20.1

The Percolation Model

Question 1

Proof of $\theta(p)$ being non-decreasing

First, let's prove that for all n,

$$\theta_n(p_1) \leqslant \theta_n(p_2)$$

if $p_1 \leqslant p_2$. We proceed by induction. For $n = 1$, want to calculate $\theta_1(p)$. As we start from $(0, 0)$, we have two options: going up to $(0, 1)$ or stepping to the right to $(1, 0)$. We cannot make it to C_1 if and only if both of these routes are closed, hence we have $\theta_1(p) = 1 - (1-p) \cdot (1-p) = p \cdot (2-p)$ which is an increasing function on the interval $[0, 1]$.
Now suppose that our result holds for $n - 1$, ie.:

$$\theta_{n-1}(p_1) \leqslant \theta_{n-1}(p_2)$$

The event $\{C_{n-1} \neq \varnothing\}$ consits of disjoint events A_i where A_i describes a possible set of points for C_{n-1} (For example we can take A_1 to be the event when $C_{n-1} = \{(0, n-1)\}$ etc.) When we are in $A_i = \{C_{n-1} = \{y_1, y_2, ..., y_m\}\}$, to achieve $\{C_n \neq \varnothing\}$ we need to have at least one of the edges starting from y_1, y_2, ... , y_m to be open. This has probability $1 - (1-p)^{2m}$. This is a increasing function on $[0, 1]$ once again, hence if $p_1 \leqslant p_2$ then

$$\mathbb{P}_{p_1}(C_n \neq \varnothing, A_i) \leqslant \mathbb{P}_{p_2}(C_n \neq \varnothing, A_i)$$

As we have this for all i, we will get that for

$$\mathbb{P}_p(C_n \neq \varnothing) = \sum_{i=1}^{N} \mathbb{P}_p(C_n \neq \varnothing, A_i)$$

we have

$$\mathbb{P}_{p_1}(C_n \neq \varnothing) \leqslant \mathbb{P}_{p_2}(C_n \neq \varnothing)$$

as required. As this is true for all n, by the Squeeze theorem we will have

$$\lim_{n\to\infty} \theta_n(p_1) \leqslant \lim_{n\to\infty} \theta_n(p_2)$$

which is the same as

$$\theta(p_1) \leqslant \theta(p_2)$$

Proof of $\theta_n(p)$ being decreasing in n

Let's fix $p \in (0, 1)$. Using the same notation as above, we have that

$$\mathbb{P}_p(C_n \neq \varnothing | A_i) = \mu_i < 1$$

where μ_i is the probability of having the right edges open when we are under A_i. Then using Bayes' formula, we have

$$\mathbb{P}_p(C_n \neq \varnothing, A_i) = \mathbb{P}_p(C_n \neq \varnothing | A_i) \cdot \mathbb{P}_p(A_i) = \mu_i \cdot \mathbb{P}_p(A_i) < \mathbb{P}_p(A_i)$$

Now summing over all possible $A_i's$,

$$\sum_{i=1}^{N} \mathbb{P}_p\big(C_n \neq \varnothing, A_i\big) < \sum_{i=1}^{N} \mathbb{P}_p\big(A_i\big)$$

As the events A_i are disjoint and as $\bigcup_{i=1}^{N} A_i = \big\{C_{n-1} \neq \varnothing\big\}$, we get

$$\mathbb{P}_p\big(C_n \neq \varnothing\big) < \mathbb{P}_p\big(C_{n-1} \neq \varnothing\big)$$

as required.

Error of estimation

We want to tell the likely size of the error $\hat{\theta}_{m,n}(p) - \theta_n(p)$. Let's denote the true parameter $\theta_n(p)$ by P_n. Then, we can think of the $I_n(j)'s$ as independent Bernoulli random variables with mean P_n. Taking

$$\hat{\theta}_{m,n}(p) = \frac{\sum_{j=1}^{m} I_n(j)}{m}$$

is the Maximum Likelihood Estimator of P_n. By the Central Limit theorem, we have

