Exam Notes—FYS4460

Alexander Fleischer

June 13, 2016

CONTENTS

A. MOLECULAR DYNAMICS

1. **Molecular-dynamics algorithms**

Discuss the algorithms for molecular-dynamics modeling: Potentials, integration, cut-off, periodic boundary conditions, efficiency improvements.

1.1 Potentials

The potentials describe the interactions between objects through $\mathbf{F} = -\nabla U$. Two particles may interact on each other, two-body interactions, which is the simplest potential. If three or more particles interact on each other, we have a many-body interaction.

In MD, we have looked at the *Lennard Jones (LJ) potential*

$$U(r) = 4\epsilon\left[\left(\frac{r}{\sigma}\right)^{12} - \left(\frac{r}{\sigma}\right)^{6}\right] \tag{1.1}$$

LJ is a simple and good starting point for MD calculations. It describes neutral atoms. Potentials are usually created specifically for a problem, and thus there are many different ones. Examples are *Stillinger-Weber* and *ReaxFF*.

1.2 Integration Schemes

The equations of motion describes how the system evolves with time

$$v'(t) = a(t) = -\frac{\nabla U}{m} \tag{1.2}$$

$$x'(t) = v(t) \tag{1.3}$$

where x and v describes all degrees of freedom in the system.

To integrate, we use the *velocity verlet* algorithm, which is similar to the *leapfrog algorithm*. Leapfrog has a staggered grid, which means that the velocity and position are updated at different times (integer and integer $^1/_2$).

Velocity verlet has the following scheme

$$v_{i+1/2} = v_i + a_i\Delta t/2 \tag{1.4}$$

$$r_{i+1} = r_i + v_{i+1/2}\Delta t \tag{1.5}$$

$$v_{i+1} = v_{i+1/2} + a_{i+1}\Delta t/2 \tag{1.6}$$

Calculating the force is the most costly operatiion. This is because we must calculate the force for each pair of particles. The leapfrog algorithm has the same error ($\mathcal{O}(\Delta t^2)$) as velocity verlet. The algorithm is time reversible, which is useful. Compared to *Runge-Kutta 4 (RK4)* it is more stable in the long term.

1.3 Cut-off

We have a potential that makes us calculate the forces between two particles at a time ($\mathcal{O}(N^2)$). This makes the force calculation costlier than for example updating the position and velocities, which are a constant, $\mathcal{O}(N)$, number of operations.

The force is however not effective over long distances ($r > 3\sigma$). We see this when we plot the potential, as it flattens, and thus the force is basically 0. This means we can get away with not calculating particle pairs that are far apart from each other.

We use *neighbor lists* to keep track of which particle pairs should not be calculated. Two kinds of neighbor lists exist: cell lists and verlet lists. They keep track of the particles in linear time, so the order of calculations is reduced to $\mathcal{O}(N)$, since the particle density $\rho = {}^{N}\!/_{V}$ is constant. Keeping track of the particles is crucial when dealing with any large-scale MD system.

We must also scale up the potential so that it is continious at the cut-off distance.

1.4 Periodic Boundary Conditions

Our simulation is a set of particles in space. We need *something* to happen when particles hit the 'edge'. Walls could be used, or the particles could just float into space.

To study this system effectively however, we apply periodic boundary conditions. This means that we imagine putting infinitely many similar systems next to each other. When a particle exits the confined space on one side, a particle from the imagined box on the other side enters. This is achieved using the *minimum-image convention* to calculate distances between individual particles. A particle that leaves the simulation box interacts only with the closest image particle. This restricts the interactions two half the size of the box.

1.5 Initialization of the System

Our system is a cube, consisting of smaller cubes/cells. We avoid placing the particles too close together by putting one on each face of the cubes. This corresponds to the crystalline structure of Argon, and is the called a *face-centered cubic lattice (FCC lattice).*

To start of the system, we give each particle a randomly generated velocity vector. The velocities are generated from the Boltzmann distribution, as that is equilibrium distribution of the system.

We must also counter the fact that the system will move in some general direction, called initial linear momentum.

It is important that we let the system reach thermal equilibrium before we do any measurements. We can do this by letting the simulation run until it is practically in equilibrium.

