FYS4460 -
Third compulsory project

Mathilde Nygaard Kamperud

May 7, 2013

1 Generating percolation cluster

1.1 MATLAB

To generate and visualize percolation clusters, we use MATLAB. We generate a binary matrix (M) of size $L \times L$, with elements either 0 or 1 (unoccupied of occupied). A variable p is defined as the probability for a site to be occupied. The matrix, M, is made by letting each entry/site be drawn randomly (from a uniform distribution) between 0 and 1. Then we let all the sites with value smaller than p be 1, and let all the sites larger than p be 0. We are interested in the areas in M where the occupied sites are connected. They are connected if they are left or right-neighbours, or south or north-neighbours, see fig 1.

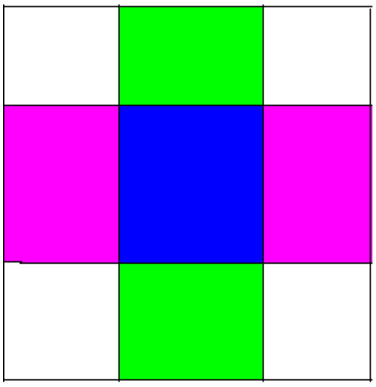


Figure 1: This drawing shows the possible connection between sites. The blue site in the middel is connected in a left-right manner with the pink sites, and in a north-south manner with the green sites. The blue site is NOT connected to the white sites.

The connected sites are called clusters, and we can label the clusters in matrix M by the convenient MATLAB-function bwlabel. From this function we receive a matrix lw, which looks like matrix M, but the occupied entries has a value corresponding to the cluster it is a part of. Example:

$$M = \begin{bmatrix} 1 & 1 & 0 & 0 & 0 \\ 1 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 1 \\ 0 & 0 & 0 & 0 & 1 \end{bmatrix}. \tag{1}$$

By bwlabel and this matrix we receive

$$lw = \begin{bmatrix} 1 & 1 & 0 & 0 & 0 \\ 1 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 2 & 2 \\ 0 & 0 & 0 & 0 & 2 \end{bmatrix}.$$

We see that the cluster to the left is marked as cluster number 1 and the cluster to the right is cluster number 2. This can be visualized in colors by a matlab-function called label2rgb. Different clusters get different colors. Figure 2 is M (eq. 20) visualized with colors.

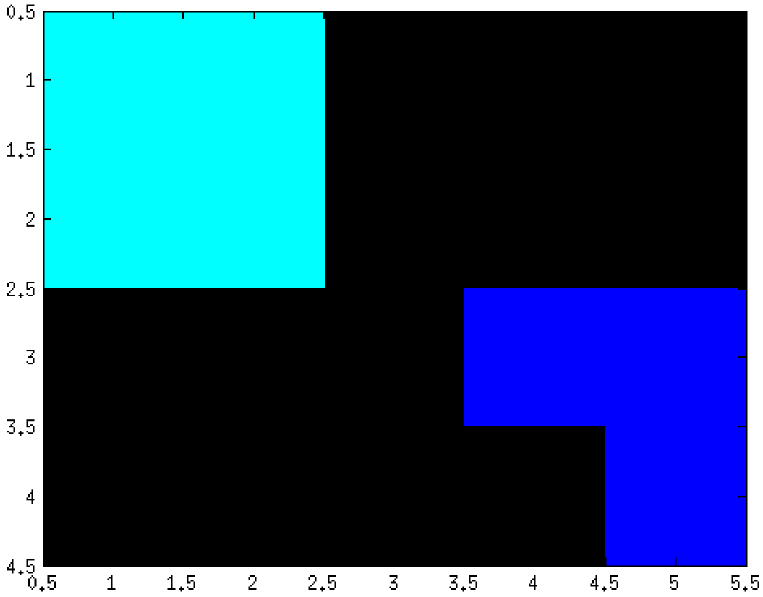


Figure 2: Visualisation of the clusters in eq. 20

1.2 Spanning clusters

The clusters visualized in fig 2 are finite clusters, which means that they do not connect one side to another (connect right to left, or south to north). The clusters that actually do connect are called spanneing clusters. If a system has a spanning cluster, we say that the system is percolated. We want to find the probability for a site to be in a spanning cluster, as a function of system size (L) and the value of the probability, p; P(p,L). This can be done by performing an experiment; for a given system size L, we make the percolation matrix *lw* as described above (with many different values of p), and count how many sites are in a spanning cluster and divide by the total number of sites. This gives us $P_{exp}(p, L)$ for this one experiment. We repeat the experiment many times, and average over P_{exp} to find $P(p, L)$ as a function of p. This should be done for a few different values of L; the larger the system, the better. I did this in my program, and the result is in fig 3.

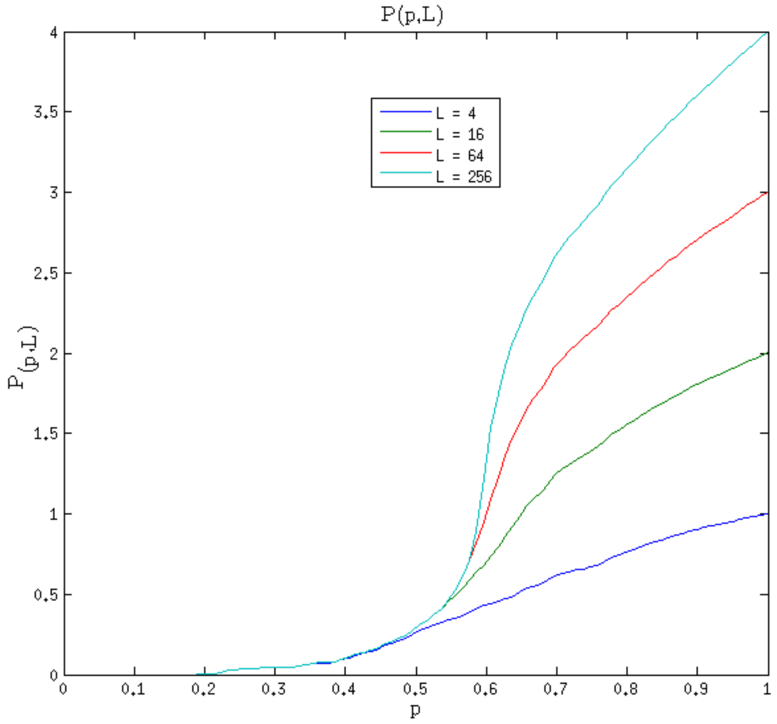


Figure 3: These graphs show the probability for a site to be in a spanning cluster.

The challenge when performing such a experiment is to count the sites in spanning clusters; how do you know if a site is in a spanning cluster or not? My method is to look at the colomn furthest to the right (in lw) and the colomn furthest to the left and see what numbers they had in common. If the had a number in common, say 4, we know that cluster number 4 stretches from the left side of the system to the right. We do the same thing with the top and bottom row. To do this I used the function called intersect in matlab. Now we have a set of numbers that tells us what clusters actually span the system (maybe cluster number 4 and 2). We must be careful not to overcount, because a spanning cluster can span both from top to bottom, and from left to right, which will count as twice. Therefore we use the matlab-function called union to find the union of numbers that we found spanned from right to left and from top to bottom. Now we have a list of the clusters that span, and all we have to do is run through our system and check if the sites belong to any of them. If they do, we count them.

1.3 Finding β

The probability p_c is the so called percolation threshold. We know that when $p > pc$, the probability $P(p, L)$ for a gien site to belong to the percolation cluster has the form

$$P(p, L) \sim (p - p_c)^\beta.$$

We can use our program to find β (an approximation, of course), given that we know the percolation threshold. Luckily, we do; $p_c = 0.59275$. If we assume $P(p, L) = (p - p_c)^\beta$, and take the logarithm

$$\log(P(p, L)) = \beta \log(p - p_c)$$

and set $x = (p - p_c)$

$$\log(P(x, L)) = \beta x$$

we see that it is a linear function with a slope equal to β. This is good news! All we have to do now is to plot the logarithm as described, and find a linear approximation. Then we say β is the slope of the linear approximation. In fig. 4 we see the logarithm of $P(x, L)$ for three different system sizes (L), together with the linear approximation. It appears that the values we get for β decreases as we increase the system size

(I am not sure why the different system sizes get different y-intercept). The largest system size I tried for was $L = 5000$, and the result is in fig. 5. This gave me $\beta = 0.206$.

