

FYS4460
SPRING 2016

Project 4

John-Anders Stende

Date: June 8, 2016

1 Cluster number density

Cluster number density $n(s, p)$ er definert som sannsynligheten for at en site er en spesifikk site i en cluster av størrelse / masse s . $n(s, p)$ er altså cluster size-distribusjonen: Vi kan lage et histogram av cluster-størrelser, dvs. hvor mange clusters som er av størrelse s for flere s . $n(s, p)$ kan måles slik:

$$\overline{n(s, p)} = \frac{N_s}{L^d} \quad (1)$$

der overlinen betyr at dette er en målt, ikke eksakt størrelse, N_s er antall clusters av størrelse s , L er størrelsen til system og d er dimensjonen. Formen til $n(s, p)$ vil være konstant / lineær til vi kommer til et visst punkt, der den faller raskt. Vi kan dermed definere en cut-off cluster-størrelse, eller en såkalt karakteristisk cluster-størrelse:

$$s_\xi \propto |p - p_c|^{-1/\sigma} \quad (2)$$

som viser hvordan denne størrelsen divergerer når $p \rightarrow p_c$. σ er en univiersell eksponent, dvs. at den ikke avhenger av detaljene til latticen, men den avhenger av dimensjonaliteten. Det er en såkalt kritisk eksponent, som beskriver et fysisk system nære en faseovergang. I perkolasjonsteori er denne faseovergangen at systemet går fra å ikke perkolere til å perkolere, dvs. når $p \rightarrow p_c$. Systemet kan alltid beskrives ved en potensfunksjon med en kritisk eksponent rundt en slik faseovergang. Ved slike overganger kan alle perkolasjonsvariablene redefineres vha. dimensjonsløse størrelser, dvs. ratioer av potenser av disse dimensjonsløse størrelsene. Disse kalles skaleringsfunksjoner. $n(s, p)$ kan bli uttrykt generelt på følgende måte:

$$n(s, p) = s^{-\tau} F(s/s_\xi) \quad (3)$$

der $\tau = 2$ i to dimensjoner. $F(s/s_\xi)$ er en skaleringsfunksjon, her er s reskalert som s/s_ξ , og vi er kvitt avhengigheten av p , altså vil alle kurvene for forskjellige p falle på hverandre, dette kalles datakollaps. Det avhenger imidlertid av at de kritiske eksponentene er riktige. Dette kan også brukes til å finne de kritiske eksponentene ved å plote $n(s, p)$ helt til alle kurvene faller oppå hverandre. Poenget med skaleringsfunksjonene er at de er uavhengige av systemstørrelse L og sannsynligheten p , men sterkt avhengige av alle andre parametre. Altså vil data for forskjellige L og p kollapse på samme kurve dersom parametrene er valgt riktig.

1.1 Måle cluster number density

$n(s, p_c, L)$ kan måles ved å realiserer et system M ganger, måle N_s for hver realisering og deretter regne ut

$$\overline{n(s, p)} = \frac{N_s(M)}{L^2 \cdot M} \quad (4)$$

der $N_s(M)$ er antall clusters av størrelse s målt i M realiseringer av perkolasjonssystemet. Vi vil altså måle for forskjellige L ved perkolasjonsterskelen p_c . Først initialiserer vi

```
M = 100;           % number of realizations / measurements
L = [25 50 100 200 400 800];
pc = 0.59275;      % percolation threshold
allarea = [];      % areas of all clusters sampled M times
```

Så looper vi gjennom alle L , M ganger:

```
for j = 1:length(L)
    for i = 1:M
        z = rand(L(j), L(j));
        m = z < pc;
```

Her lager vi en random matrise av størrelse $L \times L$, og finner så alle occupied sites, dvs. alle sites som har et tall mindre enn p_c . Clusterne identifiseres slik:

```
% lw: each occupied site is given an index that says
% which cluster it belongs to
[lw, num] = bwlabel(m, 4);
```

der 4 betyr nærmeste-nabo-konvensjon. Deretter finner vi evt. spanning clusters og fjerner dem: HVORFOR GJØR VI DETTE???

```

% find and remove spanning cluster
top = lw(1,:);
bottom = lw(L(j),:);
left = lw(:,1);
right = lw(:,L(j));
tb = intersect(top,bottom);
lf = intersect(left,right);
sc = union(tb,lf);
% remove sites that are not part of clusters
% sc now contains the indicies of eventual spanning clusters
sc = sc(sc~=0);
if ~isempty(sc);
    for k = 1:length(sc);
        % remove spanning clusters
        %length(sc)
        lw = lw(lw ~= sc(k));
    end
end
end

```

Deretter finner vi antall sites i hver cluster, dvs. N_s :

```

% find distribution of cluster sizes
s = regionprops(lw, 'Area');
% area contains number of sites in all clusters
area = cat(1, s.Area);
allarea = cat(1, allarea, area);

