# FYS-4460
# Project 3

Gullik Vetvik Killie

May 8, 2015

## Contents



## Q (a)

Using these features, you should make a program to calculate P(p, L) for various p. Hint: you can use either BoundingBox or intersect and union to find the spanning cluster. How robust is your algorithm to changes in boundary conditions? Could you do a rectangular grid where Lx >> Ly? Could you do a more complicated set of boundaries? Can you think of a simple method to ensure that you can calculate P for any boundary geometry?
**Answer**

```
%Program for the third project in the subject Fys-4460
%Inordered systems and Percolation at UiO
% 01.04.2015    Gullik Vetvik Killie

clear all
close all

Lx = 100;                  %The size of the grid
Ly = 100;
prob = (0.4:0.01:1.0);      %The probabilities we want to check
nProb = length(prob);       %# probabilites we check through
nPercolated = zeros(nProb,1);
areaPercolationClusters = zeros(nProb,1);
N = 100;


for i = 1:N

    randomArray = rand(Lx,Ly);
    %Cycle trough all the probabilites and calculates clusters for each
```

```
    %probability
    for iProb = 1:nProb

        filledNodes = randomArray < prob(iProb);     %Check the a random array

        %The nodes are given numbers according to which cluster they belong
        %to
        [sortedNodes , nClusters] = bwlabel(filledNodes, 4);

%         %Create a plot of the clusters,, only enable if you want to plot
%         img = label2rgb(sortedNodes,'jet','k','shuffle');
%         image(img);
%
%         return


        clusterProperties = regionprops(sortedNodes ,'BoundingBox', 'Area');
        boundingBox = cat(1,clusterProperties.BoundingBox);
        area = cat(1, clusterProperties.Area);

        %Finds an open channel in x or y direction respectively
        jx = find(boundingBox(:,3)==Lx);
        jy = find(boundingBox(:,4)==Lx);

        %Picks up the area that percolated through either side
        j = union(jx,jy);


        if length(j)> 0 %Percolation found
            nPercolated(iProb) = nPercolated(iProb) + 1;
            for jj = 1:length(j)
                areaPercolationClusters(iProb) = areaPercolationClusters(iProb) + area(j(jj));
            end
        end
    end
end
%Pi is the chance for a connecting cluster through the whole area, while P
%is the fraction of the nodes that are a part of a percolating cluster
subplot(2,1,1)
Pi = nPercolated/N;
plot(prob,Pi), xlabel('p'),ylabel('\Pi')

subplot(2,1,2)
P = areaPercolationClusters/(N*Lx*Lx);
plot(prob,P); xlabel('p'),ylabel('P')

%Task b: Estimate beta. P is proportional to beta above p_c
```

## Q (b)

We know that when $p > p_c$, the probability $P(p, L)$ for a given site to belong to the percolation cluster, has the form $P(p, L) \propto |p - p_c|^\beta$

Use your program to find an expression for $\beta$. For this you may need that pc $= 0.59275$.
**Answer** Since $P$ is proportional to $|p - p_c|^\beta$ a logarithmic regression can be used.

$$P(p, L) \propto |p - p_c|^\beta \tag{1}$$
$$\log\left(P(p, L)\right) \propto \beta \log\left(p - p_c\right) \tag{2}$$