Let N be the total number of particles, V the volume of the system and E the total energy. We sometimes refer to the *microcanonical ensemble* as the *NVE ensemble*. The reason is that these quantities are *constant* in our system. They can't be changed during the simulation, and thus for example the correct value for particle density $\rho = {}^{N}\!/_{V}$ is given by N and V being correct. The energy must also be correct, and at the start of our simulation, the potential energy is unproportionally low compared to the kinetic energy. This stabilizes as the system equilibrates. Therefore the temperature drops. To achieve the right temperature and total energy, we use a *thermostat* during initialization. The function of the thermostat is to achieve constant average temperature, while allowing fluctuation of temperature with a distribution typical for a canonical ensemble.

1.6 Improving the Efficiency

In addition to neighbor lists and the cut-off mentioned earlier, we use the fact of Newton's third law. It states that $\mathbf{F}_{ij} = -\mathbf{F}_{ji}$, thus we only have to do half of the force calculations.

The action of letting the system equilibrate takes time. Therefore we can save this state, and then do further experiments a number of times from this point.

MD is an area with good possibilities for splitting up calculations and doing them in parallel through *OpenMP* and *MPI*.

2. Molecular dynamics in the microcanonical ensemble

Discuss initialization and initializtion effects. Temperature measurements and fluctuations. Comment on use of thermostats for initialization.

2.1 Microcanonical Ensemble

The *microcanonical ensemble* is the ensemble where the conditions are the following:

1. Constant number of particles N
2. Constant volume V
3. Constant total energy E

That is, if we put N particles in a box of volume V with energy E, the ensemble consists of all microstates we could get in this system.

Assume we let this system evolve over time. Then, the *fundamental assumption of statistical mechanics*, says that all microstates are equally probable in this system. This lets us calculate macroscopical quantities such as free energy, heat capacity, entropy and more, given that system is at thermal equilibrium, and that we take the ensemble average.

MD is built on the idea that some small part of the system will be representative for the whole, as long as we simulate for a long enough period of time. Time averages of the system should then be a good approximation for ensemble average. (Ergodic hypothesis.)

2.2 Initialization

We must initialize the position and velocity of the atom. The constant properties V and N must also be set, and carefully chosen to give a particle density of the real system. Or else we can't user the results.

The potential, and thus the force, will be way too large if we place the particles too closely. Our system is a cube, consisting of smaller cubes/cells. We avoid placing the particles too close together by putting one on each face of the cubes. This corresponds to the crystalline structure of Argon, and is the called a *face-centered cubic lattice (FCC lattice)*.

To start of the system, we give each particle a randomly generated velocity vector. The velocities are generated from the Boltzmann distribution, as that is equilibrium distribution of the system.

We must also counter the fact that the system will move in some general direction, called initial linear momentum.

It is important that we let the system reach thermal equilibrium before we do any measurements. We can do this by letting the simulation run until it is practically in equilibrium.

2.3 Temperature Measurements and Fluctuations

The equipartion theorem states that at thermal equilibrium, every quadratic degree of freedom in the Hamiltonian of the system must have an average energy of $1/2 k_b T$. Every atom in the system has three quadratic degrees of freedom in the kinetic energy, and there are N atoms, thus

$$\langle E_k \rangle = \frac{3}{2} N k_b T \tag{2.7}$$

This is the ensemble average of the kinetic energy, and as we saw earlier, the time-average for our system should be the same. The estimated temperature is then

$$T = \frac{2}{3}\frac{\langle E_k \rangle}{N k_b} \tag{2.8}$$

We can now define the instantaneous temperature or the kinetic temperature

$$T_k = \frac{2}{3}\frac{E_k}{N k_b} \tag{2.9}$$

This must be distinguished from the real thermodynamic temperature.

Since the microcanonical ensemble is constant in volume, particle-number and energy, the balance between potential and kinetic energy can be altered. This means we must take the ensemble average of the kinetic energy to get the temperature.

In our simulation, we know that the total energy should be conserved, but the kinetic energy will fluctuate. That is why we calculate the time-averaged kinetic energy. When we let $N, V \to \infty$ while ρ is fixed (thermodynamical limit), the relative fluctuations of the kinetic energy are diminished. Thus the temperature is constant. This concludes that the system temperature T is a macroscopic property, and that T_k will change a lot for small systems.

2.4 Thermostats for Initialization

The velocities obtained from the Boltzmann distribution used a temperature T_0, but this is not the equilibrium temperature. This is due to the fact that initial kinetic energy is transformed into potential energy before we reach equilibrium.