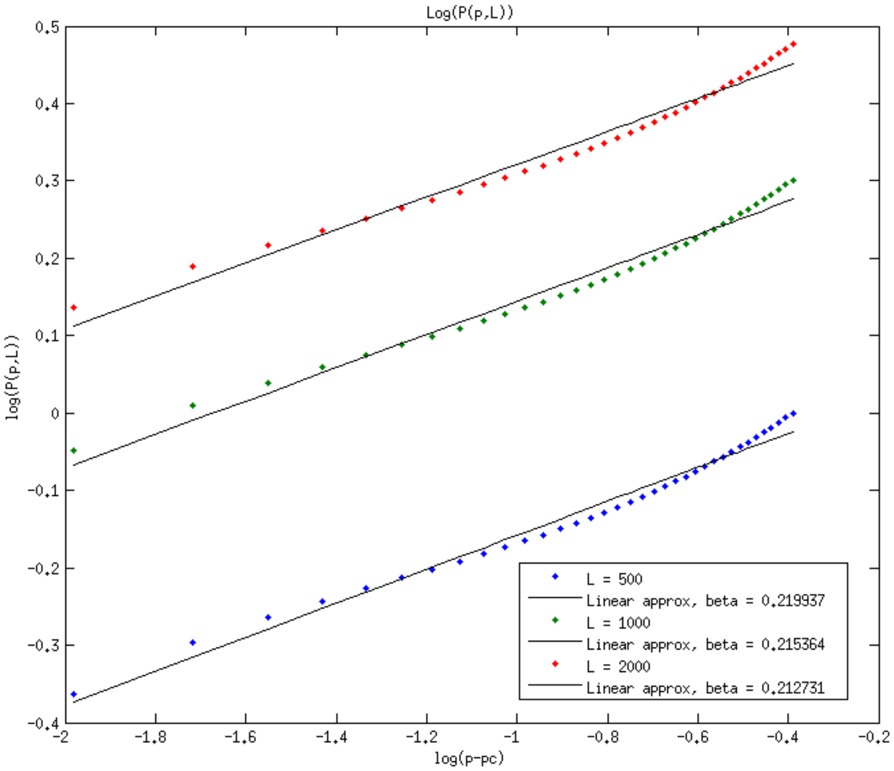


Figure 4: The logarithm of $P(x, L)$ for three system sizes (L), together with a linear approximation where the slope is an approximation to β.

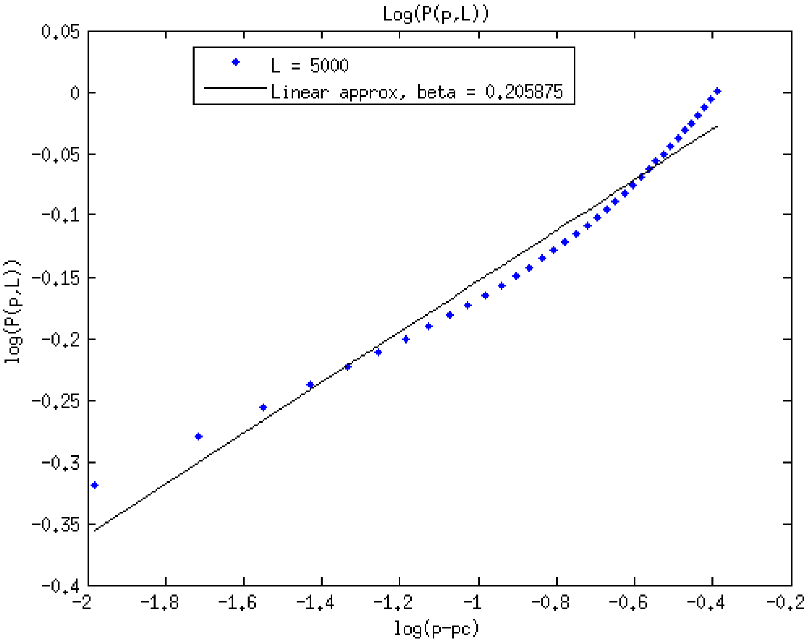


Figure 5: Finding β with $L = 5000$.

2 Determining the exponent of power-law distributions

2.1 Determine the distribution function

We generate the set of data-points in MATLAB:

```
z = rand(1e6,1).^(-2).
```

We will determine the distribution function $f_Z(z)$ which is on the form $f(u) = Au^\alpha$. We can find the prefactor A, because it is a normalization factor:

$$\int_1^\infty f(u)du \equiv 1$$
$$A\int_1^\infty u^\alpha du \equiv 1$$
$$\frac{A}{\alpha+1}\left[u^{\alpha+1}\right]_1^\infty \equiv 1$$

The last equation tells us something about α; for $f(u)$ to be normalizable, we must have $\alpha < -1$. If we assume this:

$$\frac{A}{\alpha+1}\left[0-1\right] = -\frac{A}{\alpha+1} \equiv 1A = -(\alpha+1)$$

So, our distribution has the form

$$f(u) = -(\alpha+1)u^\alpha.$$

The cumulative distribution, $P(Z < z)$, is found by

$$P(z) = \int_1^z f(u)du = -(\alpha+1)\int_1^z u^\alpha du = -\left[z^{\alpha+1} - 1\right] = 1 - z^{\alpha+1}$$

If we know the cumulative distribution, $P(z)$, we can find α (and thereby find $f_Z(z)$) by

$$z^{\alpha+1} = [1 - P(z)]$$
$$(\alpha+1)\log z = \log[1 - P(\log(z))]$$
$$ax = \log[1 - P(x)].$$

where $x = log(z)$ and $a = \alpha + 1$. We see that $\log[1 - P(x)]$ should be a linear function, with slope $a = \alpha + 1$. If we can find the slope, a, we know $\alpha = a - 1$. Let's get to it, first we find the cumulative distribution, $P(x)$. With $x = \log(z)$, we get logarithmic binning. I find $P(x)$ by making setting x to be a logspace between the smallest value in z, and the largest value in z. I then go through the x-values, and set P_i to be the number of values in z smaller than x_i. The result can be seen in figure 6.

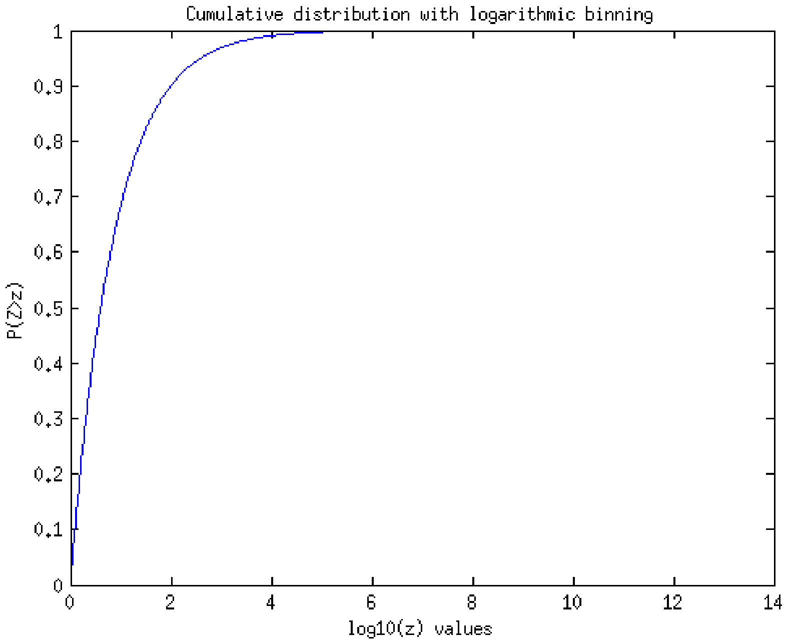


Figure 6: The cumulative distribution with logarithmic binning.

Okey, now we have to take the logarithm of this function and hope we get a straight line. Figure 7 show the logarithm together with a linear approximation ax.

