

# Computational Physics Homework Report

Fateme RamezanZade-99100693

Problem Set 4

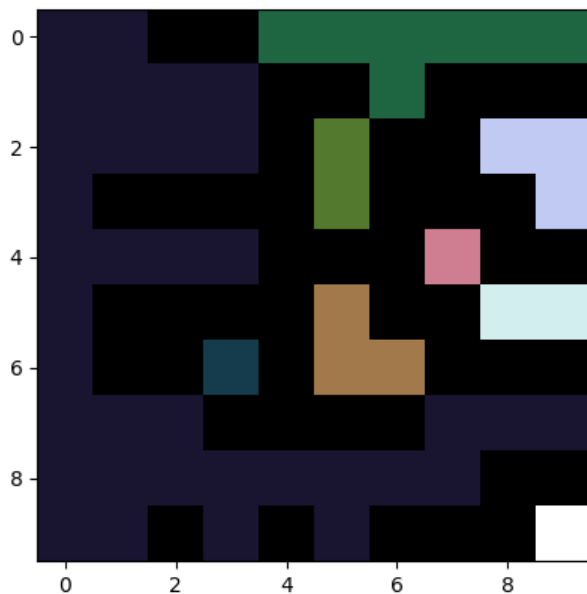
## 1. Percolation (with coloring algorithm)

### 1.1 Main Code

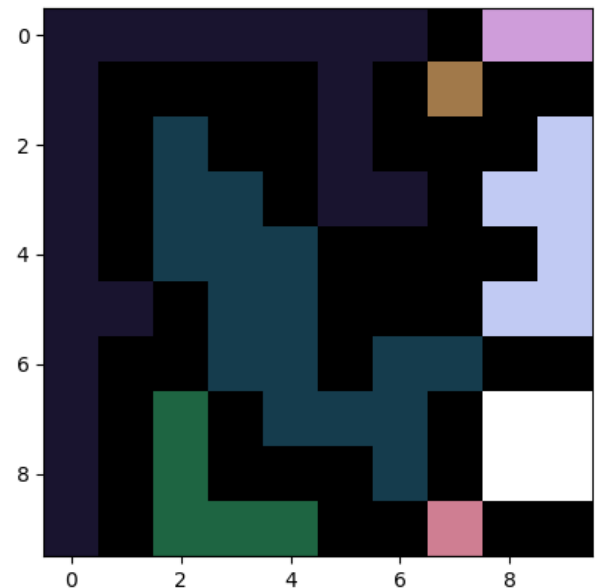
Firstly, a  $l \times l$  array is generated. The first column is set equal to zero. Then a loop starts which checks all the elements of this array. In every turn, a random number is generated and if it is less than the probability, the element will be colored with the current color amount, and then this amount is increased by 1. Then the neighbors are checked. If they are both colored, the element will be colored with the minimum amount. Then another loop starts which checks every previous element of the array, and if they had the other color, changes it to the minimum one. Lastly, if only one of the neighbors were colored, the element would get the same color.

Now in order to check whether percolation happens, the last column is checked. If 1 is found, percolation has happened. Lastly the lattice is plotted.

### 1.2 results



Percolation occurrence in  $10 \times 10$  lattice



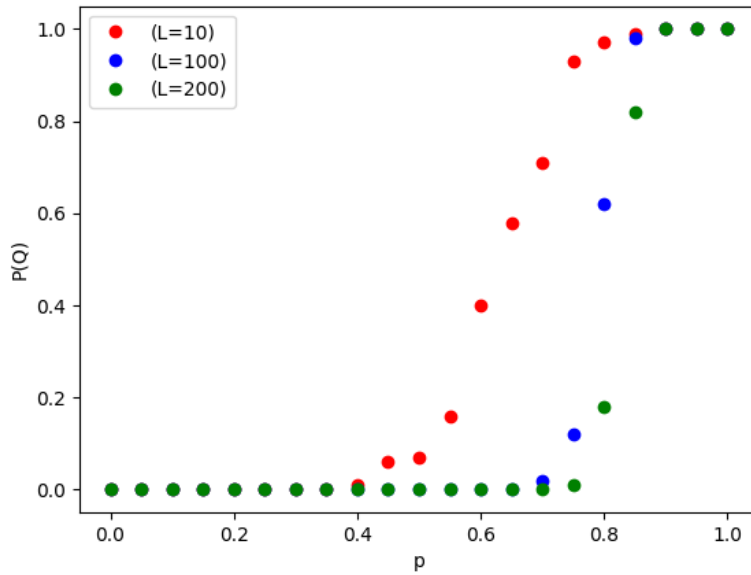
Percolation not happening in  $10 \times 10$  lattice

## 2. Percolation Probability

### 2.1 Main Code

Firstly, a loop starts with 20 rounds. Each round corresponds to a probability where another loop with 100 rounds starts. In this loop, a lattice is colored using the hoshen-keopman algorithm and then percolation occurrence is checked. If occurred, the round will be counted. After 100 rounds ends, the total number of rounds with percolation is counted and divided by 100. This amount corresponds to the probability of percolation at a specific  $p$ . So this probability is calculated all over the range and finally plotted.

