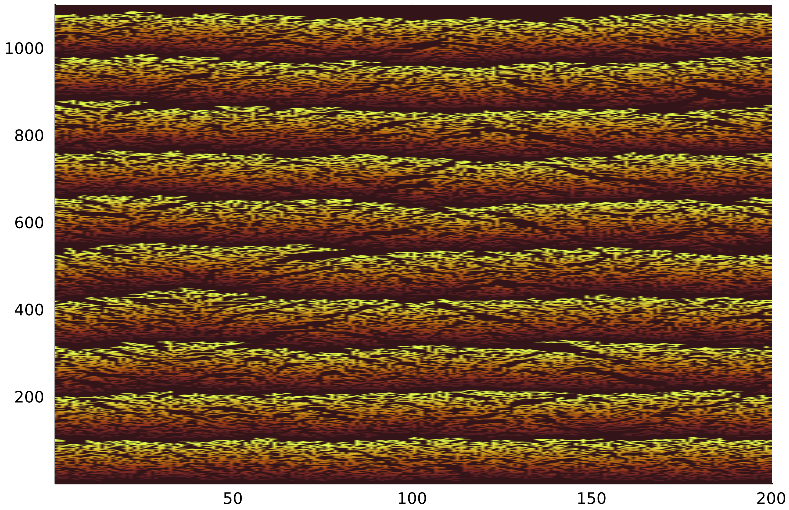


Figure 6: Ballistic Deposition : For one hundred time steps, one thousand particles per each time step will drop on a surface of two hundred lengths. For every ten steps of time, The color of the visualization will change.

Now we're going to see the statistics of the Deposition. As before, to have a better view of the dynamic, we show our w plot on logarithmic scales, and for every considered condition, the drop rate of particles is one particle per time unit.

The results of the statistic section are as they should be as before. Until the saturated, we have an oblique line. After that, we have a horizontal line. The line slope follows the rules we knew.

Now we are going to calculate the requested variables:

$$L = 50, t_s \approx 3, t_s = L^z \Rightarrow z \approx 0.281,\ \beta \approx 0.59, L^{\beta \dot{z}} = L^\alpha \Rightarrow \alpha \approx 0.166 \tag{6}$$

$$L = 100, t_s \approx 4, t_s = L^z \Rightarrow z \approx 0.301,\ \beta \approx 0.58, L^{\beta \dot{z}} = L^\alpha \Rightarrow \alpha \approx 0.175 \tag{7}$$

$$L = 200, t_s \approx 5, t_s = L^z \Rightarrow z \approx 0.304,\ \beta \approx 0.55, L^{\beta \dot{z}} = L^\alpha \Rightarrow \alpha \approx 0.167 \tag{8}$$

