Perculation

Andreas V. Solbrå
a.v.solbra@fys.uio.no
University of Oslo
Department of Computational Physics

April 20, 2013

Abstract

The goal of this project is to get used to some of the concepts in percolation.

1 Generating percolation clusters

We begin by using Matlab to generate and visualize percolation clusters. We generate an $N \times N$ matrix of random uniform numbers then set each element equal to 1 if it is larger than some chosen ϵ, and 0 if it is smaller. The we can then color the different connecting 1s in different colors, in order to study the formation of clusters. Some examples of this is shown in figure **??**.

For infinitely large grids, there is a critical quantity called the percolation threshold, p_c, so that if the probabilty p of a site beeing occupied is larger than p_c, there is a so-called *spanning* or *percolation cluster* which goes infinitely far in any direction. An interesting quantity is then the probability that a site is a part of such a spanning cluster. We call this $P(p)$. We cannot measure this on a computer, we can, however, measure its finite counterpart, $P(p, L)$. $P(p)$ has the form $P(p) = (p - p_c)^\beta$, where in this case $p_c = 0.59275$. We can assume that $P(p, L)$ takes the same form, and try to measure β for the finite versions. The results of some such atempts are shown in figure **??**. We see that there is there are some differences between the finite and infinite case. For instance, in the infinite case $P(p)$ is zeros below p_c. In the finite case there is always a non-zero probability that a connecting cluster exists, so the curve is spread out a bit. We also see that the value for β we measure is dependent on the system size. The value for β for different values of L is shown in figure **??**. The limit of β for $L \to \infty$ has been shown to be around 0.18??.

2 Determining the exponent of power-law distributions

We generate the following set of data points in Matlab:

```
z = rand(1e6,1).^(-3+1);
```

What is the probability distribution for this set of points? A straight forward approach is to simply make a normalized histogram of the data set. This will be a good approximation for a big data set. However, since a lot of the large values are very unlikely, we can make a better histogram if we use a logarithmic bin size. This will give a somewhat better view of the statistics.

A more sophisticated approach would be to find the cumulative distribution, $P(Z > z)$, and the find the probability density as

$$f_Z(z) = \frac{dP(Z > z)}{dz} \tag{1}$$

We can find $F_Z(z)$ by using a numerical derivative. However we risk running into some problems using this method if a bin does not contain any new points. We assume that $f(u) \propto u^\alpha$, and we want to measure α. If we try to do linear regression on the logarithms, we end up with $log(0)$ for one or more values. Another, more stable approach is to note that with $f(u) \propto u^\alpha$, then $F(u) = 1 - u^{\alpha+1}$. We can then measure $\alpha + 1$ from a log-log plot of $1 - F(u) = z^{\alpha+1}$. The results of these different approaches are shown in figre **??**, and we can measure $\alpha \approx 3/2$. We should note that using the latter method will allow us to use a much finer grid spacing, which is needed to get close to the correct value.

3 Cluster number density $n(s, p)$

The cluster number density is the probability for a random cluster to be of a particular size. It is espescially interesting to investigate this for p close to p_c. Figure **??** shows $n(s, p)$ for p above and below p_c.

We can also estimate $n(s, p_c, L)$ for different values of L, as shown in figure **??**. From theoretical considerations, we have $n(s, p_c) = Cs^{-\tau}$, where in the infinite case $\tau = 187/91 \approx 2.05$. We can measure τ for our finite cases. We strip off the end points that are heavily effected by the finite size of the system. and measure the increase of the loglog-plot. For $L = 512$, we find $\tau = 1.89$.

4 Mass scaling of percolating cluster

eps4-eps-converted-to.pdf

(a) $\epsilon = 0.4$

eps5-eps-converted-to.pdf

(b) $\epsilon = 0.5$

eps6-eps-converted-to.pdf

(c) $\epsilon = 0.6$

eps7-eps-converted-to.pdf

(d) $\epsilon = 0.7$

Figure 1: The plots show cluster formation for different values of ϵ.

betaFit100-eps-converted-to.pdf

(a) $L = 100$

betaFit1000-eps-converted-to.pdf

(b) $L = 1000$

Figure 2: The plots show the best fit of the data to the theoretical curve $P(p) = (p - p_c)^{\beta}$

pdf-eps-converted-to.pdf

(a) Probability density

loglog_pdf-eps-converted-to.pdf

(b) log-log of probability density

cumul-eps-converted-to.pdf

(c) log-log of probability density

loglog_cumul-eps-converted-to.pdf

(d) log-log of probability density

Figure 3: The plots show the best fit of the data to the theoretical curve $P(p) = (p - p_c)^{\beta}$

pabove-eps-converted-to.pdf

(a) $p > p_c$

pbelow-eps-converted-to.pdf

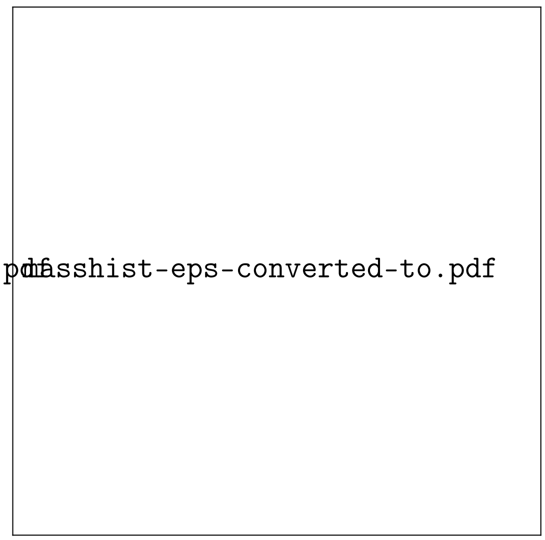
(b) $p < p_c$

Figure 4: The plots show $n(s, p, L = 400)$ for various p.

differentLs-eps-converted-to.pdf

Figure 5: The plots show $n(s, p, L = 400)$ for various p.

(a) a typical mass distribution



(b) loglog of average means

Figure 6: The figure shows the average mass of the percolating clusters for various system sizes, as well a a typical mass distribution