In our simulations, we want to be able to choose a temperature and prepare the system to achieve this. We use a thermostat for this. The thermostat adjusts the temperature of the system. Since we regulate the energy by using a thermostat, the system is not a microcanonical ensemble while the thermostat is at work. After the target temperature (and energy) is reached, we stop the thermostat, and the system is again a microcanonical ensemble.

3. Molecular dynamics in the microcanonical ensemble

How to measure macroscopic quantities such as temperature and pressure from a molecular-dynamics simulation. What challenges do you expect? What can it be used for?

3.1 Microcanonical Ensemble

The *microcanonical ensemble* is the ensemble where the conditions are the following:

1. Constant number of particles N
2. Constant volume V
3. Constant total energy E

That is, if we put N particles in a box of volume V with energy E, the ensemble consists of all microstates we could get in this system.

Assume we let this system evolve over time. Then, the *fundamental assumption of statistical mechanics*, says that all microstates are equally probable in this system. This lets us calculate macroscopical quantities such as free energy, heat capacity, entropy and more, given that system is at thermal equilibrium, and that we take the ensemble average.

MD is built on the idea that some small part of the system will be representative for the whole, as long as we simulate for a long enough period of time. Time averages of the system should then be a good approximation for ensemble average. (Ergodic hypothesis.)

To measure the macroscopic quantities temperature and pressure from a MD simulation, we have the following steps:

- Initialize the system and equilibrate
- Do measurements on the system for a long time period.
- We can estimate the ensemble average using the time-average.

We have to use what we, which in our simulations are the velocity and position of all the particles.

3.2 Temperature

The equipartion theorem states that at thermal equilibrium, every quadratic degree of freedom in the Hamiltonian of the system must have an average energy of $^1/_2 k_b T$. Every atom in the system has three quadratic degrees of freedom in the kinetic energy, and there are N atoms, thus

$$\langle E_k \rangle = \frac{3}{2} N k_b T \tag{3.10}$$

This is the ensemble average of the kinetic energy, and as we saw earlier, the time-average for our system should be the same. The estimated temperature is then

$$T = \frac{2}{3} \frac{\langle E_k \rangle}{N k_b} \tag{3.11}$$

3.3 Pressure

Clausius' virial theorem[1] gives us an estimate of the average pressure

$$P = \rho k_b T + \frac{1}{3V}\left\langle \sum_{i<j} \mathbf{F}_{ij} \cdot \mathbf{r}_{ij} \right\rangle = \frac{1}{3V}\left\langle \sum_{i} \mathbf{F}_{i} \cdot \mathbf{r}_{i} \right\rangle \tag{3.12}$$

which is as earlier an ensemble average. We can calculate it using the time average as mentioned.

3.4 Challenges

Given the estimate above, we need to take the time average. Only looking at instantaneous values of the quantities would give us very fluctuating variables. The relative size of the fluctuations would die out as the size of the system increased, so the definition of pressure and temperature are macroscopic—as they are only valid in the thermodynamical limit $N, V \to \infty$ while $\rho = constant$. We could calculate and plot the instananeous values against time, but this could be conceptually dangerous.

We are measuring a finite system and this is a challenge. In the thermodynamic limit we look at the volume V and pressure P approaching infinity while ρ is held constant. There is a difference between these two approaches, and it can be shown that the relative fluctuations are of order ${}^{1}/_{\sqrt{N}}$. Furthermore, in the thermodynamical limit, temperature and pressure are constant and well-defined, while in a finite system they are macroscopic quantities and thus not always well-defined.

A challenge when measuring the pressure is calculating the virial correctly. Looping over all the particles can be difficult due to the periodic boundary conditions and minimum image convention. The initial number of calculations seems high. But when we consider cut-off and the fact that we can do the measurements as we are calculating the force, we see that we can reduce calculations to $\mathcal{O}(n)$. Calculating the temperature correctly is also essential for the pressure.

As we are working with a set of particles in space, the whole system is prone to *drifting*—the center of mass has a velocity in some direction. This affects the temperature since T is defined by the internal motion. We counter drifting in our initialization process, but using a thermostat may give the center of mass motion again. This motion must be removed before doing temperature measurements. An example of this error is the *flying ice cube effect*, where a thermostat messes with the centre of mass motion, and the result is a system with no internal motion, but net drift. Thus resembling an ice cube flying through space.

