PHYS 8601 – Problem 2

Yier Wan

February 14, 2018

1 Introduction

Percolation refers to the movement and filtering of fluids through porous materials. In this work, typically, we are going to consider a $L \times L$ square lattice with free edge boundary conditions. The sites of this lattice can be occupied randomly with a concentration probability p. The occupied sites are jointed to build clusters. If there exists at least one cluster having a contiguous path through the lattice region from the top to the bottom or the left to the right, we say this occupied lattice have an"infinite" cluster. Hoshen-Kopelman algorithm is applied to do cluster counting. First, we study how the probability of having an infinite cluster varies with p and L. Second, we estimate the percolation limit p_c for $L = \infty$. Finally, we investigate the cluster size distribution for an infinite lattice with $p = 0.59$.

2 Calculation details

For each value of p, the probability of having an infinite cluster (P_{span}) is determined by generating many realizations of the lattice and counting the fraction of those cases in which an infinite cluster is produced. In this work, 100 realizations of the lattice are simulated in order to calculate the fraction. For each combination of p and L, 10 independent runs with different random number streams are carried out and the mean value and statistical error are estimated based on the 10 runs. In this work, we consider an $L \times L$ square lattice with free edge boundary conditions. It means the boundary does not involve any kind of connection between the end of a row and any other row on the lattice.

3 Algorithm

1.a The lattice is stored in `array(1:L,1:L)` with entries 1 for occupied site and 0 for unoccupied site.
1.b Define the array of labels `label(1:L,1:L)` 1.c Define a helper array `size(1:L*/2)` (There can be no more than L*L/2 labels.)
1.d Set the number of clusters `ncluster=0`.

2. Initially, the lattice is empty. Each site of this lattice is ccoupied randomly with probability p. And the end of this sweep, the actual concentration of filled sites is liable to be different from p, so a few sites will need to be randomly occupied or unoccupied until we reach the concentration probability p.

3. Apply Hoshen-Kopelman algorithm to label the sites.
Scan the lattice sequentially row-wise, from the top to the next, from left to right.
For next occupied site `x`, if the neighbor above and the neighbor to the left
both are empty (or not existing, such as the corner site) $\rightarrow$ new cluster label n, `size(n)++`.
only one of them is occupied and has clusrer label `k` $\rightarrow$ set this set to `k`, `size(k)++`.
both are occupied and have the same label `k` $\rightarrow$ set this site to `k`, `size(k)++`.
both are occupied but have conflicting cluster numbers `n` and `m` with `n` $<$ `m`$\rightarrow$ union the two clusters

to label `n`, `size(n)+=size(m)` and `size(m)=0`.

4. Scan `size` for cluster size distribution.

5. Check if spanning cluster exists.

We adopt $L = 10$ and $p = 0.6$ in a sample run. Results are in Figure. 1. 6 clusters are identified correctly.