```

For små s har vi god statistikk; mange clusters er på noen få sites. For store s er det imidlertid veldig få, et histogram vil derfor kun ha et datapunkt i mange bins for store s . Vi vil derfor ha større binstørrelse for større s . Dette kan oppnås ved logaritmisk binning, dvs at binkantene er a^i , $i = 0, 1, 2, \dots$. Da er det viktig å huske på å dele antallet clusters i hver bin med binstørrelse!

```

% logarithmic binning
% n: vector of number of elements in each bin
% s: location of all bin centers
[n,s] = hist(allarea, L(j)^2);
a = 1.2; % basis for logarithmic bins
logamax = ceil(log(max(s)) / log(a)); % edge of largest bin
bins = a.^(0:1:logamax); % bin edges
nl = histc(allarea, bins); % histogram with new bins
ds = diff(bins); % bin sizes
sl = (bins(1:end-1) + bins(2:end))*0.5; % bin centers
nsl = nl(1:end-1)'./ (M*L(j)^2*ds); % cluster number density
legendInfo{j} = sprintf('p = %.2f', p(j));
loglog(sl, nsl);
hold on;

```

For alle L gjør vi et loglog-plot. Når vi ikke gjør datakollaps ser det slik ut:

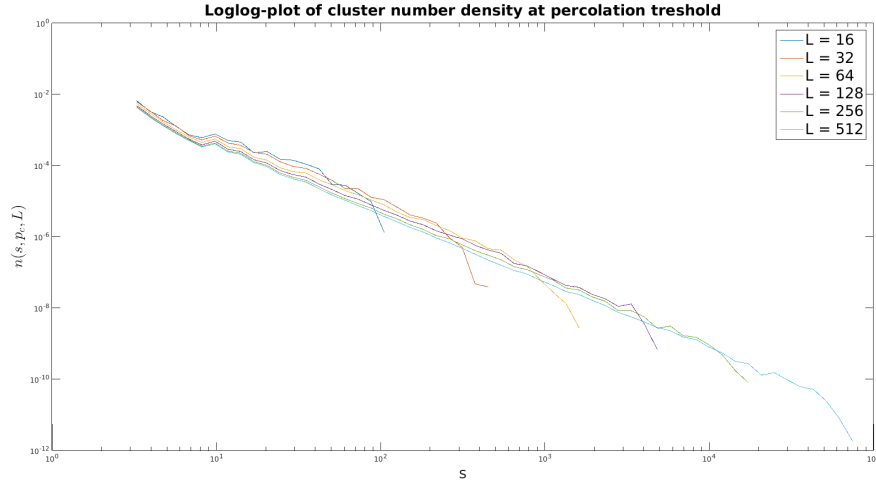


Figure 1: Cluster number density for p_c som funksjon av L .

Vi ser at formen til $n(s, p_c, L)$ nettopp er som i (3) : For lave s går funksjonen som $s^{-\tau}$, før den faller eksponentielt ved det kritiske punktet. Her er det skaleringsfunksjonen F som bestemmer oppførselen.

2 Finite size skalering

Vi må utvikle en finite size-skaleringsteori for å finne skaleringsfunksjonen. Her tar vi hensyn til at systemet er av endelig størrelse, dvs. L er ikke uendelig stor. Vi har altså at

$$n(s, p) = s^{-\tau} F(s/s_\xi) \quad (5)$$

Vi vet også at det det er kun én eneste karakterisk størrelse av en cluster:

$$s_\xi \propto \xi^D \quad (6)$$

der ξ er korrelasjonslengden, dvs. cut-off til korrelasjonsfunksjonen $g(r)$ som sier hva sannsynligheten for at to okkuperte sites i avstand r er i samme cluster. Dermed har vi

$$n(s, p) = s^{-\tau} F(s/\xi^D) \quad (7)$$

Vi har to ulike paradigmer når for endelige systemstørrelser L :

$$s^\tau n(s, p) = \begin{cases} F(s/L^D) & L \ll \xi \\ F(s/\xi^D) & L \gg \xi \end{cases} \quad (8)$$

dvs. når $L \ll \xi$ vil vi typisk se en spanning cluster, fordi den typiske størrelsen av en cluster er større en systemstørrelsen. Vi kan imidlertid ikke vite om vi er ved $p = p_c$ eller ikke, bare at p_c er et eller anet sted for $\xi(p) > L$. I dette tilfellet vil derfor den typiske cluster-størrelsen være lik systemstørrelsen:

$$\xi \propto L \quad (9)$$

Vi skal måle $n(s, p_c, L)$, vi er altså på $p = p_c$. Vi vet at ξ divergere som en potensfunksjon når $p \rightarrow p_c$:

$$\xi \propto |p - p_c|^{-\nu} \quad (10)$$

derfor har vi $\xi = \infty$ nå. Altså er vi i området $L \ll \xi$, derfor er skaleringsfunksjonen til $n(s, p)$ gitt som

$$s^\tau n(s, p, L) = F(s/L^D) \quad (11)$$

Det betyr at dersom vi måler $n(s, p_c, L)$ som ovenfor og plotter $\log_{10}[s^\tau n(s, p_c, L)]$ mot $\log_{10}[sL^{-D}]$, vil vi få en datakollaps. Dette vil demonstrere finite size-skaleringsteorien. Vi brukte de gitte verdiene $\tau = 187/91$ og $D = 91/48$ (disse kan også måles ved simuleringer).

