

Report - Problem Set No 4

Yaghoub Shahmari

October 28, 2021

Problem 1

Basic description: In this problem, we try to find the chance of being connected to the infinite cluster again. The difference from the previous attempt is that I use the fraction of the size of the infinite cluster to the size of the whole network. So all of the previous steps will go through.

Results:

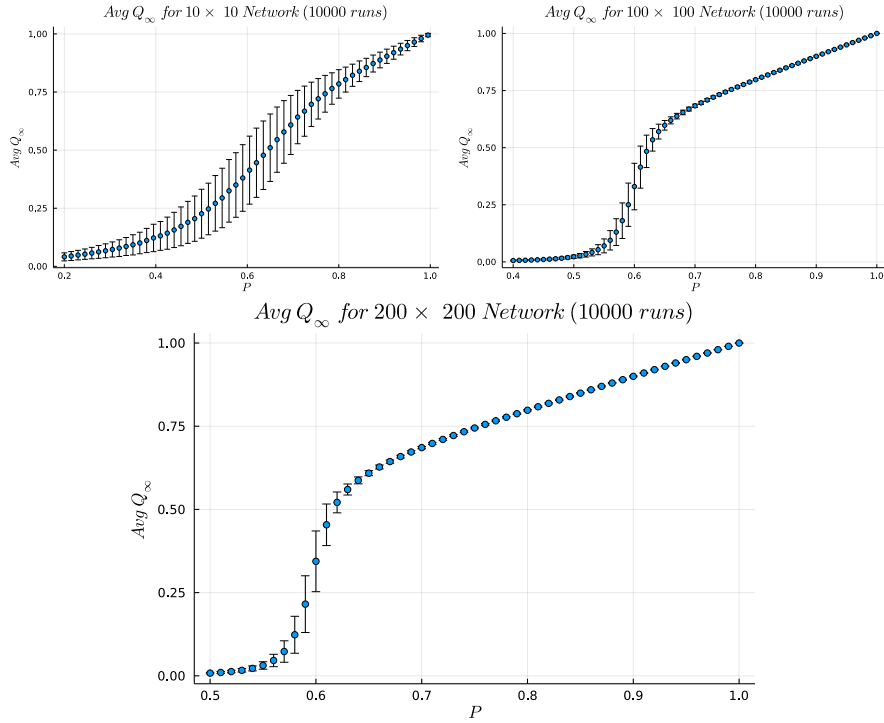


Figure 1: Q_∞ for $L = 10, 100, 200$

Problem 2

Basic description:

In this problem, we try to find the correlation length of percolation of grid graphs. In this case, we will find the largest finite cluster. Then, we have to calculate the gyration radius of that cluster as the book told us. and finally, we will run the calculation and average the data.

Results:

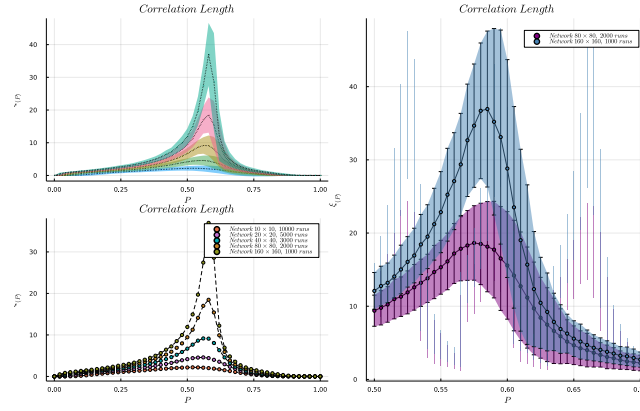


Figure 2: Q_{∞} for $L = 10, 100, 200$

Extrapolation results:

To find the requested parameters (ν and Pc_{∞}), we have to do previous steps for several P and L . After exporting data, we use the Curvefit function in the LsqFit library. Given that we intend to find the parameters and improve the accuracy we will consider that we know one of the parameters (find the amount from the internet) and find the other using the mentioned method.

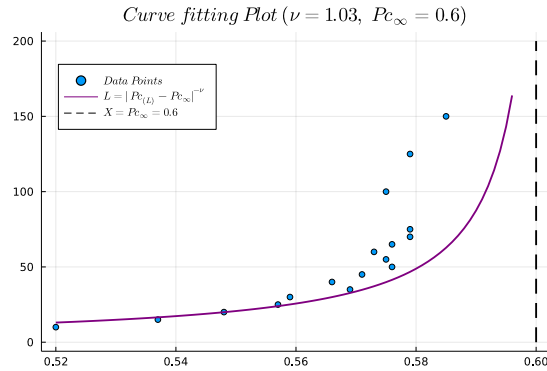


Figure 3: Curvefit result for exported data

Problem 3

Basic description:

In this problem, we want to find the cluster fractal dimension in the grid network percolation. In this case, we deploy the cluster grow algorithm, find the size and gyration radius of the created cluster, and export the data of ensemble we need for all parts.

