

Exam preparations FYS4460

Fredrik E Pettersen
f.e.pettersen@fys.uio.no

June 7, 2013

Contents

1	Molecular-dynamics algorithms	3
2	Molecular-dynamics in the micro-canonical ensemble	5
3	Molecular-dynamics in the micro-canonical ensemble	5
4	Measuring the diffusion constant in molecular-dynamics simulations	6
5	Measuring the radial distribution function in molecular-dynamics simulations	6
6	Thermostats in molecular-dynamics simulations	6
7	Generating a nano-porous material	7
8	Diffusion in a nano-porous material	7
9	Flow in a nano-porous material	7
10	Algorithms for percolation systems	8
11	Percolation on small lattices	9
12	Cluster number density in 1-d percolation	10
13	Correlation length in 1-d percolation	11
14	Cluster size in 1-d percolation	11
15	Measurment and behavior of $P(p, L)$ and $\Pi(p, L)$	12
16	The cluster number density	13
17	Finite size scaling of $\Pi(p, L)$	14
18	Subsets of the spanning cluster	15
19	Random walks	15

1 Molecular-dynamics algorithms

There are several different possibilities for “variation” when doing molecular dynamics. First of all one can use different potentials according to what one would like to model. The simplest being the Lennard-Jones potential, which only includes two-particle interactions. The Lennard-Jones potential is typically pretty good for simulations of noble gases. If we should want to study another material we will therefore need another potential. The Weber-Stillinger potential includes both two- and three-particle interactions, and can model silicon (Si). Silicon will, in equilibrium, form 4-coordinated tetrahedral structures, which the Weber-Stillinger potential reproduces. The VKRE (Vashista, Kalia, Rino, and Ebbsjö) potential is another potential which includes both two and three particle interactions, but this potential is made (specifically) to simulate SiO_2 . The two-body part includes three terms, the Coulomb interaction, steric repulsion due to ionic sizes, and a charge-dipole interaction resulting from the large electronic polarizability of O_2^- . Yet another example of a possible potential is the reaxFF potential which also includes four-particle interactions. This potential lets us study water molecules in an SiO_2 matrix.

In our Lennard-Jones studies we used the Verlet algorithm for time-integration. It is a simple, yet quite good integrator which conserves energy very well. It can be viewed as an advanced Euler-Cromer method, where we first update the velocity of each particle at half the timestep, then the position is integrated one full timestep using the new velocity. When this is completed for all particles we calculate two-body forces using the new positions before we use the new forces to fully integrate the velocities.

All of the potentials mentioned above decrease in strength with the distance between atoms. For the Lennard-Jones potential, this is illustrated in figure 1. As we see from this figure, there is for all practical purposes no interaction between particles that are a certain length apart. This means that we can skip the calculations of forces (which is very expensive) between these particles. As a practical measure we will therefore divide the system we are simulating into cells which are the same size as the cutoff length and only calculate forces between particles in neighbouring cells. In this way we are sure that no important calculations are left out, and still gain a significant speedup. In fact, should we do all pairwise force calculations we will at best need something like $\frac{n^2}{2}$ FLOPS, whereas we will only need something like n FLOPS when we use cells.

We initialize the system by placing it in a face centered cubic lattice (figure 2) and give each atom a random velocity drawn from the uniform distribution. We will of course also need to remove any initial drift in the system also by rescaling all velocities by the mean of all velocities. Finally, to get around the boundary problems, where any potential we have chosen will strictly speaking be incorrect, we implement periodic boundary conditions. That is, two particles on each side of the lattice are neighbouring.

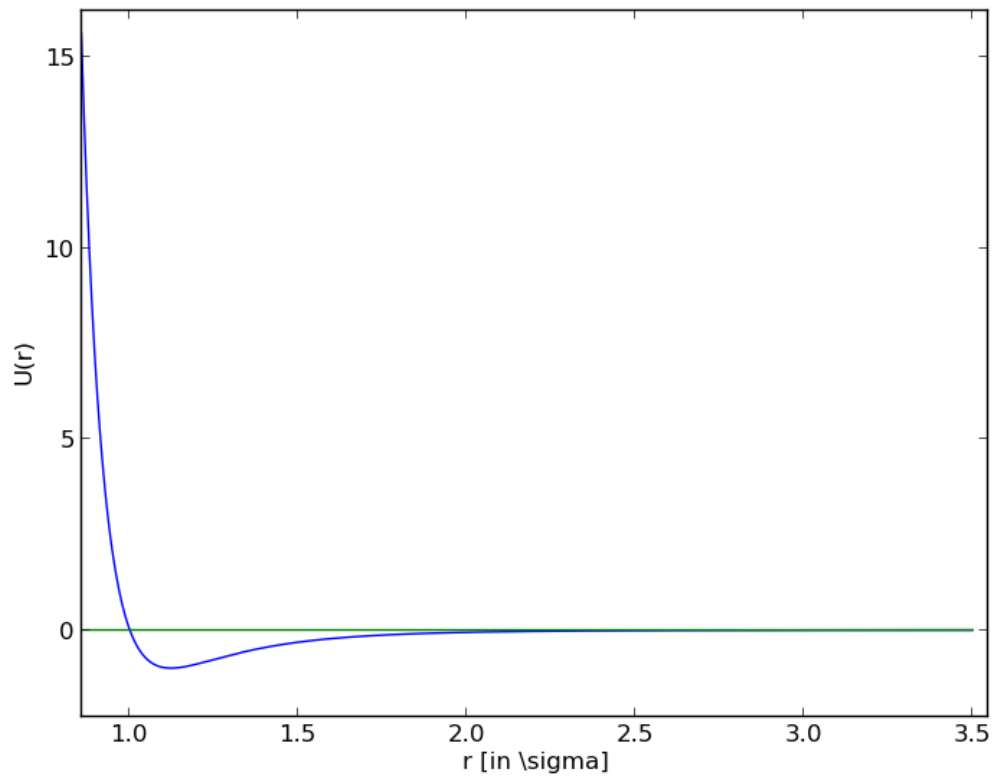


Figure 1: Strength of Lennard-Jones potential as a function of distance between atoms