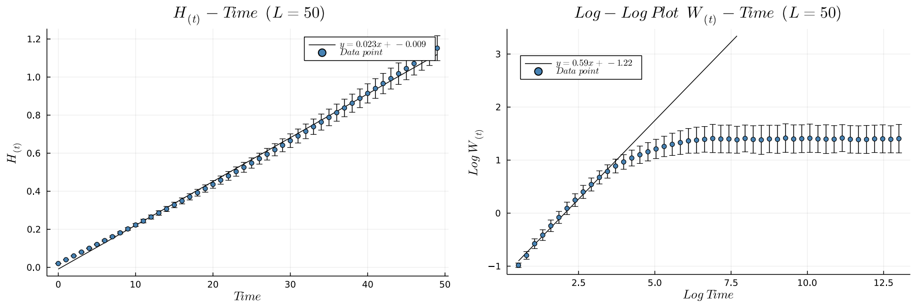


Figure 7: Ballistic Deposition for $L = 50$

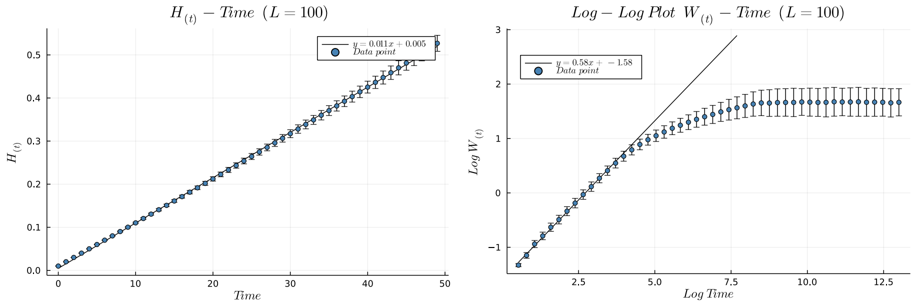


Figure 8: Ballistic Deposition for $L = 100$

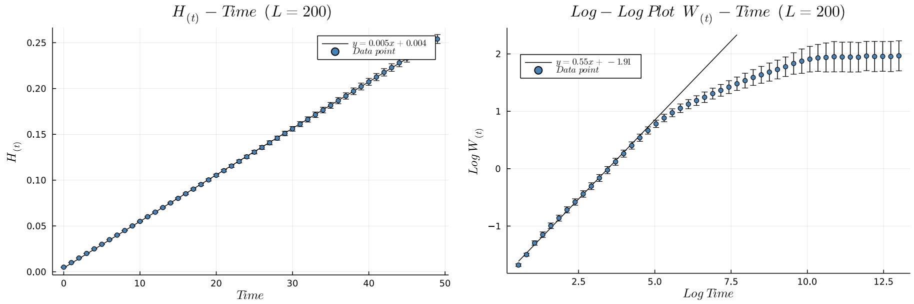


Figure 9: Ballistic Deposition for $L = 200$

$$L = 300, t_s \approx 6, t_s = L^z \Rightarrow z \approx 0.314,\ \beta \approx 0.53, L^{\beta \dot{z}} = L^\alpha \Rightarrow \alpha \approx 0.166 \tag{9}$$

The H(t) is also as we wanted to be, it has a linear behavior, and the amount of the line slope follows:

$$line\ slope = drop\ rate \div L \tag{10}$$

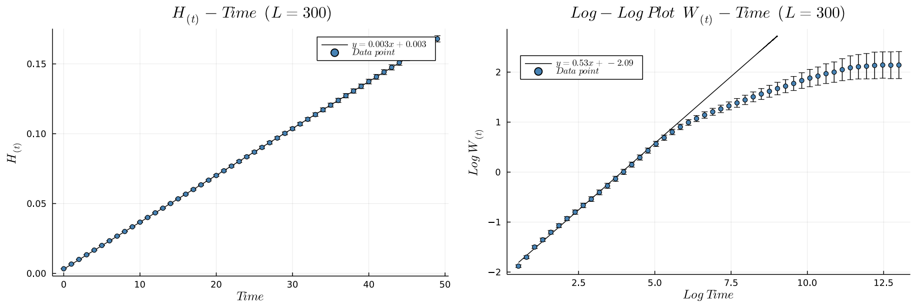


Figure 10: Ballistic Deposition for $L = 300$

Problem 3

Basic description:

For this question, we need to change the previous code a little. The only change we need to make is considering a condition. Only two types of particles can be dropped and caught. The particles that stick to the midpoint and those stick to the other particles.

The parameters are as following:

• The length of the surface is 300, and we will deposit 50000 particles.

The results:

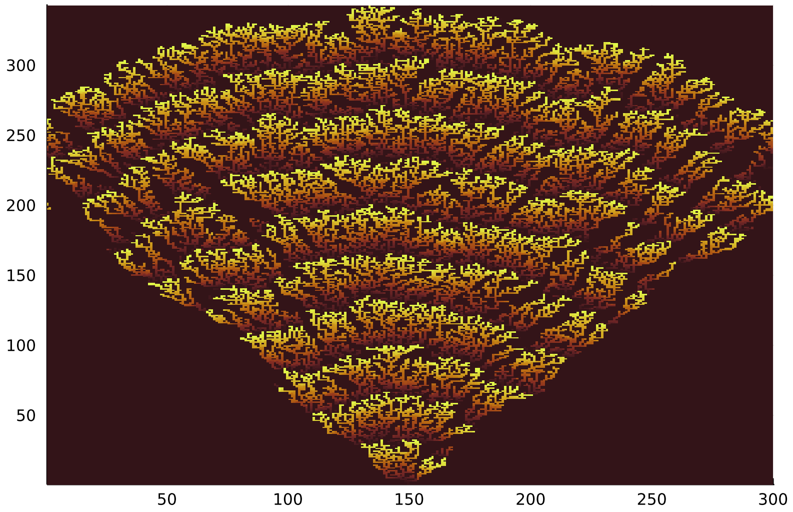


Figure 11: Competitive Ballistic Deposition : 50000 particles will drop on a surface of three hundred lengths.