$$\sqrt{m}\big(\hat{\theta}_{m,n}(p) - P_n\big) \to N\big(0, \sigma^2\big)$$

where σ^2 is the variance of I_n. So have $\sigma^2 = P(1-P)$. Then we can write

$$\sqrt{m}\frac{\big(\hat{\theta}_{m,n}(p) - P_n\big)}{\sqrt{P(1-P)}} \to N\big(0, 1\big)$$

as $m \to \infty$. Then a 95% confidence interval for P_n would be

$$\left[\hat{\theta}_{m,n}(p) - z_{0.975}\sqrt{\frac{\hat{\theta}_{m,n}(p)\cdot(1-\hat{\theta}_{m,n}(p))}{m}}, \hat{\theta}_{m,n}(p) + z_{0.975}\sqrt{\frac{\hat{\theta}_{m,n}(p)\cdot(1-\hat{\theta}_{m,n}(p))}{m}}\right]$$

the size of the likely error being $z_{0.975}\sqrt{\frac{\hat{\theta}_{m,n}(p)\cdot(1-\hat{\theta}_{m,n}(p))}{m}}$. As $z_{0.975} \approx 1.96$ and $\hat{\theta}_{m,n}(p)\cdot(1-\hat{\theta}_{m,n}(p)) \leqslant 1/2$, we have

$$\left|\hat{\theta}_{m,n}(p) - \theta_n(p)\right| < \frac{1.386}{\sqrt{m}}$$

So for example if we want to have a likely error less than 0.05, we would take $m \geqslant 770$.

Question 2

For plotting the function $\hat{\theta}_{m,n}(p)$, I have made two algorithms, one which creates an instance of the model for given n and calculates the values of $Z(y)$ for $y \in Q_n$ (*instance.m*), and one which then plots it the function between 0.5 and 0.75 (*plotmyfn.m*) Let's calculate the complexity of the first algorithm. Firstly, it is known, that generating a pseudo-random number has complexity O(1). Let's say that it takes k basic operations. In each step of the for loop, we have to generate

$2i$ random numbers, this taking $2i \cdot k$ operations. Then, we will find the values $Z(y)$ for the $i+1$ points in Q_i. Finding the value for one new point takes approximately 4 operations, so this gives $4 \cdot (i+1)$ in total. As we run from $i = 1$ to n, the total complexity is

$$\sum_{i=1}^{n} 4(i+1) + 2ik = 4n + (4+2k)\frac{n(n+1)}{2} = O(n^2)$$

Now, to plot the actual function, we will need m insances, and need to calculate $\hat{\theta}_{m,n}$ for l points between 0.5 and 0.75.
- Checking that Q_n has a point with $Z(y) < p$ has complexity $O(n)$. This is contained in *haselement.m*
- For each instance, we need to generate the numbers $Z(y)$, which has complexity $O(n^2)$.
- Then, for each point p, *haselement.m* is calculated, that has complexity $O(pn)$.
- If we have m instances, that gives $O(n^2 m + pnm)$ for the complexity.

Hence we can see that the running time depends greately on n but it also does on m as a linear factor. As we have calculated in Question 1, using m large enough, we can guarantee that the error is less than 0.05. So let's use $m = 800$, and $n = 1000$. The graph we get can be seen in Figure 1.