Results:

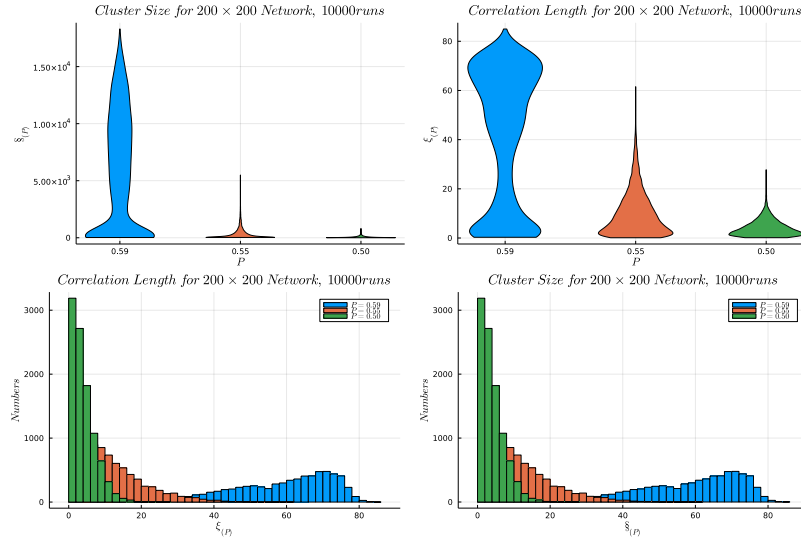


Figure 4: Figures of ensemble of wanted P data

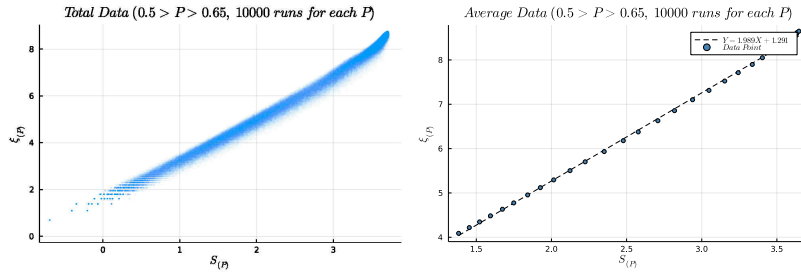


Figure 5: Scatter and average of total data. The line slope is equal to the cluster fractal dimension. The axis is log scaled.

There is also a few animation in the Figs folder for cluster grows.

Problem 4

Basic description:

In this problem, we want to simulate the 1D Random Walk. The Random Walker coded as the book explained it. The plots show the simulation for the 10000 time-steps for different P and $X_0 = 0$.

Results:

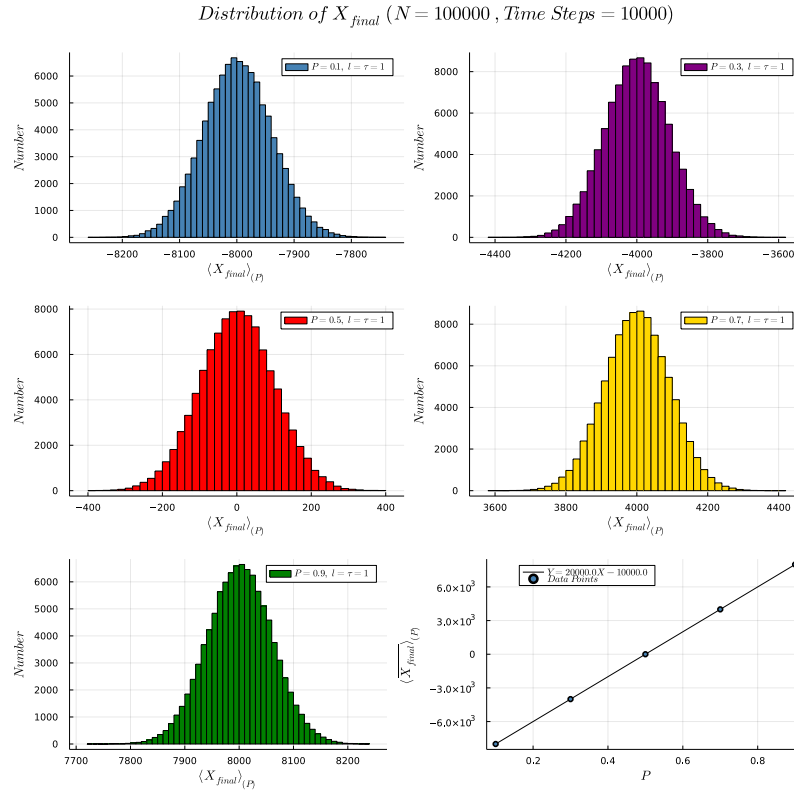


Figure 6: The histogram plots for exported data and average X_{final} for each.