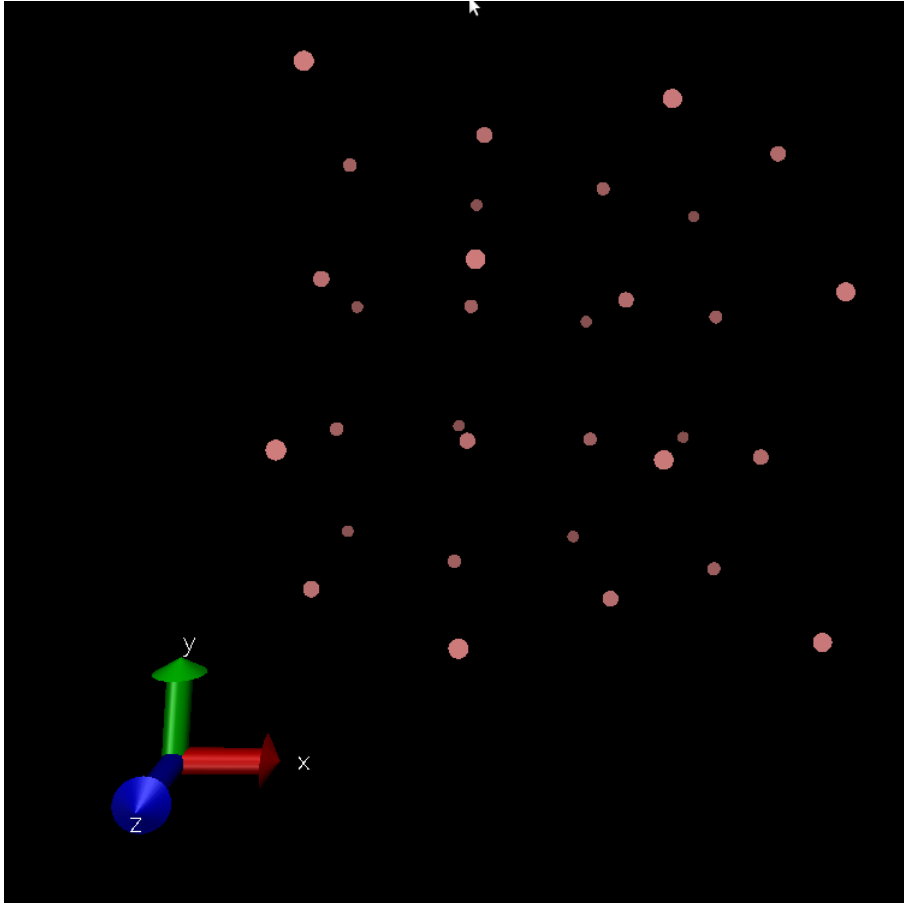


Figure 2: Face centered cubic lattice

2 Molecular-dynamics in the micro-canonical ensemble

If we leave the simulation to itself after setting it up, the temperature will drop quite dramatically and then fluctuate around some steady state solution. This is first off all unwanted, and (possibly) unphysical. One way to solve this problem is by introducing a thermostat. In these simulations we are in the microcanonical ensemble. That is, The number of particles, volume, and energy are all fixed. In the microcanonical ensemble it can be shown that the temperature is

$$T = \frac{2E_k}{3N} \quad (1)$$

where the kinetic energy is the sum of the kinetic energy off all the particles. This means that we can manipulate the temperature of the system by rescaling the velocity of each particle. There are, as usual, a few alternatives on how to do this. The berendsen thermostat lets us simulate the canonical ensemble (poorly) by “surrounding” our system with a heat bath, and rescaling the velocities accordingly. The scaling constant is shown in eq 2.

$$\gamma = \sqrt{1 + \frac{\Delta t}{\tau} \left(\frac{T_{bath}}{T} - 1 \right)} \quad (2)$$

Another possibility is the Andersen thermostat which simulates collisions between atoms inside the system and in the heat bath. Atoms which collide will gain a new, normally distributed velocity with standard deviation $\sqrt{k_B T_{bath}/m}$. For all atoms a random, uniformly distributed number is drawn. If the number is less than $\frac{\Delta t}{\tau}$, the atom is assigned a new velocity. A downside of the Andersen thermostat is that it disturbs the dynamics of lattice vibrations.

3 Molecular-dynamics in the micro-canonical ensemble

The ultimate goal of molecular dynamics simulations is to measure macroscopic quantities such as the pressure, permeability, viscosity etc. from the microscopical simulations. This can be achieved via statistical mechanics. As mentioned in section 2 the temperature can be expressed from the microcanonical ensemble as equation 1

(Insert mathematical argument).

In a similar way we can find the pressure of the entire system as it is expressed in equation 3.

$$p = \vec{F} \cdot \vec{r} \quad (3)$$

These quantities can for example be used to say something about how a system will react to a fluid being driven through it.

4 Measuring the diffusion constant in molecular-dynamics simulations

We can measure the diffusion constant by measuring the mean square displacement of the particles in the system. This will relate to the diffusion constant as

$$\langle r^2 \rangle \simeq 2dtD \quad (4)$$

where d denotes the dimensionality of the system (in most cases 3), t is directly relatable to the number of timesteps that have passed and D is the diffusion constant. One problem by using this relation is that, to my knowledge, it has only been derived for particles in a homogenous fluid which is much larger than the average size of the particles. Also, one must account for the periodic boundary conditions. This is done by summing the distance traveled by each particle at each timestep, and carefully accounting for “jumping” from one side of the system to the other.

Equation 4 is also widely used to measure the diffusion constant in random walks computations. For example we can use it to measure the diffusion constant of random walkers on a spanning cluster in percolation theory. INSERT SOME MORE STUFF

5 Measuring the radial distribution function in molecular-dynamics simulations

The radial distribution function can be measured in by counting the number of particles in spherical shell with increasing radius.