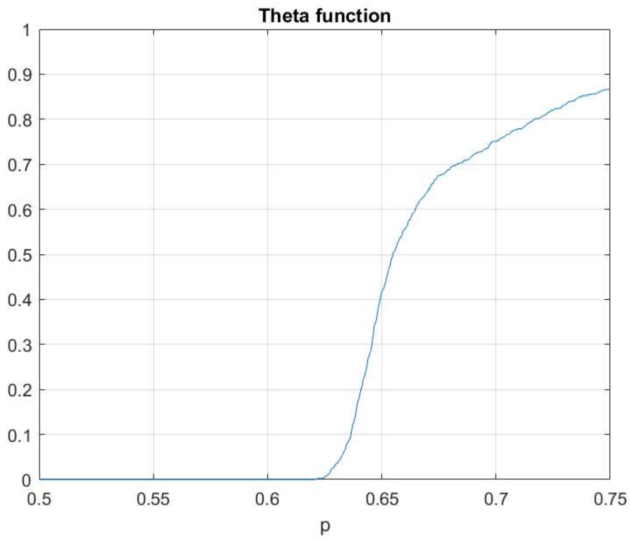

Figure 1: Approximation of the function $\theta(p)$ with $m = 800$, $n = 1000$

Question 3

To investigate the relationship between the estimate $\hat{p}_c$ and n for fixed m we will calculate the estimate for the ciritical value for different n and draw a graph. To make the program a little bit faster, we can use the same instance to calculate all the values I_1, I_2, ..., I_n. To do this we just have to store all random numbers assigned to the vertices, instead of storing only the values in Q_n (as in Question 2). This is done in $'instance2.m'$. The algorithm used to do the plotting can be found in $'criticalpointplot.m'$.

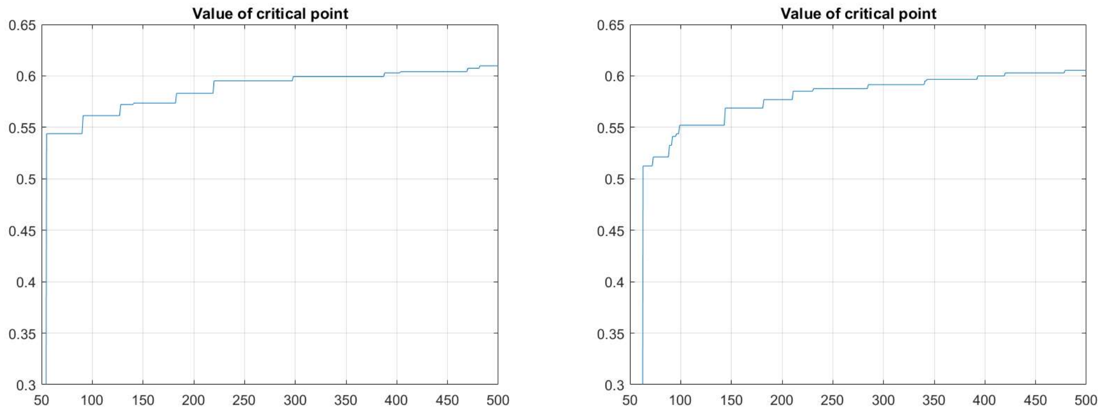

Figure 2: The value of $\hat{p}_c$ for n between 50 an 500 and for $m = 100$ and 1000

We can see on Figure that for fixed m, $\hat{p}_c$ is an increasing function of n. The reason for this is that as we have shown before that $\hat{\theta}_{n,m}(p)$ is a decreasing function of n for fixed p (the is a very similiar proof to showing that $\theta_n(p)$ is decreasing). This means that for $n_1 < n_2$,

$$\hat{\theta}_{n_1,m}(p) \geqslant \hat{\theta}_{n_2,m}$$

for all $p \in (0,1)$. Then,

$$\hat{\theta}_{n_1,m}(p) = 0 \Rightarrow \hat{\theta}_{n_2,m}(p) = 0$$

then this implies

$$\hat{p}_c(n_1) \leqslant \hat{p}_c(n_2)$$

which is what we wanted.

If we fix n and vary m, we will see that the estimated critical value will decrease slightly with m. Also, it will fluctuate less. The reason for this is that when m is larger, the error of estimation is less, and it is more likely that we have found the right value of p_c.

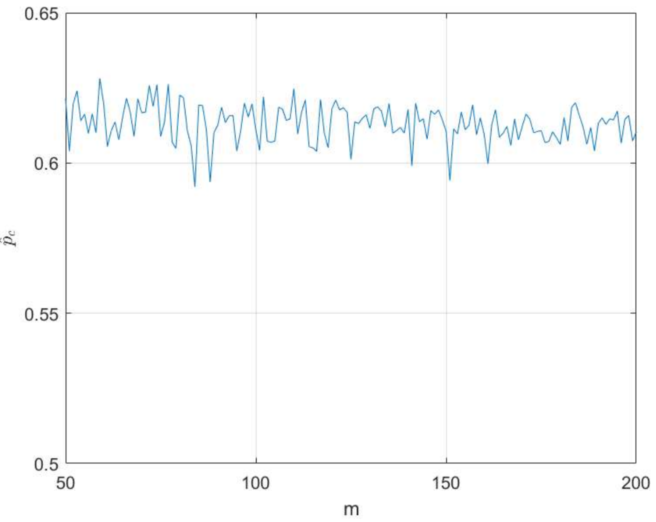

Figure 3: The value of $\hat{p}_c$ for m between 50 and 200 and for fixed $m = 500$