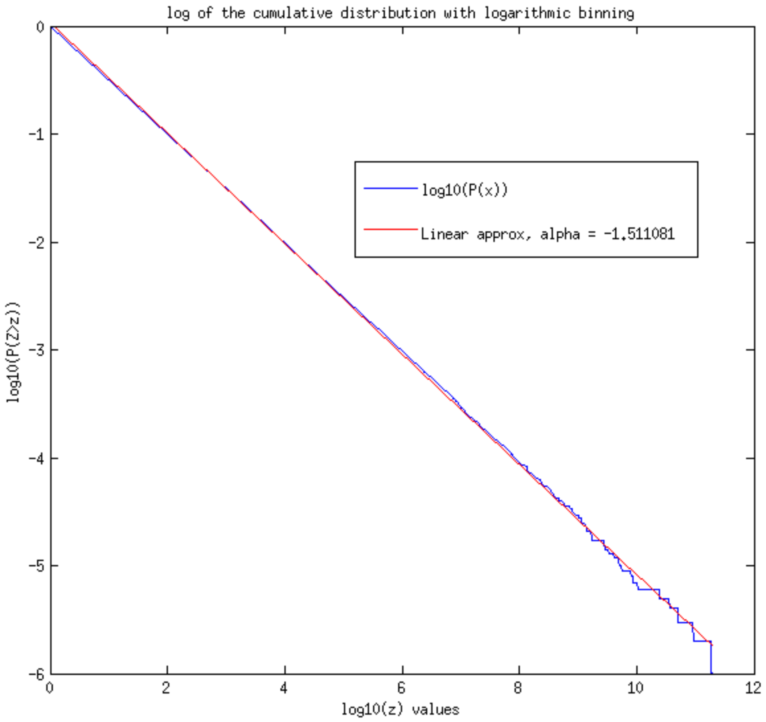


Figure 7: The logarithm of the cumulative distribution with logarithmic binning. The estimated α is -1.511.

The distribution function is $f(z) = (0.511)z^{-1.511}$. Figure 8 shows the distribution function found by taking the derivative of the cumulative distribution and the function with the estimated α.

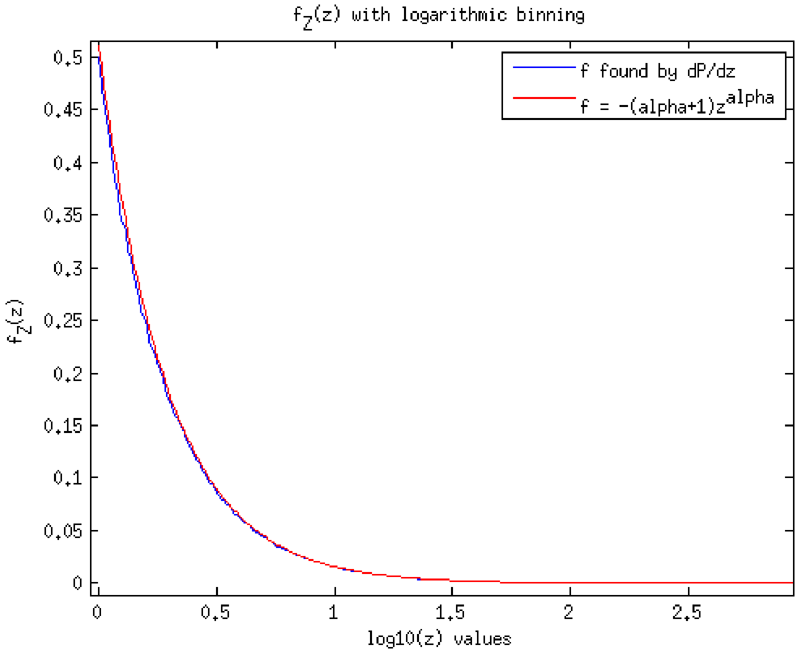


Figure 8: The distribution function and the estimated function.

3 Cluster number density

The cluster number density, $n(s,p)$ is defined by

$$n(s,p) = \frac{N_s}{L^d} \tag{2}$$

where N_s is the number of clusters with size s, L is the size of the system and d is the dimension. To find N_s all we have to do is count, but the spanning clusters should not be included in the counting. We will estimate $n(s,p)$ for a sequence of p values approaching p_c from above and below. The result can be seen in figure 9 and figure 10

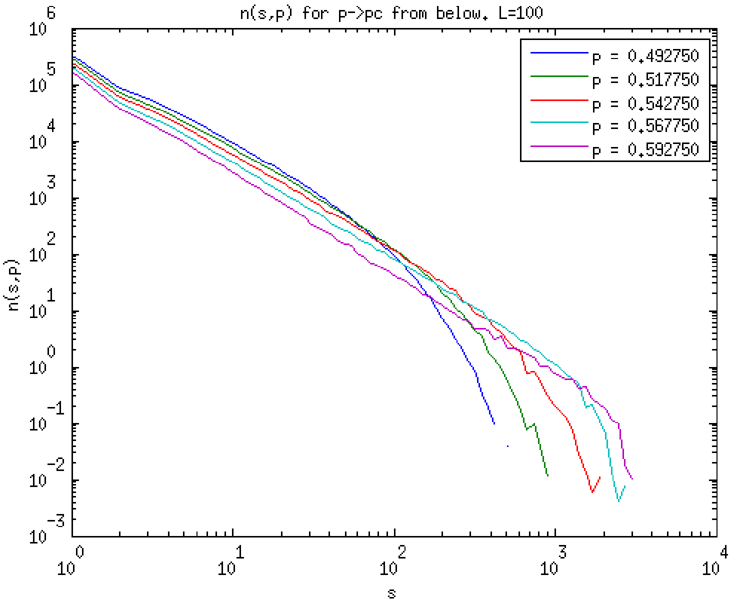


Figure 9: The cluster number density for p approaching pc from below.

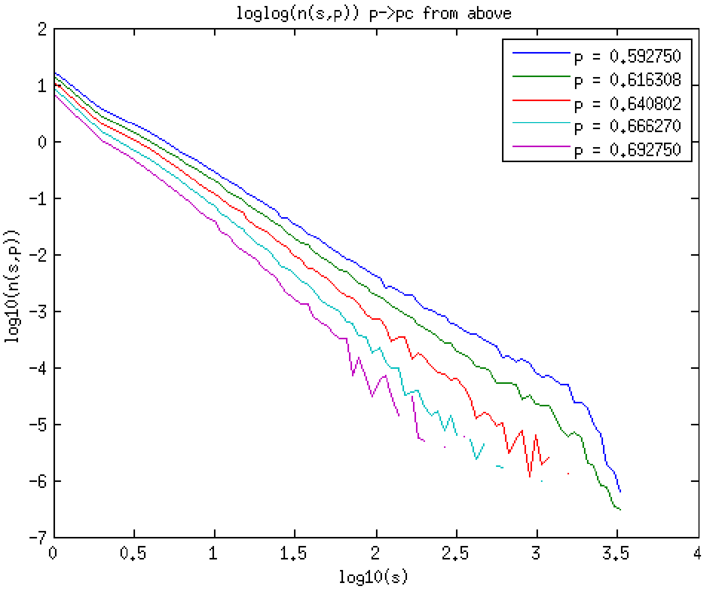


Figure 10: The cluster number density for p approaching pc from above.

We see that $n(s,p)$ plotted with logarithmic axes looks linear, exept for large s. For large s it falls off, which is due to the fact that we have a finite size system (size L=100 in this case). This linear behavior

tell's us that $n(s,p)$ follows a power-law function.
The cluster number density for $p = p_c$ is proportional to the size to some power $-\tau$:

$$n(s,pc) = Cs^{-\tau}$$

We find $n(s,pc)$ by eq. 2, for different system sizes; $L = 2^k$, for $k = 4,5,6,7,8,9$. The result is shown in figure 11.

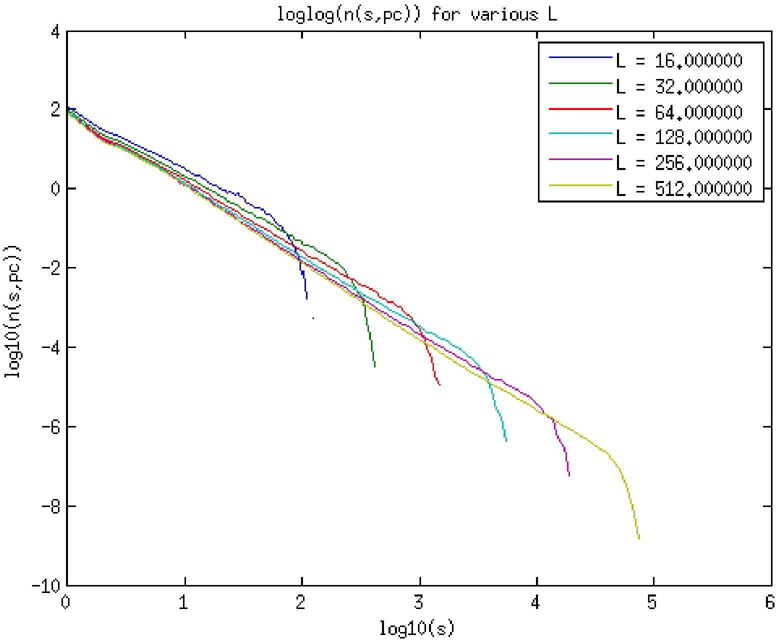


Figure 11: The cluster number density for various system sizes; L.