To calculate the pair distribution function from a simulation, the neighbors around each atom or molecule are sorted into distance bins. The number of neighbors in each bin is averaged over the entire simulation. For example, a count is made of the number of neighbors between 2.5 and 2.75, 2.75 and 3.0 Å and so on for every atom or molecule in the simulation. This count can be performed during the simulation itself or by analyzing the configurations that are generated.

The radial distribution function tells us something about the density distribution in a material, and can be used to calculate several macroscopic quantities such as the pressure, structure factor, compressibility, etc. Difficulties in measuring $g(r)$ include it being computationally intensive, troubles with getting the scaling right (I still have not gotten it right), and similar problems.

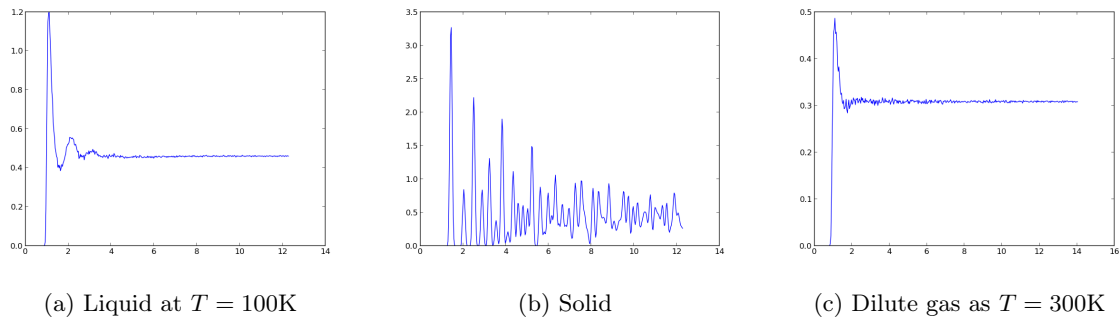


Figure 3: Radial distribution function for liquid, solid and gas. x-axis shows distance in σ

6 Thermostats in molecular-dynamics simulations

As mentioned in section 2 the system will be represented by the microcanonical ensemble if we leave it to itself. We can adjust the system so that it is represented by the canonical ensemble by “surrounding” it with a heat

bath. This is done by introducing a thermostat which (for example) rescales the velocities of the particles so that the temperature of the system matches that of the heat bath. The best thermostat for simulations of the canonical ensemble is the Nosé-Hoover thermostat, however this is (apparently) rather difficult to implement. In section 2 two other alternatives are described, the Andersen and the Berendsen thermostats.

7 Generating a nano-porous material

Generally we generate a nanoporous matrix by first thermalizing a “normal” system at some temperature. Next, we need some algorithm to make it porous. A typical way of doing this is by removing some of the particles in the system by some rule, or at least mark them as stationary. The pores can be specified by, for example, drawing spheres of a random radius placed at a random position. Atoms outside the sphere are marked as stationary (or the other way around). In more complex simulations one usually heats the system a lot, expands it and rapidly cools it again. This will give a more realistic geometry.

The structure of a nanoporous matrix can be characterised by its porosity (which we have a somewhat poor measure for), but this raises the problem of having several possible permeabilities for each porosity. One could have all the pores in one large sphere (no permeability), or in one large cylindrical pore which has a large permeability. The fluid is typically characterized by its viscosity and its density. This also poses somewhat of a problem in that the viscosity is not independent of the geometry of the matrix or (by extension) its permeability. It is also “strongly” dependent on density. In short the viscosity is a hard measurement.

8 Diffusion in a nano-porous material

The diffusion constant is measured in the same way as we described in section 4.

I guess one could expect the diffusion constant to depend on the porosity of the system. It would also make sense if the diffusion constant was different in different directions because the connectivity might depend on direction. This should (at least for randomly generated matrices) cancel out on larger scales.

Diffusion in larger scale porous media is governed by nonlinear diffusion equations which are immensely complicated to model (using upwind differences etc). However, the thing all diffusion processes have in common is reaching a steady state after some time. This is also expected from the low density fluid in the nanoporous system.

9 Flow in a nano-porous material

We can introduce flow in one direction in the nanoporous material by adding a driving force to all particles, much like a gravitational force. This should make all particles drift in one direction.

The model can be checked by simulating the flow through a cylindrical pore. From continuum mechanics we know that the velocity profile for such a system should be a parabola, and we expect this to be the case for our MD simulations as well. Figure 4 shows this if you really put your imagination into it.

To calculate the viscosity of the model we need the superficial velocity distribution of the fluid. One way to estimate the fluid viscosity is by simulating flow through a cylindrical pore of a given radius and with a given driving force we find the viscosity from

$$u(r) = \frac{\Delta P}{4\mu L} (a^2 - r^2)$$

$$\implies \mu = \frac{nFa^2}{4u(0)}$$

The permeability of the nanoporous material can be measured by counting the flux of particles through one part of the system (say one square area). The easiest part to pick is the periodic boundary. We have from Darcy’s law that

$$\frac{Q}{V} = \frac{k}{\mu} \left(\frac{\Delta P}{L} - \rho g \right) = \frac{kN}{\mu V} F \iff k = \frac{\mu Q}{NF} \quad (5)$$