3.5 Use of Measurements

- We can estimate real equilibrium values for real systems. For example, we can estimate equation of state, phase transitions etc.
- We can gain insight into statistical properties such as the variance of the fluctuations.
- We can study system that are experimentally challenging or under conditions which are difficult to create experimentally.

[1] The virial W of a real gas is the sum of the virial from an ideal gas $W_1 = -3PV$ and from particle interaction $W_2 = \sum_{i<j} F_{ij} \cdot r_{ij}$. The total virial is $W = -3Nk_bT$, which given $\rho = {}^{N}/_{V}$ and the expression for $W_1 + W_2$, implies eq. (3.12).

4. Measuring the diffusion constant

How to measure the diffsion constant in molecular-dynamics simulations and discuss limitations and challenges. Compare with methods and results from random-walk modelling.

4.1 How to Measure the Diffusion Constant

Diffusion is the process of particles mixing due to their random motion. It's described by Fick' law

$$J = -D\frac{\partial \phi}{\partial x} \tag{4.13}$$

with D as the diffusion constant/coefficient. In physics terms it's the proportionality factor between the concentration gradient and the net diffusion flux.

In our MD system there isn't a concentration gradient and thus no net flux. However, there is an internal microscopic motion of the particles, a self-diffusion in the system. If we look at a single particle, it will not tend to move in a general direction, so the expected value of the position is zero, $\langle r \rangle = 0$. Though the particle doesn't have a tendency to move in some direction, the actual distance travelled is not zero. Therefore the *mean displacement* is a measure of this quantity, given by $\langle r^2 \rangle(t)$.

The macroscopic diffusion can be predicted since it is the motion of all the individual particles in a gas. To model this, we use *random walkers*. A walker is a model of a single particle, discretized to take steps in some directions. For each step, it has a probability to move in some direction or another. The displacement of a single walker is then the sum of the displacements $\mathbf{x}_i$ of all the n steps,

$$\mathbf{X}_n = \sum_{i=1}^{n} \mathbf{x}_i \tag{4.14}$$

We can study this system both analytically and through Monte-Carlo simulations. For most probability distributions of x_i, we let it be symetrical, such that the probability for a step in any direction is equal. When the distribution is symmetric, it is easy to show that the expected displacement is zero and that the mean displacement is proportional to the number of steps,

$$\langle \mathbf{X}_n \rangle = 0, \quad \langle \mathbf{X}_n^2 \rangle \propto n \tag{4.15}$$

A walker that moves in d dimensions, for example a particle in a box, has a proportionality factor of $2d \cdot D$. In a Monte-Carlo simulation with a large number of random walkers moving in a box with a concentration gradient, we predict the diffusion from the random distribution.

The random walk model is of course different from our system, in that the particles do interact. Luckily this doesn't matter, and the interesting part is the mean displacement. The fact that particles interact is embedded into the measure of the displacement. Because the system consists of a large number of particles, we expect that the particles average (squared) displacement estimates the expectancy of a single particle (squared) displacement in a good way. We trace the motion of every atom and can therefore estimate the diffusion constant from the mean displacement squared (in one dimension)[2] from all the particles by

$$\langle x^2(t) \rangle = \frac{1}{N}\sum_{i=1}^{N} (x_i(t) - x_i(0))^2 \tag{4.16}$$

By plotting the mean distance, we can find the slope by linear regression to estimate the diffusion constant D since they are proportional. Figure 4.1 shows us that the mean distance from the starting position of all walkers is proportional to $\sqrt{n}$. We will not see this behavior for only one walker—thus increasing the number of walkers will make the curve approximate $\sqrt{n}$.

[2] The mean displacement squared in three dimensions is of course $\langle r^2 \rangle = \langle x^2 \rangle + \langle y^2 \rangle + \langle z^2 \rangle$.