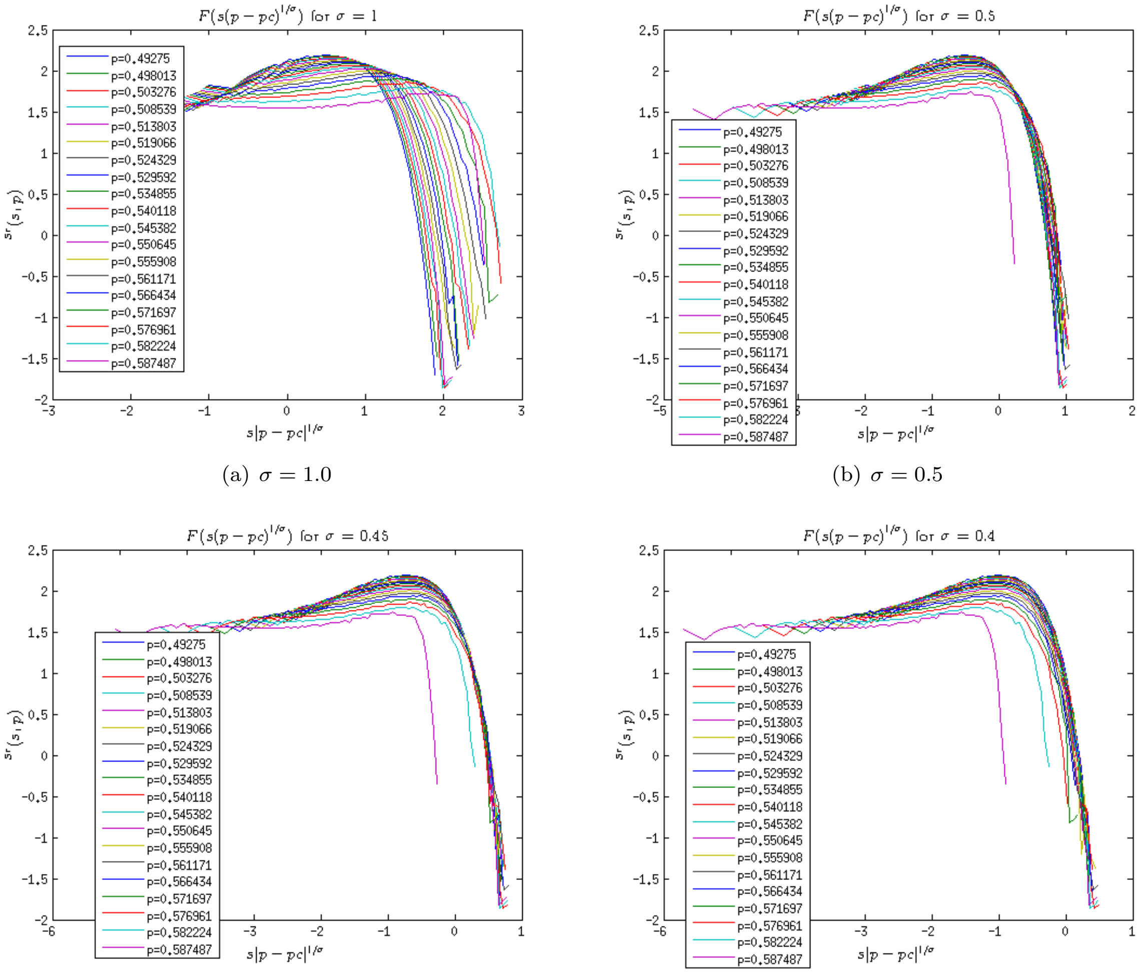
We expect the curves in figure 11 to be straight lines (it is a loglog plot of an expected power function), but again we see this behavior where they fall off as s increases towards L^2. To find the slope of the curves I simply only use the points before they fall off. I did this for all of the system sizes, and the average value I got for τ was $\tau = 1.87$ ($-\tau$ is the slope of the curves).
We expect the general cluster number density to have the form

$$n(s,p) = Cs^{-\tau}F(\frac{s}{s_\xi}) \tag{3}$$

where F is some function, and s_ξ is the scaling $s_\xi \sim |p-p_c|^{-\frac{1}{\sigma}}$. We want to give an estimate of the scaling, and use that

$$F(s/s_\xi) = \frac{n(s,p}{n(s,p_c)} \propto \frac{n(s,p)}{s^{-\tau}} = n(s,p)s^{\tau}$$

We insert $s_\xi \sim |p-p_c|^{-1/\sigma}$ into F: $F(s/s_\xi) \to F(s|p-p_c|^{1/\sigma}) = s^\tau n(s,p)$. If we plot F as a function of $s|p-p_c|^{1/\sigma}$, we should get the same shape for any value of p, assuming we have chosen the right σ and τ. We found $\tau \approx 1.87$ earlier, so we use this value and try different values of σ, for many values of p, and check if the curves fall on each other (same shape). Figure 12 shows shows my attempt to find σ. I tried for four different values; $\sigma = 1, 0.5, 0.45, 0.4$. We see that $\sigma = 1$ is far from right, but at $\sigma = 0.5$ we are getting closer. I found that $\sigma = 0.45$ gives the best value of the four. Two of curves with values for p close to p_c does not fall nicely on top of the others as they should.



(a) $\sigma = 1.0$

(b) $\sigma = 0.5$

(c) $\sigma = 0.45$

(d) $\sigma = 0.4$

Figure 12: Plots for estimation of the scaling $s_\xi \sim |p - p_c|^{-1/\sigma}$.

4 Mass scaling of percolating cluster

The next task is to find the mass $M(L)$ of the percolating cluster at $p = p_c$ as a function of L. The mass of the spanning cluster depends on the system size L, according to the power law $M \propto L^D$ (eq. 5.55 in book.pdf), which means our task is to find the exponent D. I make a loglog plot of the result, which can be seen in figure 13.

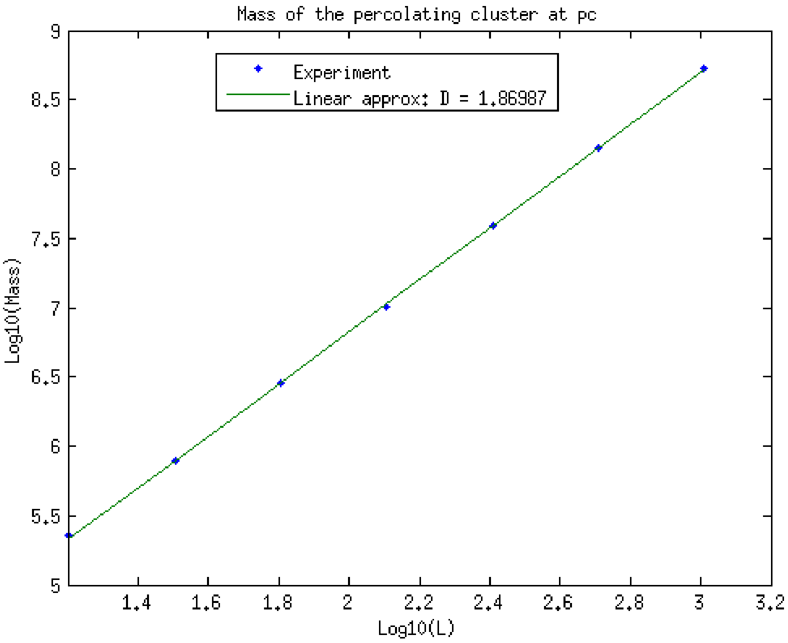


Figure 13: The mass of the percolating cluster as a function of L, in a loglog plot to find the exponent.

From this plot we see that the experimental points lies nicely along a straight line. The linear approximation has the slope $D = 1.87$.

5 Finite Size Scaling

We define $p_{\Pi=x}$ so that $\Pi(p_{\Pi=x}) = x$. This means the probability, p, that gives $\Pi = x$. Π is the probability for a spanning cluster to exist. $p_{\Pi=x}$ is a function of system size L. I found $p_{\Pi=x}$ for two value of $x = 0.3, 0.8$ as a function of L. Figure 14 shows us $p_{\Pi=x}$ for the two x-values.