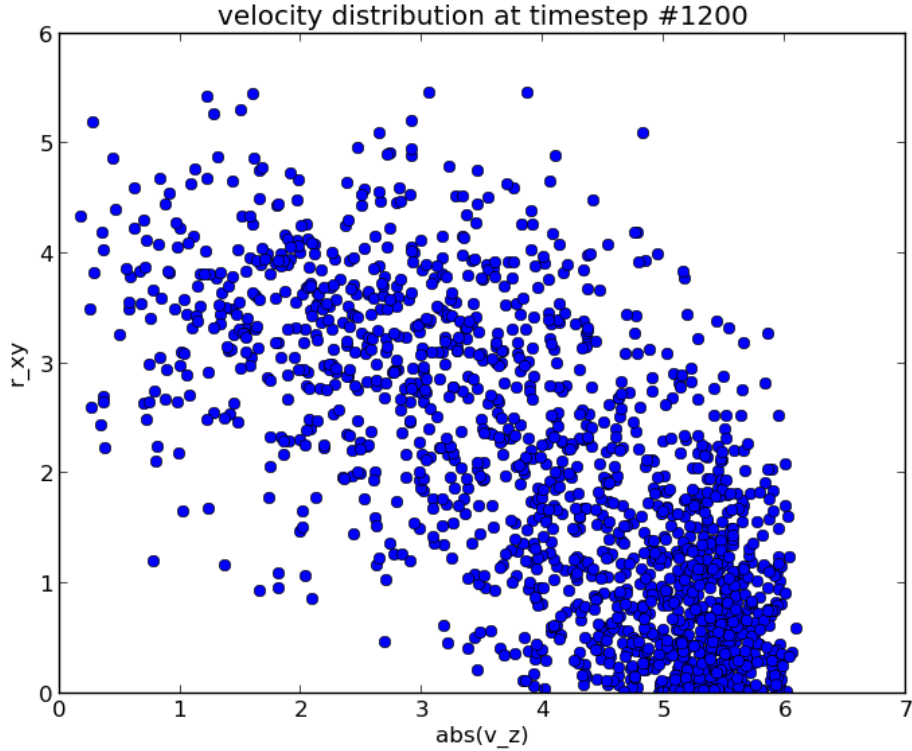


Figure 4: v_z for all moving particles projected onto a radius centered in the middle of the cylinder.

10 Algorithms for percolation systems

To generate percolation systems for simulation we need a (square) matrix of random numbers. Commonly these are drawn from the uniform distribution. Next we will need to set a porosity for the system, and mark all matrix entries smaller than the porosity as filled sites, and all entries larger than the porosity as unfilled sites. This gives us clusters of filled sites which have neighbours that are filled as well (or clusters of connected, filled sites), see figure 5. These clusters are labelled with a number. We then find the spanning cluster (should it exist) by checking if the same label exists at both sides of the matrix (and remove the label 0). The percolation probability, which is the probability of there being a spanning/percolating cluster, is measured by checking whether or not there is a percolating cluster, and averaging the measurements over several matrices.

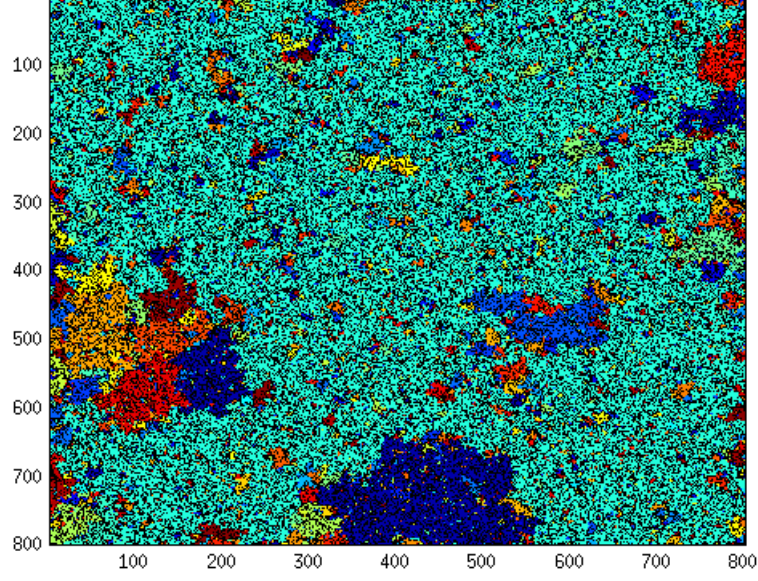


Figure 5: Percolation matrix of size 800×800 with porosity $p = 0.6$

11 Percolation on small lattices

If we want to study percolation on very small lattices, say 2×2 lattices, we can find exact solutions for $\Pi(p, L)$. As we found in project 3, $\Pi(p, L)$ goes to a step function as L goes to infinity. Similarly, we would expect $\Pi(p, L)$ to go to a polynomial of degree 1 as L goes to 1. This can be shown mathematically. The porosity of a 1×1 system is either 0 or 1, in which case the percolation probability is either 0 or 1. This means that the percolation probability has to be a straight line from $(0, 0)$ to $(1, 1)$ since we cannot well define any “path” the function should take. For the 2×2 case the system can be in either of the states shown in figure 6.