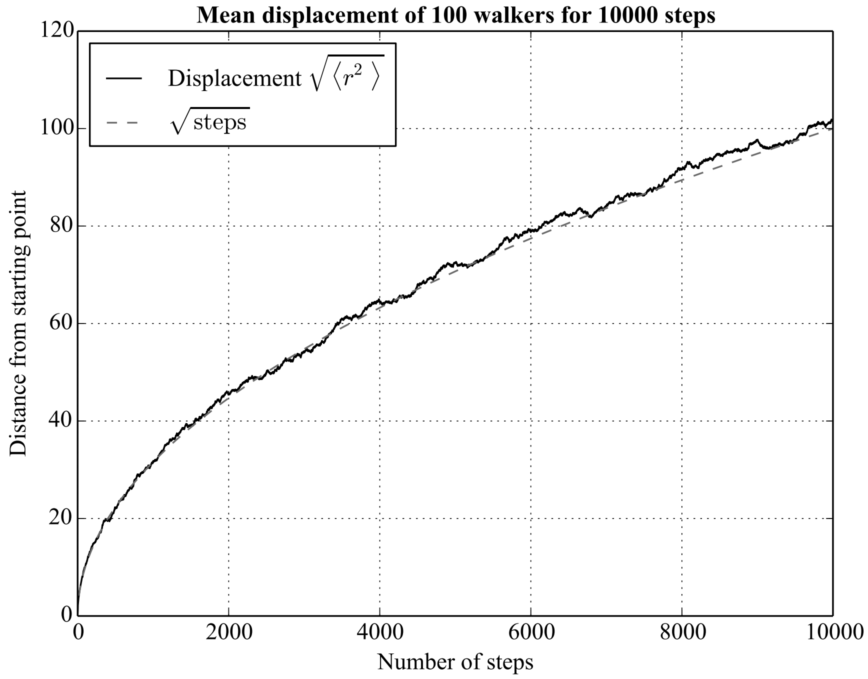


Figure 4.1: Mean distance from starting point for all N particles after n steps.

4.2 Limitations and Challenges

The particles in our simulation are bound by the minimum image convention, where each particle in the box has a 'mirror particle' outside to enforce periodic boundary conditions. Thus we can't measure the real displacement unless we take this into account. One way to solve the problem is to give the particles a displacement property d, a three-dimensional vector, whcih keeps track of the displacement. The displacement is then updated in the same way as the position in the velocity verlet algorithm

$$d_{i+1} = d_i + v_{i+1/2}\Delta t \tag{4.17}$$

With this solution, we are also free to set the initial position $d_0 = 0$, instead of keeping track of $r(0)$ for every particle. The periodic boundary conditions are not enforced on the displacement vector, and has no impact on the system—it just records the mean displacement while r_i is the particle position.

We still have to correctly initialize the system and thermal equilibrium is needed before we start recording the mean displacement. It is also important to initialize the system in the correct phase. If we look at a system of a solid, there won't be much to diffusion to be seen.

In a nano-porous system it is not obvious how we could measure the mean displacement. In the nano-porous system we have some closed-off pores and the particles in these won't contribute to the transport through the system. Should we then focus only on the particles that are able to go from one side of the system to the other?

5.

Measuring the radial distribution

How can you measure the radial distribution function? What does it tell? What challenges will you face? compare the measurement of the radial distribution function to the measurement of the probability densities for a random walk.

5.1 The Radial Distribution Function

Consider a particle in our system. The *radial-distribution function*—also known as the *pair-correlation function*—$g(r)$ is the probability that we will find another particle some distance r away from our reference. If we imagine a shell around the particle, the distribution function is the density of that shell with radius r. One could measure this distribution by counting all particles in the shell for increasing r. The distribution is normalized with respect to uncorrelated particles, in other words an ideal gas,

$$\lim_{r\to\infty} g(r) = 1 \tag{5.18}$$

The local time-averaged particle density is then found as $\rho g(r)$.

5.2 Measuring the Distribution Function

The fundamental properties that we have in our simulations are the positions and velocities of every particle. To measure the radial-distribution function $g(r)$, we have the following procedure[3],

- Choose a reference particle
- Calculate the distances to all other particles
- Bin the distances appropriately
- Create a histogram that is an estimate for $g(r)$

If we do this procedure with all particles as references, and take the average, we will have a good estimate. To this more efficiently, we should instead calculate the distances between all particle-pairs, bin them, and then create the histogram. We must remember to divide by the volume, since we are interested in the particle density we are interested in.

5.3 Using the radial-distribution function

We can measure the distribution function experimentally by looking neutron and x-ray scattering of matter, and thus verify the numerical methods.

The structure of the system we are looking at tells us what kind matter we are looking at. Fluids will have other forms than solids. Therefore the distribution function gives us information about the average particle density and therefore the type of structure. A solid will for example have a typical lattice formation, and the system of Argon atoms that we looked would give us clear spikes—defining the FCC lattice structure we used—with just small vibrations. Increasing the temperatures gives the particles larger kinetic energies, and the spikes will be broader. Even higher temperatures, such that the system becomes a gas, will yield a spike at the beginning and less form as the radius increases. The reason for the first spike is the strong repulsive forces for small distances.