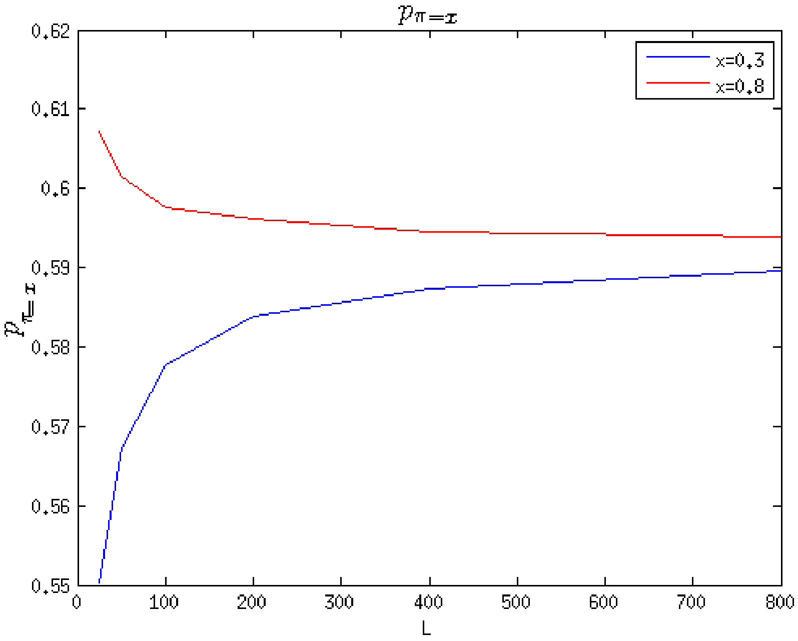


Figure 14: $p_{\Pi=x}$ as a function of system size,L, for two x-values: x = 0.3 and x = 0.8.

According to the scaling theory we have

$$p_{x_1} - p_{x_2} = (C_{x_1} - C_{x_2})L^{-1/\nu}.$$

We can use this and the result from the two x values to estimate the value ν. We do this by plotting $\log(p_{\Pi=0.8} - p_{\Pi=0.3})$ as a function of $\log(L)$, and hope to get a straight line. The slope of this line should then be $a = -1/\nu$. The plot can be found in figure 15.

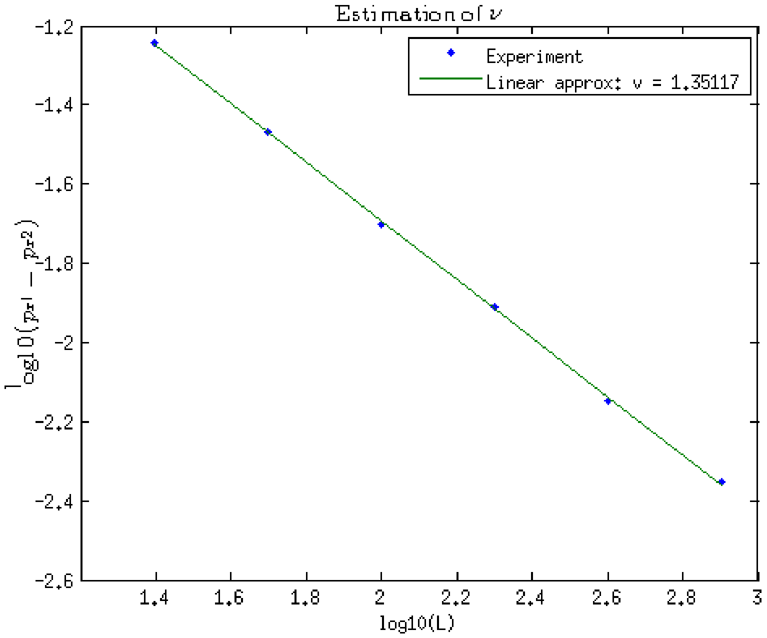


Figure 15: LogLog plot of $p_{\Pi=0.8} - p_{\Pi=0.3}$. Used to estimate ν.

The scaling theory also predicts that $p_{\Pi=x} = p_c + C_x L^{-1/\nu}$. If we plot $p_{\Pi=x}$ as a function of $L^{-1/\nu}$ we should get a linear function that crosses the y-axis at $y = pc$. We have two functions $p_{\Pi=0.8}$ and $p_{\Pi=0.3}$, and I plottet both of them in figure 16.

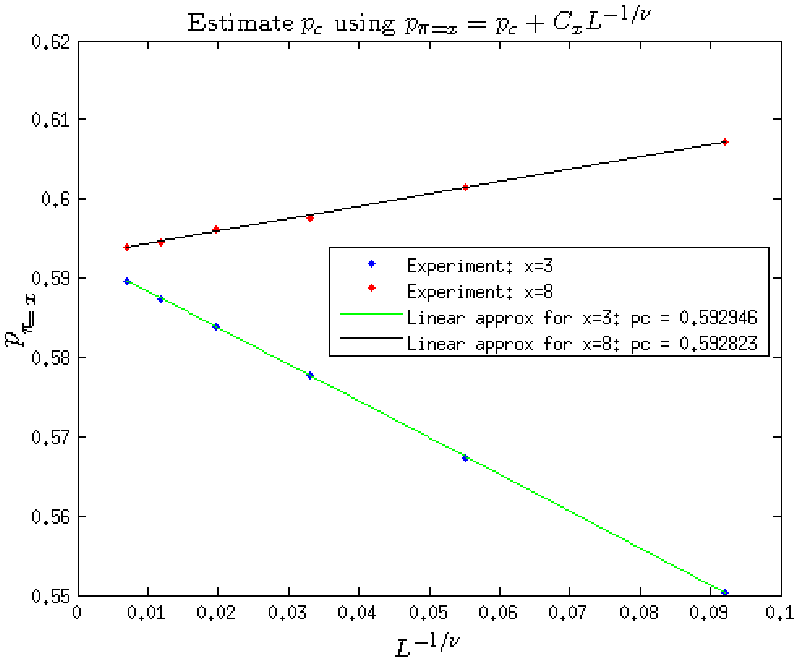


Figure 16: $p_{\Pi=x}$ as a function of $L^{-1/\nu}$.

The linear approximation gave us two values for pc:

$$x = 0.3 : pc = 0.592946$$
$$x = 0.8 : pc = 0.592823$$

From equation 6.27 in the notes we have

$$\Pi(p, L) = \Phi[(p - p_c)L^{1/\nu}]$$

If we plot Π as a function of $u = (p - p_c)L^{1/\nu}$ the curves should fall on top of each other if we have chosen the right parameters. In figure 17 we have done just that, and we see that the curves for different L-values have the same shape and lie very close to each other.

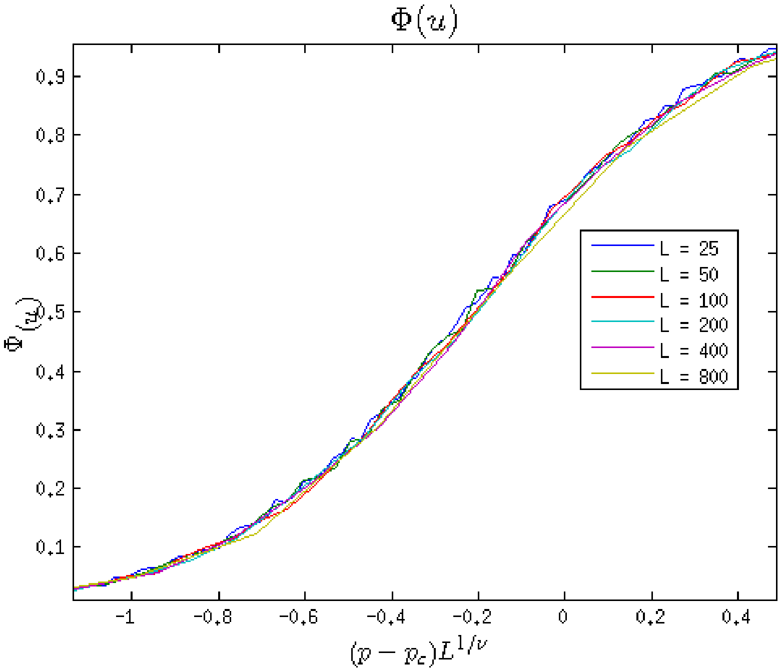


Figure 17: Data-collapse plot for $\Pi(p, L)$ to find $\Phi(u)$.