### 2.2 results

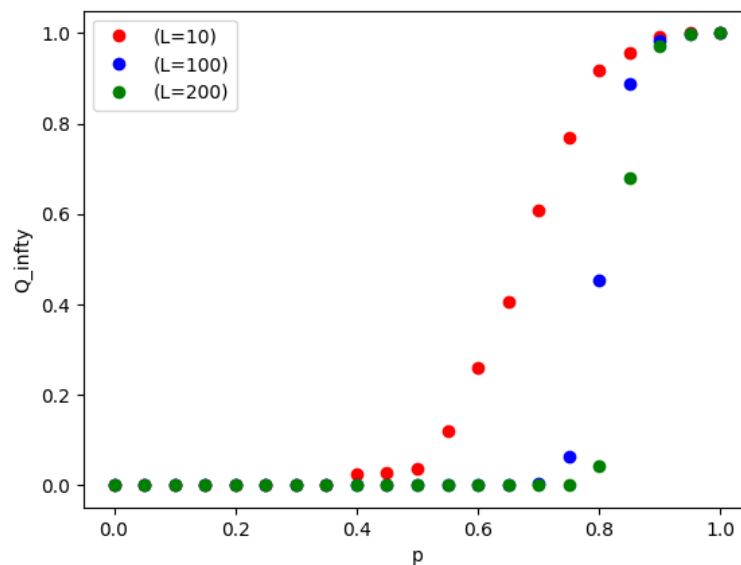


### 3. Probability of Existing in Infinity Cluster

#### 3.1 Main Code

Firstly, a loop starts with 20 rounds. Each round corresponds to a probability where another loop with 100 rounds starts. In this loop, a lattice is colored using the hoshen-keopلمان algorithm and then percolation occurrence is checked. if happened, the size of the infinity cluster is divided by the number of all non-zero elements. This value corresponds to the probability that a nonzero element is in the infinity cluster. These values are all added together for each round and after 100 rounds ends, is divided by 100. This amount corresponds to the mean probability at a specific  $p$ . so this probability is calculated all over the range and finally plotted.

#### 3.2 results



#### 4. Correlation Length

##### 4.1 Functions Description

Com

This function calculates the center of mass of each cluster by summing the coordinated of all elements with the same color and calculating their average, it returns two arrays which are x and y coordinates of center of mass for each color.

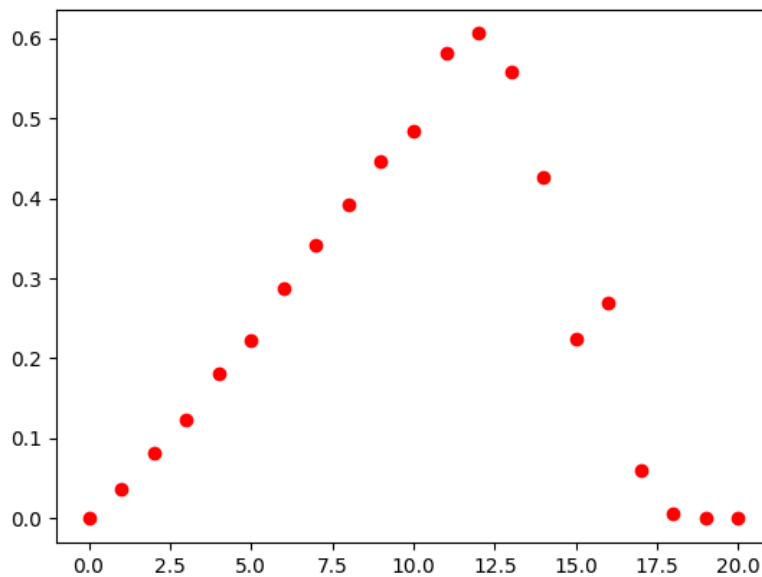
Gy\_rad

This function calculates the gyration radius for each cluster using the coordinates of the elements and the aenter of mass of the cluster. Lastly, returns an array of them for every color,

##### 4.2 Main Code

Firstly, a loop starts with 20 rounds. Each round corresponds to a probability where another loop with 100 rounds starts. In this loop, a lattice is colored using the hoshen-keopلمان algorithm and then percolation occurrence is checked. Then the gyration radius of each color is calculated using gy\_rad function. If an infinity cluster exists, gyration radius of it is deleted from the result. Then the mean of this array is calculated, which is the correlation length. All the correlation lengths for each p are summed and their average is calculated. Lastly, an array of correlation lengths is printed and plotted over the range of probabilities.

##### 4.3 Results



I was only able to plot the graph for l=100 because my run time was so long for larger amounts.