The radial-distribution function lets us see a link between the macroscopic and microscopic properties of the matter. This is due to our use of microscopic information to say something about the macroscopic properties.

[3] See below the procedure for a more effective method.

5.4 Challenges

Computing the distribution function is computationally intensive. We loop over all particle pairs, and thus incur $\mathcal{O}(N^2)$ operations and a large dataset. Doing this for many time-steps is costly operation-wise. To counter this, we could use the approach of cut-off distances: for some $r > r_{\text{cut}}$ we stop doing the calculations. In this case, we assume that distances past r_{cut} implies $g(r) \sim 1$. The already integrated cell-list functionality for force-calculations can then be used to achieve this.

If we choose bin sizes that are too small, we might divide by a very small volume and obtain errors because of machine representation of floating point numbers. It might be difficult to know how to choose the bin sizes. One could do it linearly with respect to r, or some exponential of r.

6. Thermostats

Discuss the microcanonical versus the canonical ensemble in the simulations. How can we obtain results from a canonical ensemble? Introduce two thermostats, and describe their behavior qualitatively. How can you use such a thermostat for rapid initialization of a microcanonical simulation?

6.1 The Canonical Ensemble and Thermostats

The microcanonical ensemble describes a system where the volume V, number of particles N and total energy E is held constant. This is different to the canonical ensemble where the system is permitted to exchange energy, but the temperature T is held constant. Generally in molecular dynamics, we want to study a microcanonical ensemble. However, sometimes it is of interest to simulate the canonical ensemble. To achieve this we need something to keep the temperature constant. We use a thermostat to do this: an external heat bath in contact with our system.

A thermostat should satisfy some requirements:

1. It should be able to control the temperature of the system such that it is close to the temperature of the heat bath.
2. It should not disturb the system dynamics, and it should sample the quantities of the phase space (positions and velocities).
3. It should be adjustable.

Our best bet for a thermostat that satisfies these demands is the *Nosé-Hoover thermostat*. We have already discussed this thermostat, so we'll look at two others instead.

6.2 The Berendsen Thermostat

Many thermostats rescale the velocities of the systems' particles by a factor γ. One of these is the *Berendsen thermostat*. The scaling factor is

$$\gamma = \sqrt{1 + \frac{\Delta t}{\tau}\left(\frac{T_{\text{bath}}}{T} - 1\right)} \tag{6.19}$$

Here τ is called the *relaxation time*: which lets us adjust the (weak) coupling to the heat bath. A greater τ value gives a weaker coupling. This means that the Berendsen thermostat works by dampening the fluctuations of the systems' kinetic energy. The velocity scaling is such that change in temperature is proportional to difference in temperature between the heat bath and the system.

$$\Delta T = \frac{\Delta t}{\tau}(T_0 - T(t)) \tag{6.20}$$

The thermostat is not able to represent a correct canonical ensemble, but for large enough systems, we will achieve approximately correct results. It is often used because it effectively thermalizes the system to the temperature of the heat bath, and can be used to initalize a system before applying the Nosé-Hoover thermostat.

6.3 The Andersen Thermostat

Another simple thermostat is the *Andersen thermostat*. It works by simulating collisions between particles in the system and in the heat bath. For every particle in the system we generate a random number uniformly generated between 0 and 1. If the number is less than ${}^{\Delta t}/_{\tau}$, we let the particle be one of the colliding particle. Thus the particle needs a new velocity, which is generated from a Boltzmann distribution with standard deviation $\sqrt{k_B T_{\text{bath}}/m}$. We can adjust the thermostat by changing τ, which can be considered the *collision time*. The collision time describes the strength of the coupling beween the system and the heat bath.

Assuming ergodicity, the Andersen thermostat gives us a canonical distribution of microstates. The Algorithm is non-deterministic and thus not time-reversible. It should only be used when measuring time-independent properties. Diffusion, which is something we have looked at, should not be calculated if the system has been influenced by the Andersen thermostat.

The thermostat is useful for equilibrating the system, but disturbs the natural vibrations of the crystal structure.

In project 1 we experienced that the Berendsen thermostat gave smaller fluctuations in the system than the Andersen thermostat.