6 Singly connected bonds

We generate systems with spanning clusters, and let two walkers start on one side inside the spanning cluster. The walkers can only move inside the spanning cluster, and they follow different rules; one of them will always try to move left, and the other will always try to move right. If the walker can not move in the preferred direction, it tries to move straight ahead, and if that is impossible it has to take the opposite turn of the preferred turn. Sometimes it reaches a dead end, and then it just goes back where it came from. Somewhere in the cluster the two walkers' path may cross (they may at some point in time walk into a site where the other walker has been). These sites are called singly connected bonds. If we imagine a fluid flowing through the spanning cluster, these singly connected bonds are bottle necks where all of the fluid has to go through to get to the other side. All possible paths from one side to another will be on the inside of the two paths made from the walkers, så when the walkers pass the same site, it means all paths must go through this site. We were given a example program (exwalk.m), that visualizes the singly connected bonds. The result of this program is found in figure 18. The upper picture to the left shows us the spanning cluster in the system. The sites in the upper picture to the right are the singly connected bonds. In the lower picture to the left we see only the singly connected bonds. The lower picture at the right shows where the two walkers have been (the union of their paths).

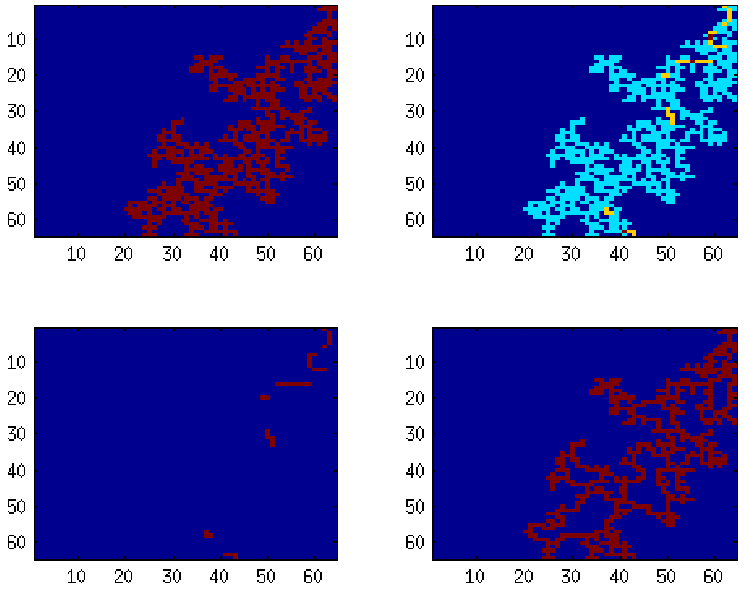


Figure 18: The result of the example program exwalk.m.

Now we want to find the mass of the singly connected bonds as a function of system size, L, for $p = p_c$. The mass is denoted M_{SC}. The mass follow the power law

$$M_{SC} \propto L^{D_{SC}},$$

and it is our job to find the exponent D_{SC}. We follow the usual procedure and make a loglog-plot of M_{SC} as a function of L. This is shown in figure 19. I made a linear approximation to the experiment points, and found that the slope was $D_{SC} = 0.76$. Figure 20 shows the actual experimental points together with the power law $M_{SC} \propto L^0.76$. The power law seems to follow the experimental points, but most of the points lie above the curve which means we have missed a little with D_{SC} (not by much though).

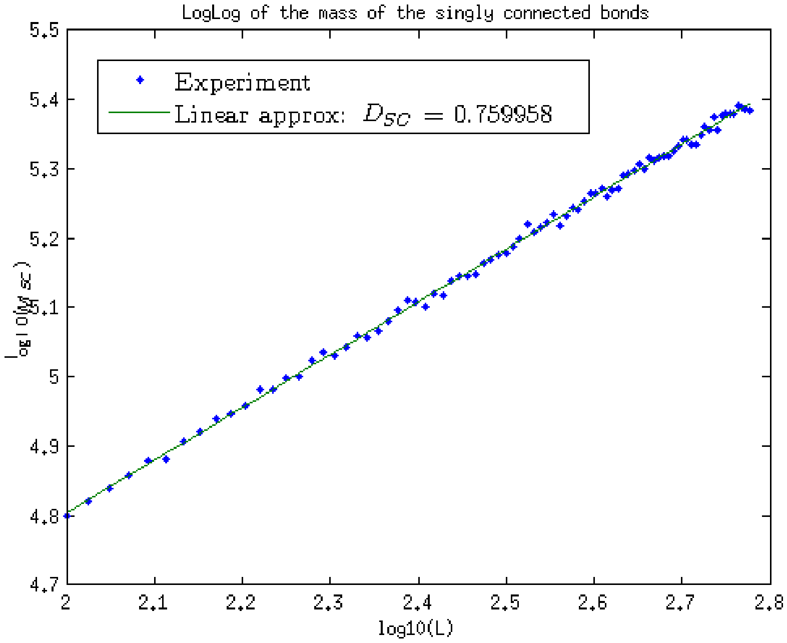


Figure 19: A loglog-plot of M_{SC} as function of L made to help us estimate the exponent D_{SC}.

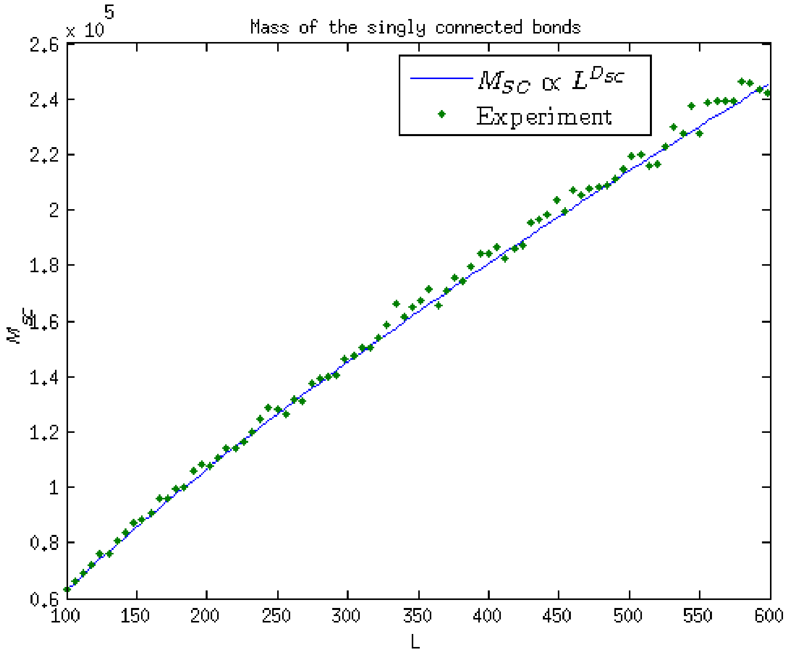


Figure 20: The experimental points together with the power law $M_{SC} \propto L^{D_{SC}}$.

We can also find the probability for a site being a singly connected bond: $P_{SC} = M_{SC}/L^d$. We find this probability for different values of $p - p_c$, see figure 21. We see that the probability goes down as system size increases, and as p increases, which seems natural to me.

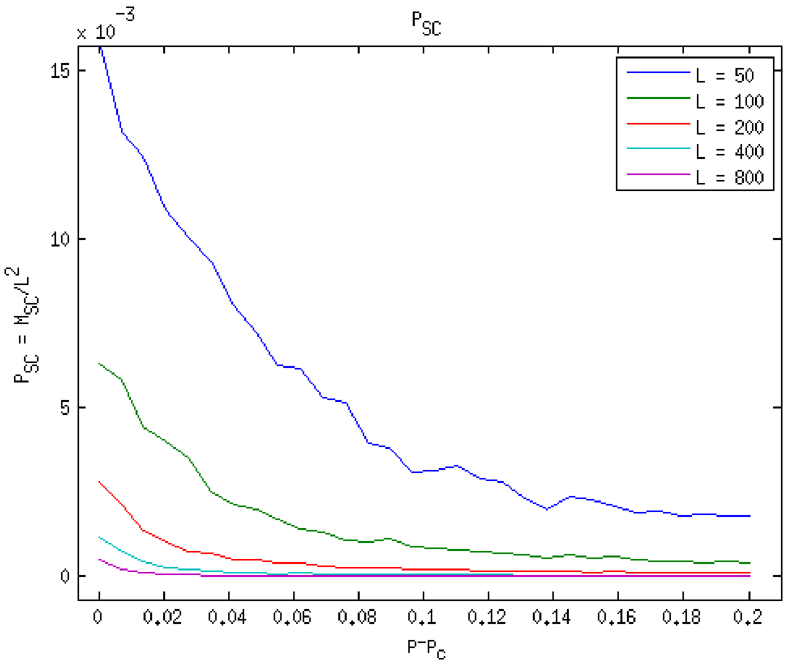


Figure 21: The probability for a site being a singly connected bond as a function of $p - p_c$.