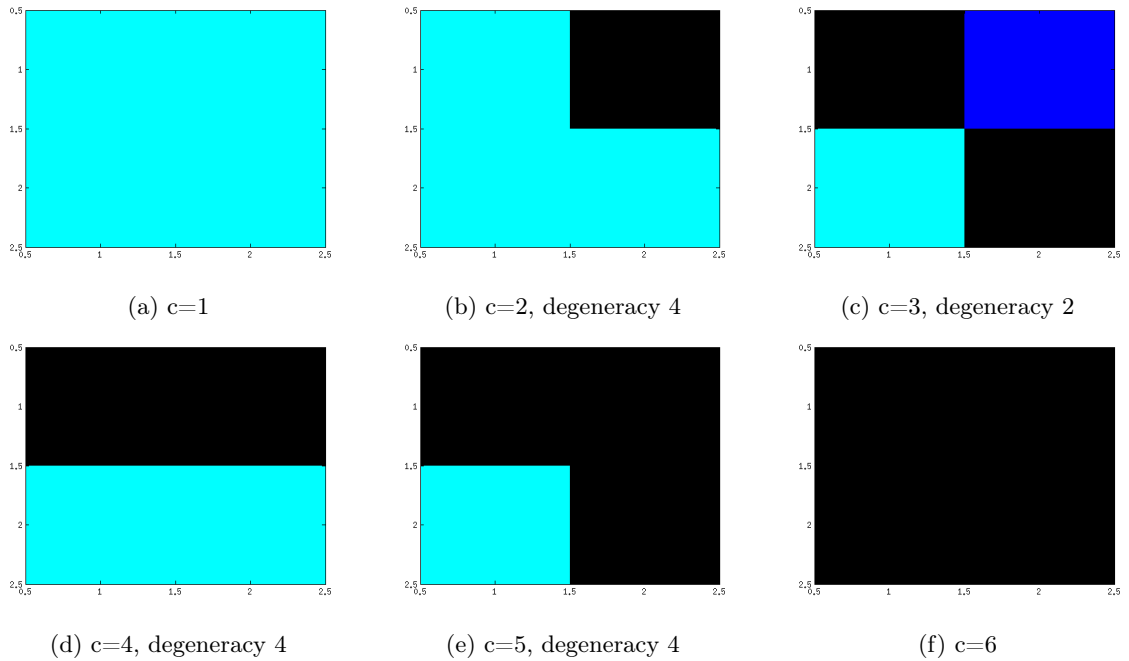


Figure 6: Possible states and their degeneracy for a 2×2 system

From figure 6 we can find Π directly

$$\begin{aligned}\Pi &= 0 \cdot 1 \cdot p^0(1-p)^4 + 0 \cdot 4 \cdot p^1(1-p)^3 + 0 \cdot 2 \cdot p^2(1-p)^2 + \\ &1 \cdot 4 \cdot p^2(1-p)^2 + 1 \cdot 4 \cdot p^3(1-p)^1 + 1 \cdot 1 \cdot p^4(1-p)^0 \\ \Pi &= 4 \cdot p^2(1-p)^2 + 4 \cdot p^3(1-p) + p^4\end{aligned}$$

12 Cluster number density in 1-d percolation

The cluster number density $n(s, p)$ is defined as the probability for a site to be a specific site (say the leftmost) in a cluster of size s . The probability of this is quite simply defined from equation 6; there are s sites filled, with a probability p , and one unfilled site on each site, with a probability $(1-p)$. The probability for this site to be any part of a cluster of size s is $s \cdot n(s, p)$.

$$n(s, p) = (1-p)p^s(1-p) = p^s(1-p)^2 \quad (6)$$

If we write the number of clusters of size s as N_s and the number of filled sites $\mathcal{N} = pL$, this should logically be equal to the sum over all clustersizes times the number of clusters of that size $pL = \sum_{s=1}^{\infty} sN_s$. We can now estimate the probability for a given, filled site to belong to a cluster of size s as

$$(sn(s, p))' = \frac{sN_s}{\mathcal{N}}$$

Now, in order to estimate whether a given site (filled or not) to be a part of a cluster of size s , all we need to do is multiply by p , giving us $sn(s, p) = \frac{psN_s}{pL} \implies n(s, p) = \frac{N_s}{L^d}$, where d is the dimensionality of our system. Rewriting equation 6 gives us $n(s, p) = (1-p)^2 e^{s \ln(p)} = (1-p)^2 e^{-s/s_\xi}$ where $s_\xi = \frac{-1}{\ln(p)} \simeq (p_c - p)^{-1/\sigma}$. As we can see, s_ξ is going to diverge as $p \rightarrow p_c$. This will make $n(s, p)$ a decaying exponential function with decreasing slope.

The expression $n(s, p) = \frac{N_s}{L^d}$ is the same as we use to measure the cluster number density in 2-D. The resulting curves are also similar (see f.eks figure 7).

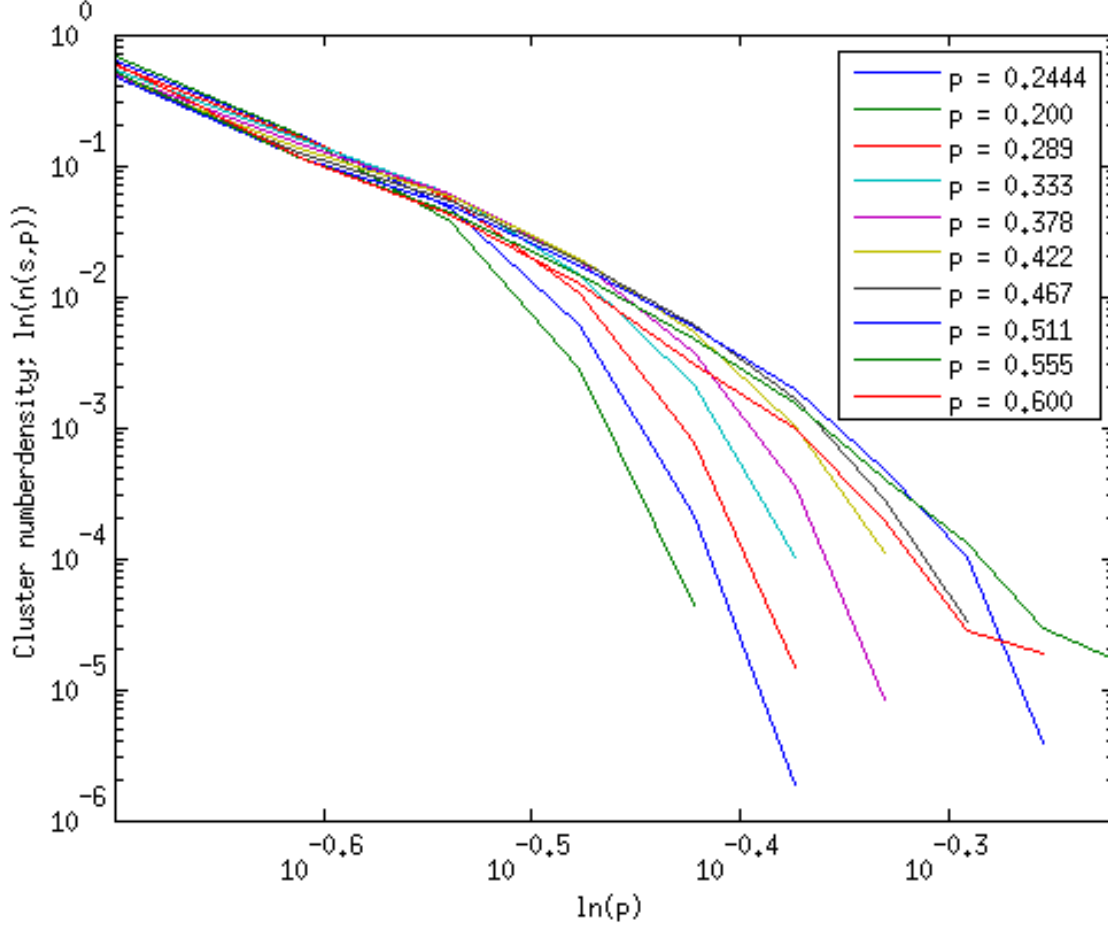


Figure 7: Cluster number density measured for $p < p_c$.