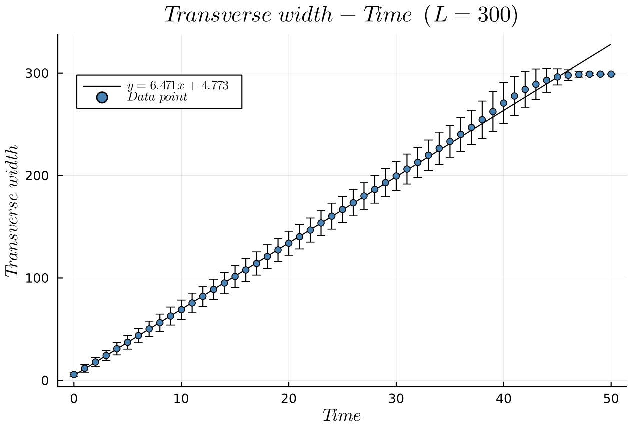


Figure 12: Competitive Ballistic Deposition for $L = 300$.

As it is clear, the transverse width of the shrub grows linearly, until it becomes equal to the surface length.

Problem 4

Basic description:

Here we want to simulate the percolation of a binary defined network. In that case, we create a random binary network using the python networkx. Then we will check if there is a path between every node on the sides of the network. If we find a route, we will consider the percolation successful. And we check that for a range of P. So here is the result:

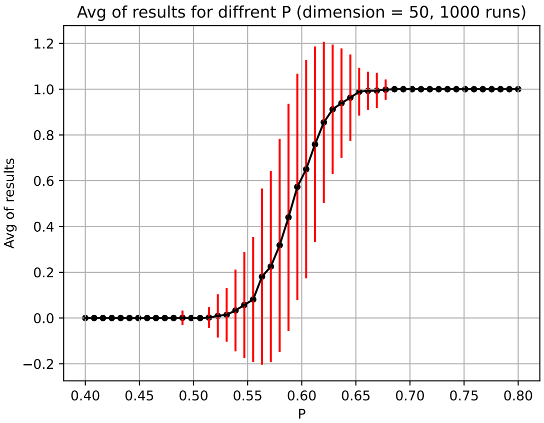


Figure 13: The results of percolation for a range of P for 1000 runs.

Problem 5

Basic description:

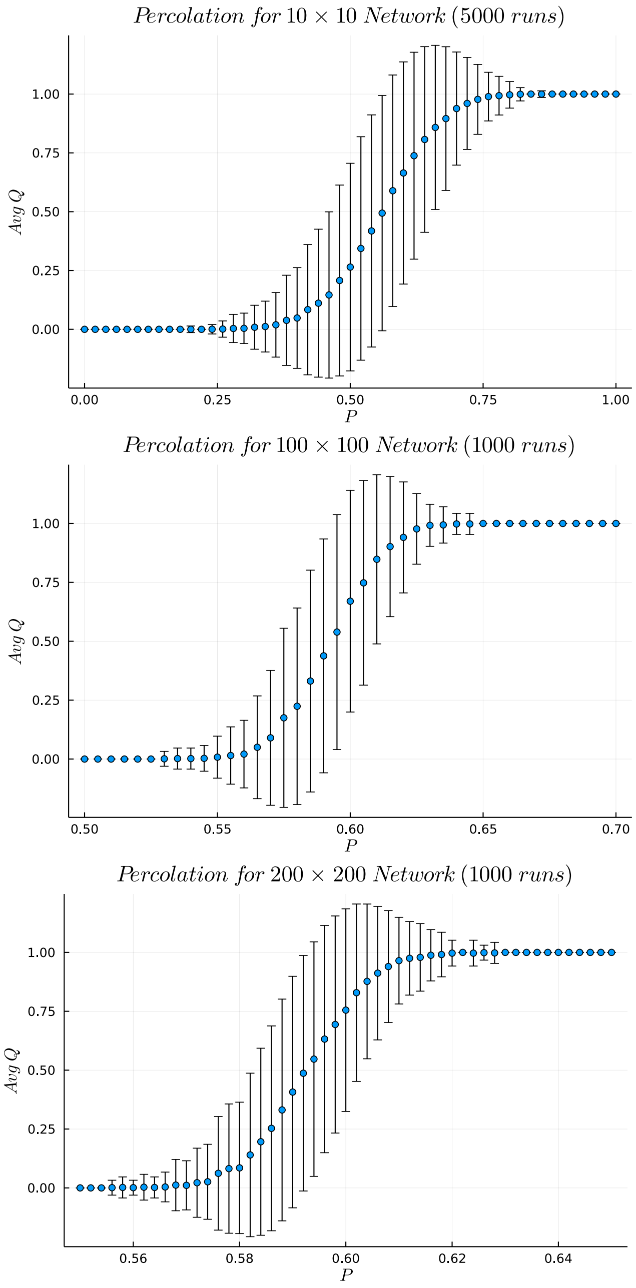
The coloring algorithm deployed as the question wanted. My program can find percolation in a binary network and return the result as 0 or 1. For some critical P (or at least near to critical point), I drew an animation of stages of the dynamic of percolation for colors. The results are available in the Q5 folder in the Figs folder.