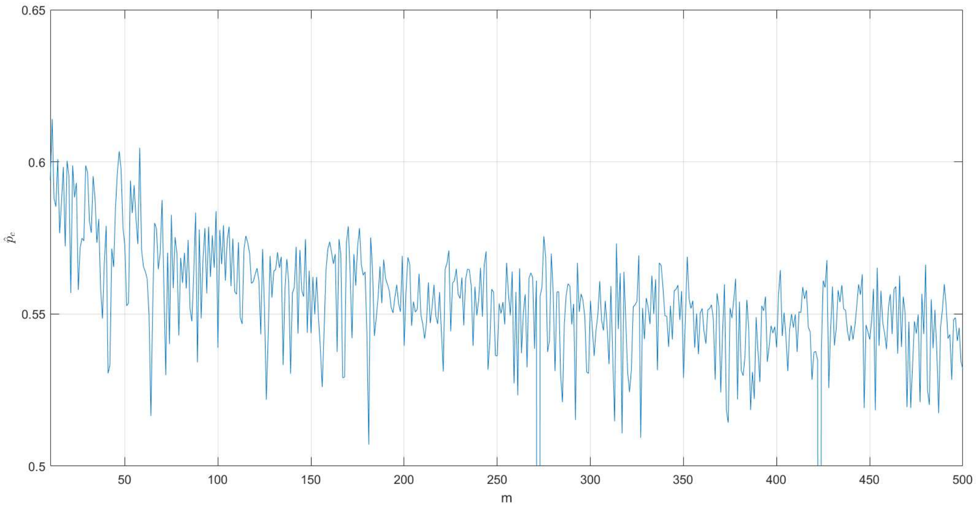

Figure 4: The value of $\hat{p}_c$ for m between 50 and 500 and for fixed $n = 100$

Question 4

When we are estimating a limit, the first idea that comes to mind is to choose n as big as possible as to reach smaller errors in the estimation. But this in not entirely the right approach.
We can see from Figure 2, that for $n = 60$ and $m = 1000$, the estimated value of p_c is already larger

than 0.5. Hence it follows that choosing $p = 0.3$ or 0.4, it is not advisable to choose large n, as by precision error, we would already get $\hat{\theta}_{n,m} = 0$. To get the precise answer, it is sensible to choose m as large as possible, as we are more likely to get the true value for $\hat{\theta}_{n,m}$ and not 0. From Question 2, we have seen that the complexity of the algorithm that estimates $\theta_n(p)$ is $O(n^2m+pnm) = O(n^2m)$ (we are taking one value for the probability, so $p = 1$). Therefore, for $p = 0.3$, I have chosen $n = 20$ and $m = 400000$. For bigger n, we would also need a much larger m, but that would raise the computational time immensely. The estimation is $\gamma = 0.5251$

```
>> [~, value] = thetavalues(400000, 20, 0.3, 0.3, 1);
>> 1/20*log(value)

ans =

   -0.5251
>> [~, value] = thetavalues(100000, 40, 0.4, 0.4, 1);
>> 1/40*log(value)

ans =

   -0.2878
>> [~, value] = thetavalues(50000, 70, 0.5, 0.5, 1);
>> 1/70*log(value)

ans =

   -0.1179

>> [~, value] = thetavalues(5000, 450, 0.6, 0.6, 1);
>> 1/450*log(value)

ans =

   -0.0149
```

Figure 5: Estimating γ for $p = 0.3, 0.4, 0.5, 0.6$

As we raise the value of p, we can also raise n. But in return, we need to lower m to get sensible computatuional times. The estimations we get for γ can be seen on the Figure above. We can notice that choosing p closer to p_c, γ gets closer to 0.