13 Correlation length in 1-d percolation

The correlation function $g(r)$ describes the probability for two sites separated by a distance r to belong to the same cluster, given that both sites are filled. In 1-D, the only way for this to happen is if all the sites between the two sites are filled as well. This means that we get

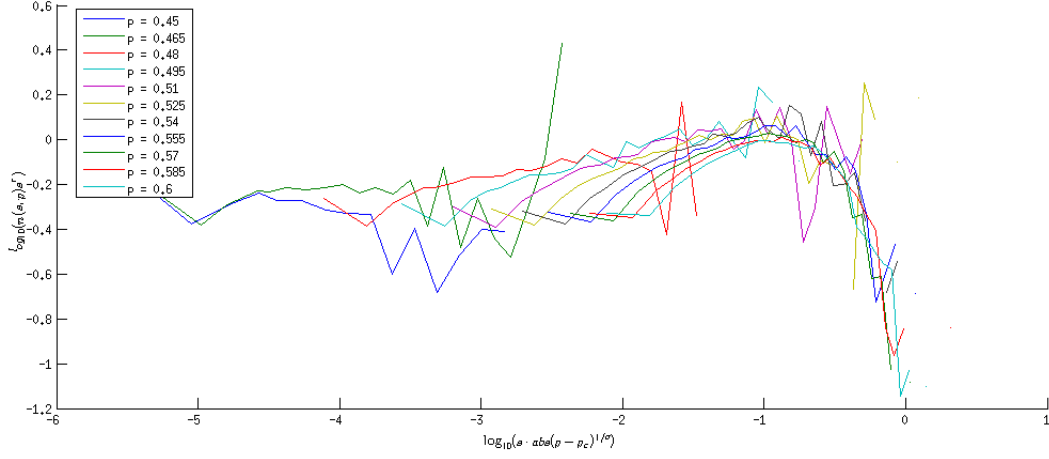
$$g(r) = p^r = e^{r \ln(p)} = e^{\frac{-r}{\xi}} \quad (7)$$

Where we, again, set $\xi = \frac{-1}{\ln(p)}$. ξ diverges as $p \rightarrow 1 = p_c$. Utilizing this, we can find the way it diverges by expanding $\ln(p) \rightarrow_{p \rightarrow p_c} -(p_c - p)$. This gives $\xi = \xi_0(p_c - p)^{-\nu}$ with $\nu = 1$

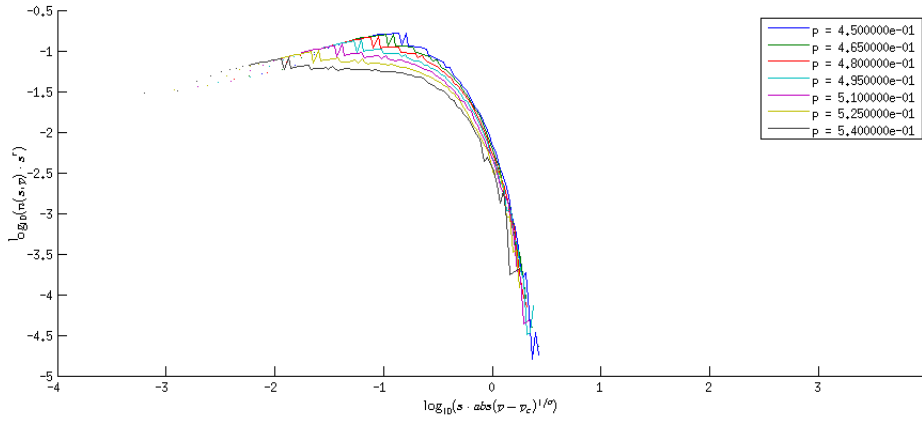
14 Cluster size in 1-d percolation

As mentioned in section 12, the characteristic cluster size, s_ξ , is defined from the cluster number density where it enters as $n(s, p) = (1 - p)^2 e^{-s/s_\xi}$ and $s_\xi = (p_c - p)^{-1/\sigma}$ where $\sigma = 1$ in 1-d. From this it is obvious that s_ξ will diverge as $p \rightarrow p_c$. In two dimensions, this is true when $p \rightarrow p_c$ both from above and below.

In the 2-d simulations we can (not without trouble) measure σ either by plotting $n(s, p)$ as a function of s and measure where $n(s, p)$ is halved, or we can cheat by making a datacollapse plot (rescaling the axes by s_ξ) for various values of σ until we find a decent value. This will give us s_ξ since we have all values entering in its expression. See figure 8.



(a) Data collapse plot estimating $\sigma \simeq 0.4$.



(b) Data collapse plot estimating $\sigma \simeq 0.4$ using a more correct algorithm.

Figure 8: Data collapse plots

15 Mesurment and behavior of $P(p, L)$ and $\Pi(p, L)$

The percolation probability is the simplest quantity to describe the behavior of of finite matrices. As we have allready mentioned, $\Pi(p, L) \rightarrow_{L \rightarrow \infty} H(p - p_c)$, and will be more like a hyperbolic tangent for small L . This is measured in the trivial way of averaging over several matrices (see figure 9a). The probability for a site to belong to the percolating cluster $P(p, L)$ will of course be zero as long as there is no percolating cluster, and will then increase very rapidly for a little while before (more or less) linearly going to 1 as the porosity of the system goes to 1. $P(p, L)$ will behave (naturally) in much the same way as $\Pi(p, L)$ does for finite system sizes. This quantity is also measured in the trivial way of counting and averaging.