7 Alternative 1: Flow on fractals

We will use and modify the program exflow.m to study flow on a spanning percolation cluster. This program sets up the linear equations to find the pressure at the sites, based on conservation of fluid. For every site we have that the flux going in to the site from above and from the right, must be equal to the flux moving out of the system to the site below and to the left. This gives us this equation for all the sites exept for the boundary:

$$\phi_{in}^{\downarrow} + \phi_{in}^{\rightarrow} = \phi_{out}^{\downarrow} + \phi_{out}^{\rightarrow} + rand \tag{4}$$

The flux from site i to site j (neighboring sites) is related to the local pressure drop through Darcy's law (eq. 11.3 in book.pdf):

$$\phi_{i,j} = g_{i,j}(p_i - p_j)$$

where $g_{i,j}$ is the conductance between the sites. In our problem the conductance between two sites is either 0 or 1; 1 if the two sites are in the spanning cluster, 0 if one of them are not. We convert eq. 4 into the notation with i indices. Assume our system is divided into sites numbered like this:

$$\mathbf{z} = \begin{bmatrix} 1 & 2 & 3 \\ 4 & 5 & 6 \\ 7 & 8 & 9 \\ 10 & 11 & 12 \end{bmatrix}. \tag{5}$$

The number of sites in x-direction is L. This translates the fluxes of a site i to

$$\begin{aligned} \phi_{in}^{\rightarrow} &= (P_{i-1} - P_i) g_{i-1,i} \\ \phi_{out}^{\rightarrow} &= (P_{i+1} - P_i) g_{i+1,i} \\ \phi_{in}^{\downarrow} &= (P_{i-L} - P_i) g_{i-L,i} \\ \phi_{out}^{\downarrow} &= (P_i - P_{i+L}) g_{i+L,i} \end{aligned}$$

and equation 4 becomes

$$(P_{i-L} - P_i) g_{i-L,i} + (P_{i-1} - P_i) g_{i-1,i} = (P_i - P_{i+L}) g_{i,i+L} + (P_{i+1} - P_i) g_{i+1,i}$$
$$P_i(-g_{i-L,i} - g_{i-1,i} - g_{i,i+L} + g_{i+1,i}) + P_{i-1}(g_{i-1,i}) + P_{i+1}(g_{i+1,i}) + P_{i-L}(g_{i-L,i}) + P_{i+L}(g_{i,i+L}) = 0$$

We see that we can write this as a sparse sort-of diagonal matrix multiplied with a vector $\mathbf{P}$, but to simplify the notation

$$\begin{aligned} A_{j,i} &= -g_{i-L,i} - g_{i-1,i} - g_{i,i+L} + g_{i+1,i} \\ A_{j,i-1} &= g_{i-1,i} \\ A_{j,i+1} &= g_{i+1,i} \\ A_{j,i-L} &= g_{i-L,i} \\ A_{j,i-L} &= g_{i+L,i} \end{aligned} \tag{6}$$

$$\mathbf{AP} = \begin{bmatrix} A_{0,0} & A_{0,1} & 0...0 & 0...0 & A_{0,L} & 0...0 & 0...0 \\ 0 & A_{1,0} & A_{1,1} & A_{1,2} & 0...0 & A_{1,1+L} & 0..0 \\ A_{j,i-L} & 0...0 & A_{j,i-1} & A_{j,i} & A_{j,i+1} & 0...0 & A_{j,i+L} \end{bmatrix} \begin{bmatrix} P_0 \\ P_1 \\ \dots \\ P_{i-L} \\ \dots \\ P_{i-1} \\ P_i \\ P_{i+1} \\ \dots \\ P_{i+L} \\ \dots \end{bmatrix} = 0 \tag{7}$$

We see that **A** is a sparse matrix with elements along the diagonal, and along four other 'lines' parallell to the diagonal.
The program solves this linear problem and finds the P-vector which tells us the pressure at each site. The next step is to find the absolute flux into each site, and to do that we use $\Phi_{ij} = \Delta P_{i,j}/g_{i,j}$ for every neighboring sites. I ran the example program, and it plottet the spanning cluster, see figure 22.

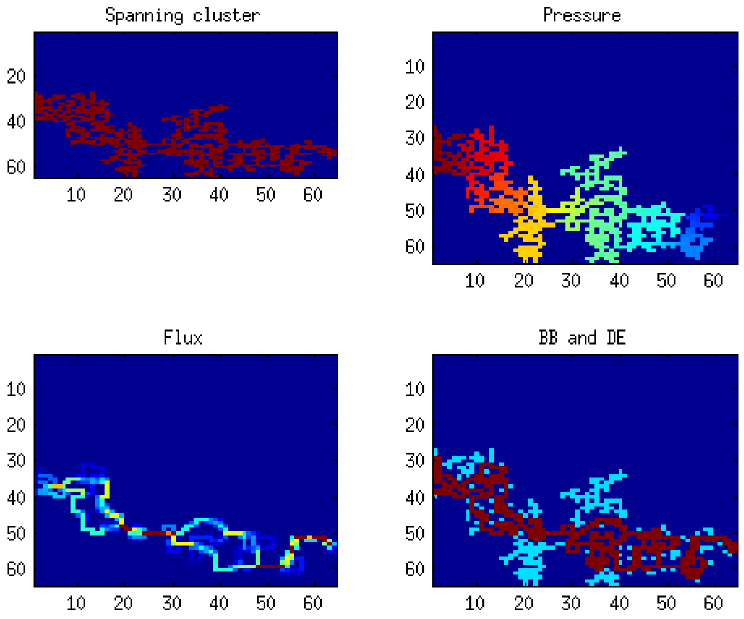


Figure 22: The spanning cluster from exflow.m.

To find the singly connected bonds we can first find how much flux runs through the whole system, and then find the sites which has that amount of flux (this will be the maximum value of the flux in the system). So from the flux part of 22 (bottom to the left), we can find how much flux is going through the system by summing up the flux of the first column of sites (or the last). To separate the backbone from the dangling ends we set a small limit and check if the flux in each site is below or above this limit. As in previously we find the dimensionalities: $M_{BB} \propto L^{D_{BB}}$, $M_{DE} \propto L^{D_{DE}}$ by measuring the masses and producing a loglog-plot. This is done in figure 23 and 24.

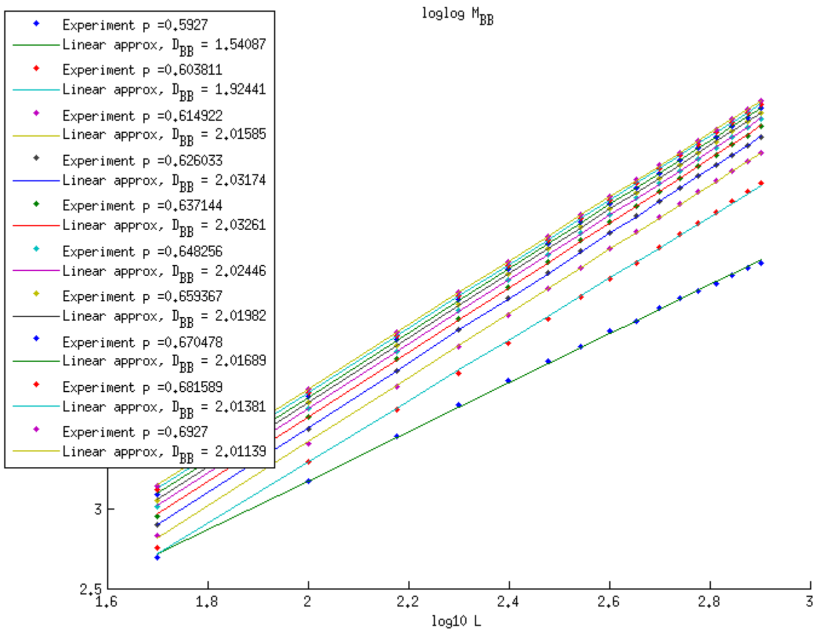


Figure 23: A loglog-plot of the mass of the backbone, to find the dimensionality D_{BB}.