Problem 5

Basic description:

In this problem, we add a condition to the previous simulation. We consider that twenty surface steps are surrounded by traps and in every step, we check if the random walker is trapped or not. After being trapped, we return the time steps that the Random walker was walking. (τ & $l = 1$).

Results:

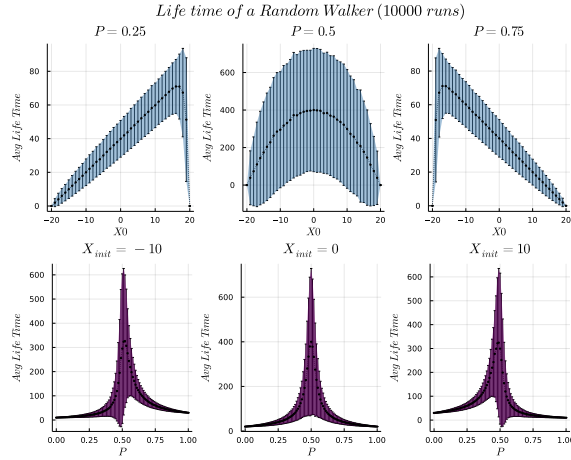


Figure 7: The Scatter figures for several X_0 and P and average X_{final} for each.

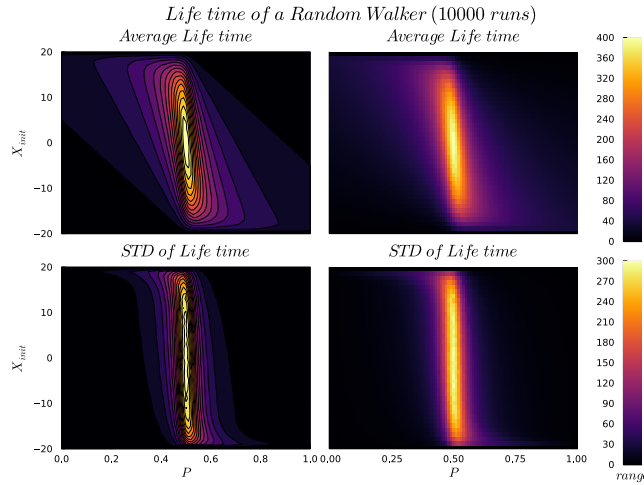


Figure 8: The contour figures for all possible X_0 and P and average and STD X_{final} for each state.

Problem 6

Basic description:

In this problem, we code a deterministic algorithm that calculates the presence density of a random walker on a cell on each time step for a trapped situation. Then we find when the random walker presence density will be near to 1.

Results:

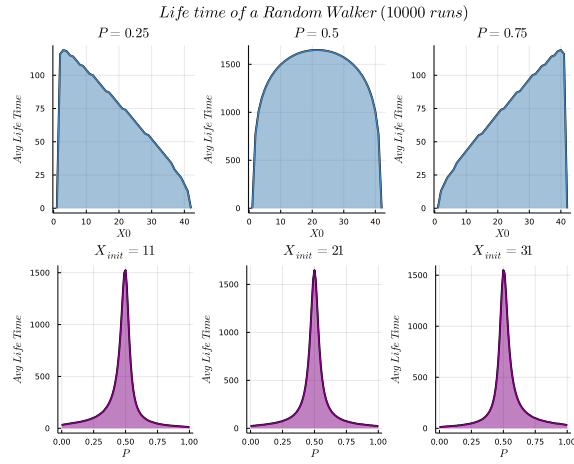


Figure 9: The plots for several X_0 and P and average X_{final} for each.

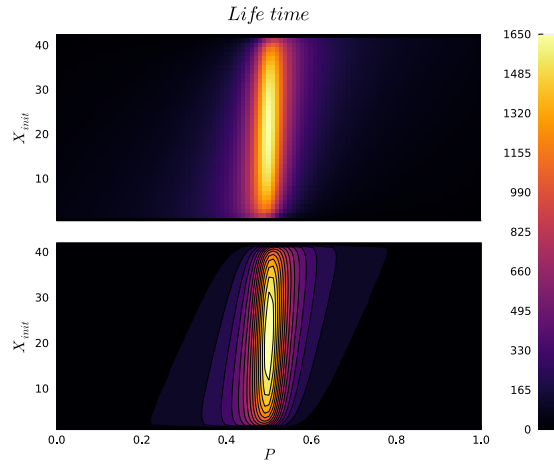


Figure 10: The contour figures for all possible X_0 and P and average X_{final} for each state.

The whole data I gathered is in [this link](#)
Thanks for watching :)