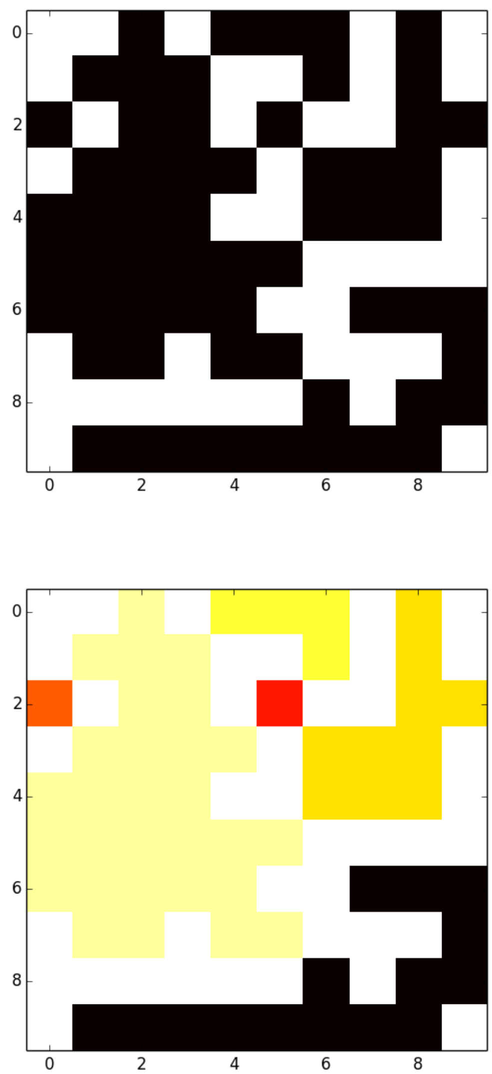

Figure 1: Upper panel: occupation of the lattice (black block represents occupied site); lower panel: identification of the lattice by H-K method (different color represents different cluster).

4 Results and discussion

First we adopt three different values of L (50, 100,150) in the simulation and vary p from 0.3 to 0.9 by step 0.1. Results are in Table. 1. We find that the probability of having an infinite cluster is zero for small values of p and one for big values of p. Then we vary p by a finer step 0.01 from 0.55 to 0.62. The results show that the probability of having an infinite cluster increases dramatically from almost zero to nearly one. Another 9 simulations are drawn, so that we estimate the mean value and standard error of the probability of having an infinite cluster based on the 10 simulations in total. From Figure. 2, we can clearly see the "big jump" happens when p is close to 0.6. Also, the curve becomes more similar to a step function, as the value of L increases. Figure. 3 plot two examples of cluster configurations for $L = 50$ with different values of p.

Table 1: Simulated results

p	L50	L100	L150	L200	L250
0.90	1.00	1.00	1.00	1.00	1.00
0.80	1.00	1.00	1.00	1.00	1.00
0.70	1.00	1.00	1.00	1.00	1.00
0.65	1.00	1.00	1.00	1.00	1.00
0.62	0.98	0.99	1.00	1.00	1.00
0.61	0.94	0.98	1.00	1.00	1.00
0.60	0.781(±0.011)	0.842(±0.010)	0.887(±0.007)	0.90	1.00
0.59	0.655(±0.013)	0.624(±0.011)	0.596(±0.019)	0.43	0.00
0.58	0.482(±0.018)	0.353(±0.016)	0.254(±0.014)	0.14	0.00
0.57	0.317(±0.011)	0.134(±0.011)	0.057(±0.005)	0.02	0.00
0.56	0.15	0.02	0.00	0.00	0.00
0.55	0.09	0.01	0.00	0.00	0.00
0.50	0.00	0.00	0.00	0.00	0.00
0.40	0.00	0.00	0.00	0.00	0.00
0.30	0.00	0.00	0.00	0.00	0.00

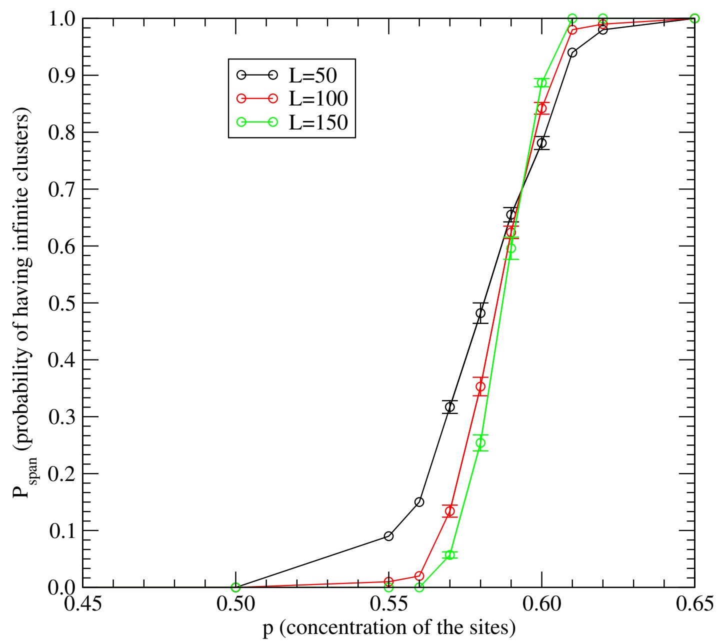

Figure 2: The probability of having an infinite cluster as a function of p for three different values of L.