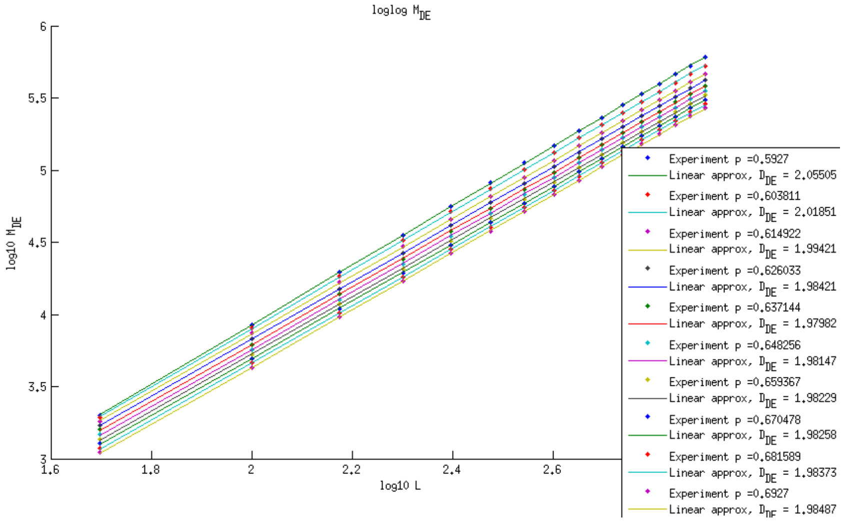


Figure 24: A loglog-plot of the mass of the dangling ends, in order to find D_{DE}.

In figure 23 we see that we get a bit different results for different values of p. The curve for p closest to p_c stands out, and looks a bit wrong to me. If I ignore this first result and find the average value of the rest I get $D_{BB} = 2.01$. If I include the first result I get $D_{BB} = 1.95$. For the dangling ends we get $D_{DE} = 1.99$.

7.1 Conductivity

In our case the conductivity is the same as the conductance because $G = L^{d-1}\sigma/L$ and we have a $d = 2$ system. If we have a pressure drop across the system we can find the total flux as $\Phi = G\Delta P$, and since $G = \sigma$ and in our case $\Delta P = 1$, we have $\Phi = \sigma$. This means we can find the conductivity by finding the total flux in the system. In figure 25 we see the conductivity plottet as a function of $p - P_c$ for two different system sizes.

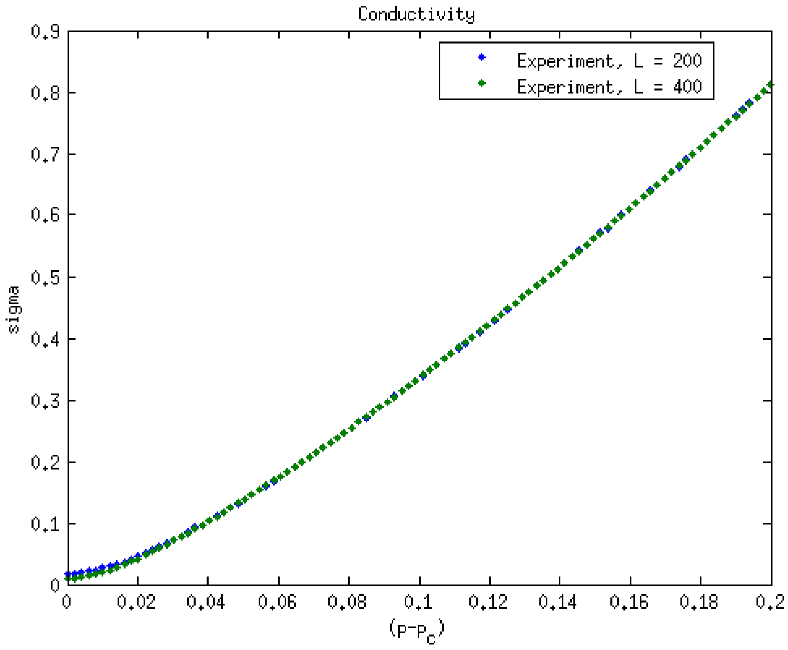


Figure 25: The conductivity as a function of $p - P_c$.

The two curves in figure 25 lies on top of each other, exept $p - p_c$ near 0. for p far away from p_c it looks like a straight line. Let's investigate the conductivity for $p = p_c$ as a function of system size; L.

From equation 11.7 in book.pdf we see that the conductance, and in our case the conductivity, can be described by the scaling exponent ζ_R:

$$\sigma = G(\infty, L) \propto L^{-\zeta_R}$$

where we have a spanning cluster at $p = p_c$. To find the exponent we make a loglog-plot and find an approximation to the slope. The loglog-plot is in figure 26, and from this we find $\zeta_R \approx 0.988$. Figure 27 shows the conductivity as a function of L together with the power law $\sigma \propto L^{-\zeta_R}$. We see that it is a good fit, but there is an overweight of points beneth the curve.

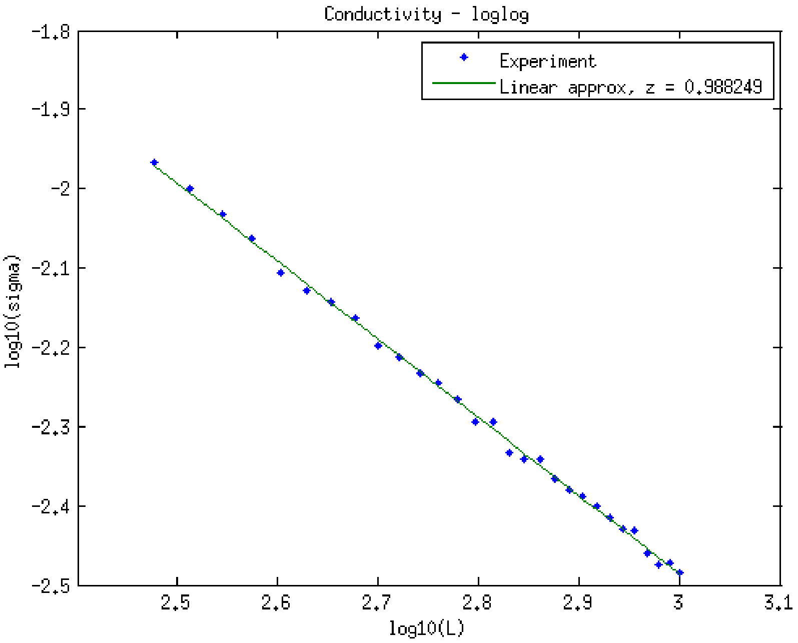


Figure 26: A loglog-plot of the conductivity as a function of system size L.

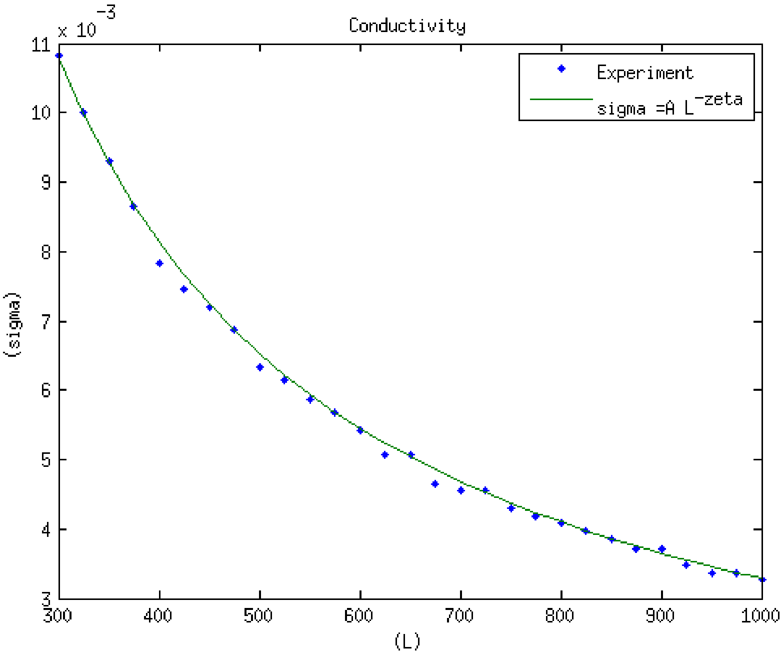


Figure 27: The conductivity as a function of L.