We can make a similar finite size scaling theory for $P(p, L)$ from the relation $P(p, L) \propto \frac{M(p, L)}{L^d}$. We also have for $M(p, L)$:

$$M(p, L) \propto \begin{cases} \xi^{D-d} L^d & , L \gg \xi \\ L^D & , L \ll \xi \end{cases}$$

Inserting for $\xi \propto (p - p_c)^{-\nu}$ gives us

$$P(p, L) \propto \begin{cases} (p - p_c)^{-\nu(D-d)} L^d & , L \gg \xi \\ L^{D-d} & , L \ll \xi \end{cases}$$

Which means that we can easily determine $\beta = -\nu(D - d) = \frac{-4}{3}(1.89 - 2) = 0.146666\dots$

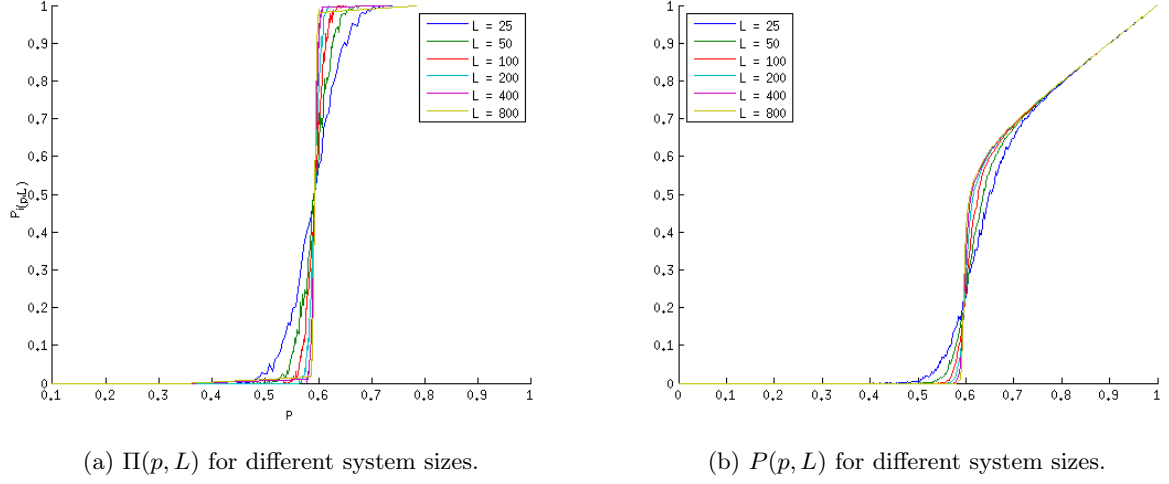


Figure 9: blergh

16 The cluster number density

As we discussed in section 12 the cluster number density is the probability for a given site to be a specific site in a cluster of size s . In 1-d this is given by equation 6, but this will (of course) break down in more than 1-d because it will be more difficult to define the boundary, and because there are several ways of having a cluster of size s . To meet this challenge, we can instead set $n(s, p) = \sum_t g_{s,t} p^s (1-p)^t$, where $g_{s,t}$ is the degeneracy of the cluster of size s , and t is all the possible numbers of neighbours.

The cluster number density is still measured in much the same way as in 1-d, with some more details. We will now have to do logarithmic binning of the cluster size, s , because most of the cluster sizes will be rather small, but a few will be very large. So we divide the “ s -axis” into increasingly larger bins, and count the number of clusters within each bin.

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

Dividing by the bin size gives us our estimate for $n(s, p)$.

We can estimate σ by producing a data collapse plot of $n(s, p)$ as shown in figure 8.

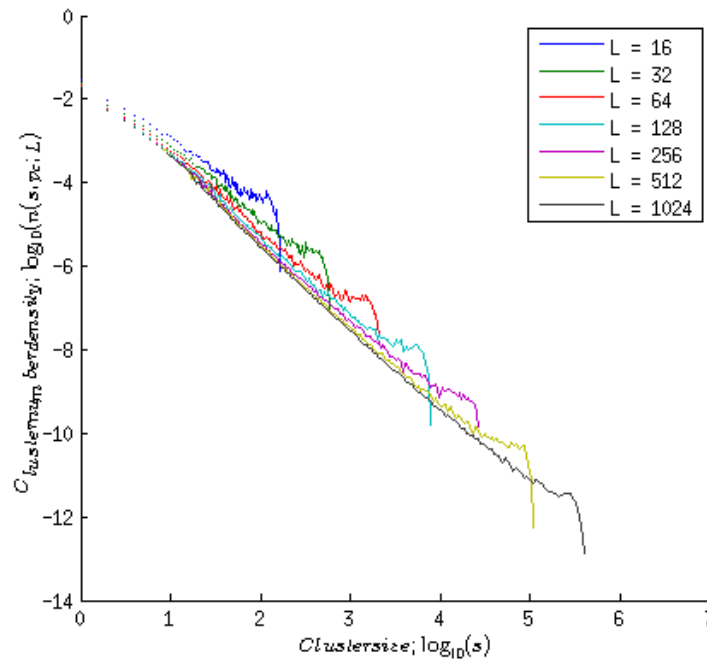


Figure 10: The cluster number density behaves as shown with increasing system size.