Problem 6

Basic description:

For large networks, the coloring algorithm is not fast enough. So I deployed the H-K algorithm for a network of sizes 100 and 200. So I exported the result of percolation of P in a range many times and showed the statistic results as following plots:

The results:


Percolation for 10×10 *Network* (5000 *runs*)

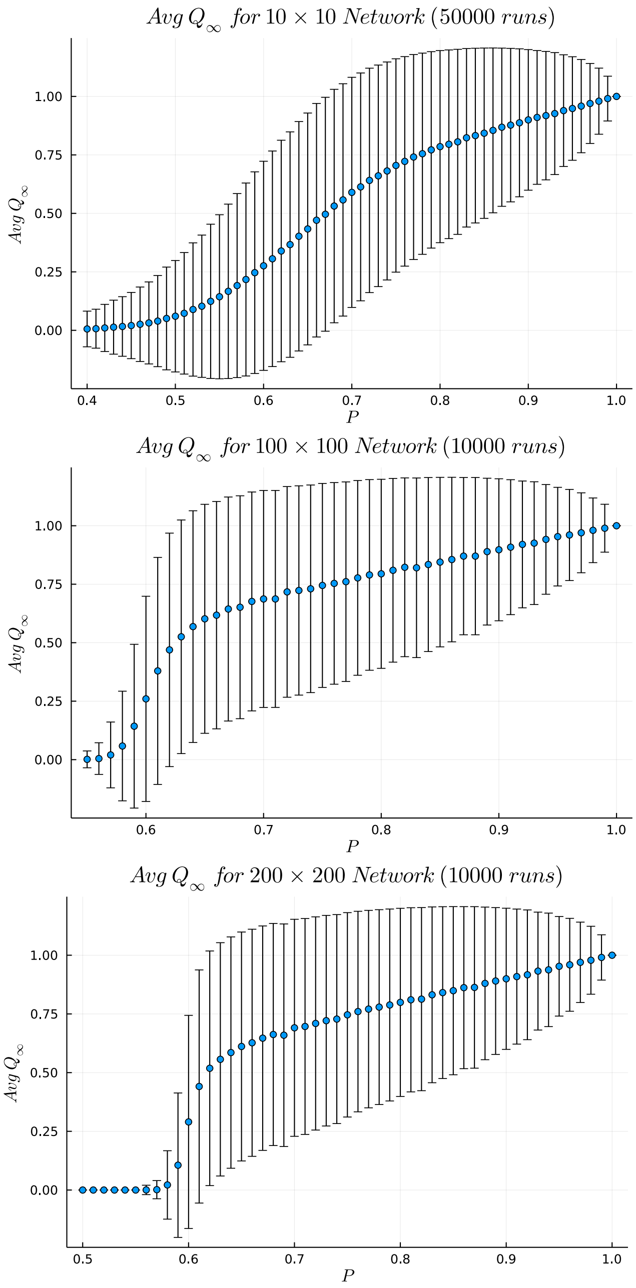
Percolation for 100×100 *Network* (1000 *runs*)

Percolation for 200×200 *Network* (1000 *runs*)

Figure 14: Percolation result for $L = 50, 100, 200$

Problem 7

Basic description: There is only one small change in the previous question in detecting percolation to achieve the answer for this question. We just need to check if a random point is in the connector cluster.



$Avg\ Q_{\infty}\ for\ 10 \times 10\ Network\ (50000\ runs)$

$Avg\ Q_{\infty}\ for\ 100 \times 100\ Network\ (10000\ runs)$

$Avg\ Q_{\infty}\ for\ 200 \times 200\ Network\ (10000\ runs)$

Figure 15: Q_∞ for $L = 50, 100, 200$

The whole data I gathered is in this link