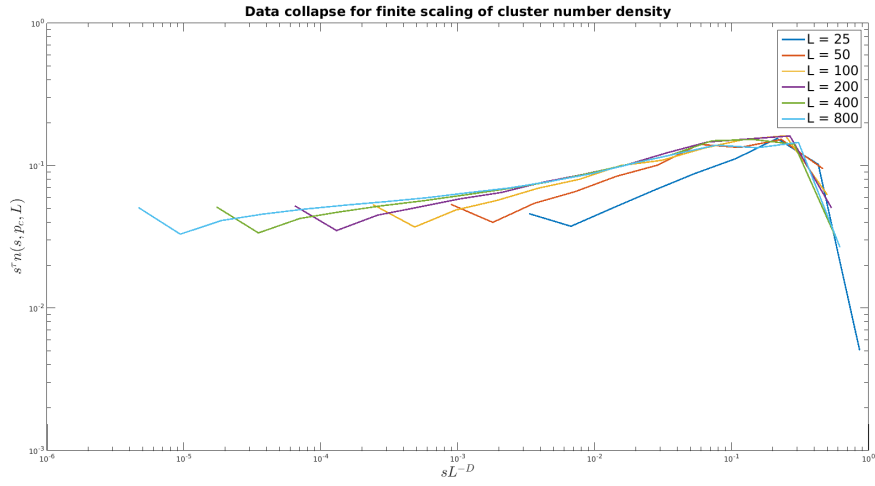


Figure 2: Cluster number density for p_c som funksjon av L .

Hvis man bruker andre verdier for de kritiske eksponentene enn de ovenfor, ser vi at det ikke blir datakollaps:

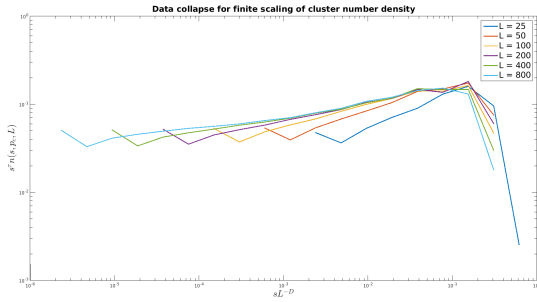


Figure 3: $D = 2.0$, $\tau = 187/91$

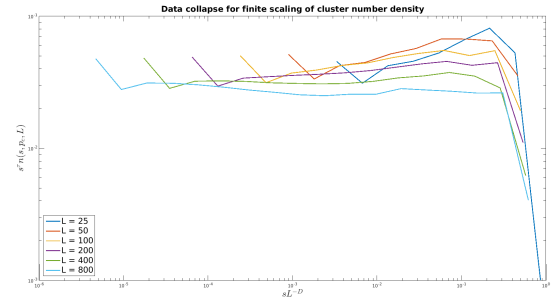


Figure 4: $D = 91/48$, $\tau = 1.9$

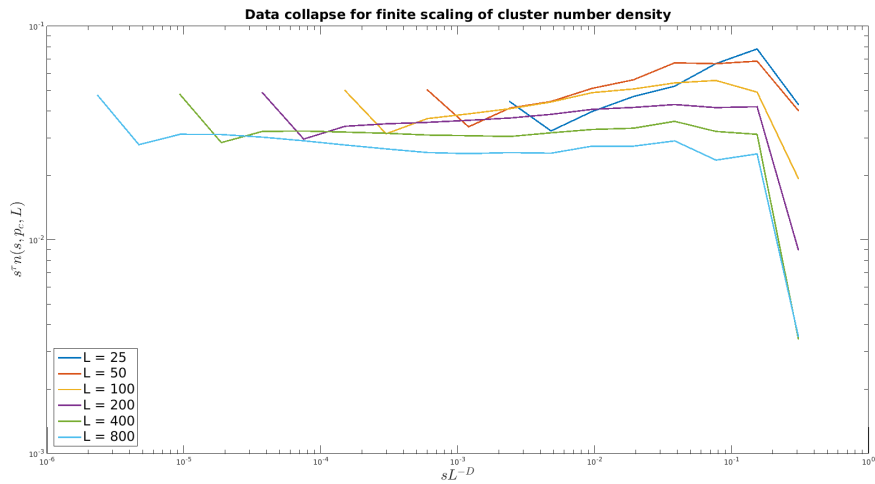


Figure 5: $D = 2.0$, $\tau = 1.9$

JEG MÅ SI NOE OM HVORDAN DETTE ENDRER SEG MED DIMENSJONER. D AVHENGER VEL AV DIMENSJONEN?