The percolation limit p_c for $L \to \infty$ has such characteristics:
if $p > p_c$, the probability of having an infinite cluster is one;
if $p < p_c$, the probability of having an infinite cluster is zero.
Because the limit of computation capability, we can only simulate with several big L values and observe when $P_{span}(p)$ behaves like a step function. The last column of Table. 1 is calculated by $L = 250$. $P_{span}(p \leq 0.59) = 0$ and $P_{span}(p > 0.59) = 1$. This result satisfies the infinity condition for $L \to \infty$. So we can take $L = 250$ as parameter for an "infinite lattice" in the next problem and the percolation limit p_c should be in region (0.59, 0.60). The estimated percolation limit is 0.595 with an error 0.005.

Figure 3: Upper panel: cluster configuration for $L = 50$ and $p = 0.6$; lower panel: cluster configuration for $L = 50$ and $p = 0.3$.

We know the relation between the number of cluster per lattice site which contain n sites (C_n) and cluster size (n) is

$$C_n(p) = an^{-\tau}. \tag{1}$$

C_n is calculated by

$$C_n(p) = \frac{\text{number of cluster of size } n}{L * L}, \tag{2}$$

where we take $p = 0.59$ and $L = 250$. In this run, the occupied sites build 1799 clusters, while no infinite cluster exists. Figure. 4 shows the cluster size distribution. In order to estimate the value of τ, we rewrite 1 into a linear form

$$\log C_n(p) = \log a - \tau \log n, \tag{3}$$

where $\log C_n(p)$, $\log n$, $-\tau$ and $\log a$ are y, x, b and d of the standard linear relation $y = bx + d$ separately. The linear fitting is done by applying a statistical software *R*. *R* command `summary(lm(y~x))` is used and it returns estimated coefficients and standard error. The fitted results are in Table. 2. The estimate τ is 0.6328±0.0628, which is a very small value. From Figure. 4, we can see that although the fitted curve cross most of the data points, it cannot repeat the real simulated results. The main reason is that Eq. 1 represents an ideal exponential decrease, which requires C_n nearly and smoothly to be zero as n close to infinity. However, in the simulation, the two smallest values of C_n are 0 and $\frac{1}{L*L}$. So the right tail of distribution obtained from Eq. 1 and simulation behave differently.

Table 2: Simulated results

	Estimated value	Standard deviation
$\log a$	-3.2653	0.1207
$-\tau$	-0.6328	0.0628

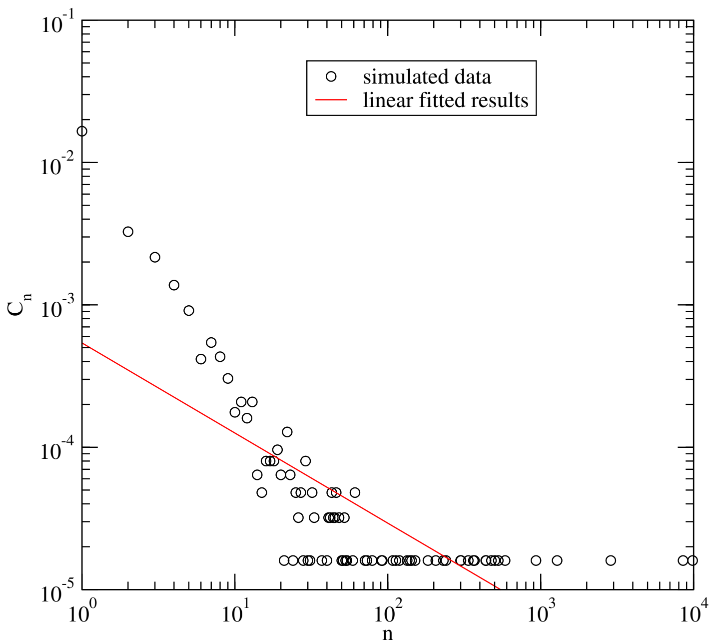

Figure 4: The number of cluster of size n normalized by the lattice size as a function of the value of n.