17 Finite size scaling of $\Pi(p, L)$

As we discussed in section 15, the percolation probability will behave in a predictable manner for finite system sizes. This is a characteristic we can make good use of to measure important exponents with high precision, and also measure the percolation threshold.

$$\begin{aligned}
 \Pi(p, L) &= \xi^0 f\left(\frac{L}{\xi}\right) = f\left(\frac{L}{\xi}\right) \\
 &= f\left(\frac{L}{\xi_0(p-p_c)^{-\nu}}\right) \\
 &= f\left(\frac{L(p-p_c)^\nu}{\xi_0}\right) \\
 &= \tilde{f}\left(\left[L^{1/\nu}(p-p_c)\right]^\nu\right) \\
 &= \Phi\left[(p-p_c)L^{1/\nu}\right]
 \end{aligned}$$

Where we have used that $\xi = \xi_0(p-p_c)^{-\nu}$ when $p > p_c$, $\tilde{f}(u) = f\left(\frac{u}{\xi_0}\right)$ and $\Phi(u) = \tilde{f}(u^\nu)$. This scaling ansatz can be used as follows. We start by fixing p which gives $\Pi(p) = x$. This can be denoted as

$$p_{\Pi=x}(L) \tag{8}$$

The scaling ansatz now gives us

$$x = \Phi\left[(p_{\Pi=x}(L) - p_c)L^{1/\nu}\right] \tag{9}$$

which we solve as

$$(p_{\Pi=x}(L) - p_c)L^{1/\nu} = \Phi^{-1}(x) = C_x$$

and finally rewrite as

$$p_{\Pi=x}(L) - p_c = C_x L^{-1/\nu} \tag{10}$$

From equation 10 we see that we can easily measure ν by doing a double logarithmic plot. This should give a straight line which has the incline $\frac{-1}{\nu}$.

If we now reverse the situation we can find p_c by plotting $L^{-1/\nu}$ as a function of p . This should also give a straight line which intersects the y-axis at p_c . An example of this is shown in figure 11.

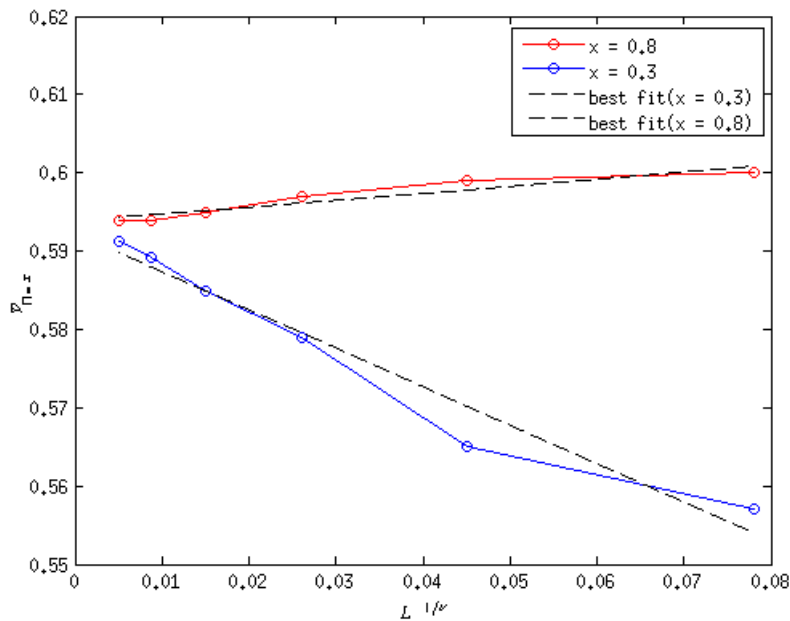


Figure 11: Estimating p_c . This is only a rough estimate, where some 5-10 matrices have been used to average the measurements. This plot puts p_c somewhere in the interval $[0.5923, 0.594]$

18 Subsets of the spanning cluster

To look at subsets of the spanning cluster we will need to look at flow on the spanning cluster. The spanning cluster typically has an immensely complex geometry where there may at some places be a high degree of connectivity, and at other places there may be connectivity through only one narrow channel. These places will, during simulations of flow on the spanning cluster be places where all the “liquid” will have to pass through. We call these places the singly connected bonds.

Other subsets of the spanning cluster include dangling ends, which in principle are simply dead ends, and so-called blobs which is a place where we have two “singly” connected bonds connecting to each other.

We can measure the singly connected bonds by placing (random) walkers on the spanning cluster from two directions and let them walk across the spanning cluster. The singly connected bonds will now have been visited by both walkers. We typically measure the singly connected bonds by their mass (area), and we want to find out how they scale with system size. This is shown in figure 12. We see that this as well behaves as a power law with the exponent $M_{SC} \propto L^{D_{SC}}$, where we measure $D_{SC} \simeq 0.7745$.

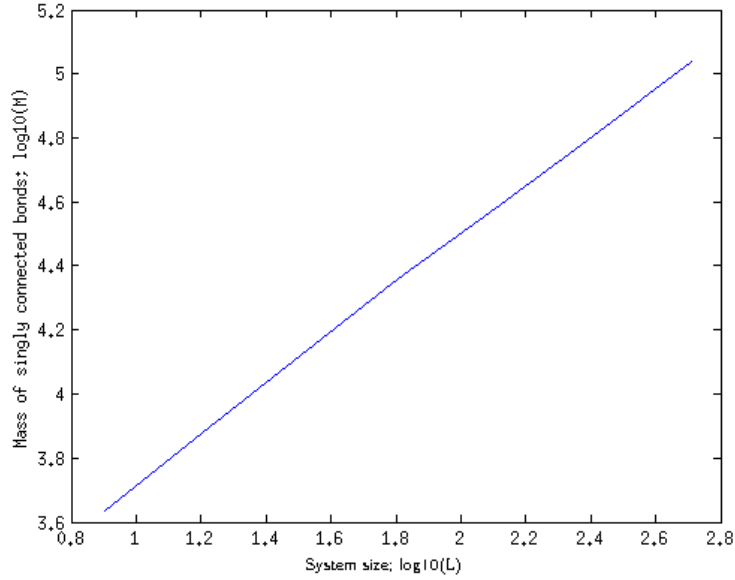


Figure 12: Mass scaling of the singly connected bonds for increasing system size. Estimate of MSC as a function of L for $L = 2.(3:9) \cdot 500$ samples used per L .

19 Random walks

Behaviour of $\langle r^2 \rangle$ of a random walker on the percolating cluster. Compare with diffusion in a bulk fluid and in a nanoporous system.

Generally we can say that as long as the cluster we land upon is a spanning cluster, $\langle r^2 \rangle$ will (after a brief overlinear period) scale linearly with the number of steps just like