B. ADVANCED MOLECULAR DYNAMICS

7. Generating a Nanoporous Material

Discuss how we prepare a nanoporous matric with a given porosity. How do we characterize the structure of such a material and the dynamics of a fluid in such a material?

7.1 Simulating Nanoporous Materials with MD

Nanoporous materials are porous materials with pores less than ~ 100 nanometers. We want to simulate fluids going through the pores of the material and so we need to create a solid matrix with pores. The solid material is then filled with a fluid, and we can do normal MD measurements on it. It could for example be a Silicon dioxide block with water in it.

7.2 Generating the Solid Matrix

We usually generate the matrix in the 'opposite' way of what an experimental physicist would do. By creating a bulk material of for example Argon, and then 'freezing' part of it so it becomes a solid, while the rest of the atoms are kept liquid. In percolation theory we have a probability p of which atoms to freeze. This would give us too small pores, so we should use another approach. Like a pumpkin, we could carve out parts of the solid—either with randomly distributed spherical pores of some radius or using cylindrical pores—removing the atoms (either the ones inside or outside the spheres) until we have the wanted porosity. An approximation to the porosity is the ratio between the particles inside and outside the spheres, $\phi = 1 - N_{\text{matrix}}/N(= V_{\text{pore}}/V)$, since the porosity is the relative part of the total pore-volume. The fluid is then the atoms which have not been frozen, and we should reduce the density of these.

7.3 Characterizing the Material-structure and Dynamics of the Fluid

The fluid in the system can be characterized by two properties:

- Porosity ϕ: The ratio between pore-volume and total volume.
- Permeability k: The ability of a fluid to travel through the porous medium. It is a function of the size of the pores, the connection between pores and the tortuosity of the medium.

The permeability k is measured by simulating the flow through the system. We do this by applying a pressure difference to the system. The permeability is affected by the porosity which allows transfer, the size of the pores, and the amount of twists and turns: the *tortuosity*.

In addition to the above properties, which are properties of the material, the flow also depends on the properties of the fluid, for example the density.

We can measure the flow by asserting a pressure gradient ∇P, and the flow is described by *Darcy's law*[4],

$$U = \frac{k}{\mu}(\rho g - \nabla P) \tag{7.1}$$

where ρg is the gravitational part. The pressure gradient is from hydrostatic conditions. μ is here the viscosity of the fluid.

To generalize this to our needs, we rewrite the gravitational term to a global force by

$$\rho g = \frac{N}{V} m g = \frac{N}{V} F_g = n F_g \tag{7.2}$$

[4] The derivation of this expression comes from the simplified Navier-Stokes equation ('Stokes equation'), `https://en.wikipedia.org/wiki/Darcys_law#Derivation`

We can now switch out the gravitational force F_g by a general force-term F_x. We dub the number of particles per volume as the *number density* ${}^{N}\!/_{V} = n$. If we ignore the pressure gradient, our equation now reads

$$U = \frac{k}{\mu} n F_x \tag{7.3}$$

To find the remaining unknowns, we look at the known velocity profile for a cylindrical pore of length L and radius a,

$$u(r) = \frac{\Delta P}{L} \frac{1}{4\mu} (a^2 - r^2) \tag{7.4}$$

By creating a system with only one of these pores, we can calculate the viscosity μ. We'll have to let the system reach equlibrium before we do the measurements, of course. This is when the particles on average is no longer accelerated, and the temperature only has small fluctuations. This could take some time. The velocity profile is then measured by splitting the radii into bins, finding the mean velocity and then dividing by the number of particles in the respective bin. This is plotted in a histogram.

If we now keep the same density ρ in our actual system, we should be able to use this μ in our calculations (since the viscosity is fluid dependent). Then we can find the permeability as

$$k = \frac{U\mu}{nF_x} \tag{7.5}$$

8. Diffusion in a nanoporous material

How can you measure the diffusion constant for a low-density fluid in a nanoporous system? Discuss what results you expect. Compare with diffusion in a bulk liquid and in a larger-scale porous medium.

8.1 How to Measure the Diffusion Constant

Diffusion is the process of particles mixing due to their random motion. It's described by Fick' law

$$J = -D\frac{\partial \phi}{\partial x} \tag{8.6}$$

with D as the diffusion constant/coefficient. In physics terms it's the proportionality factor between the concentration gradient and the net diffusion flux.

In our MD system there isn't a concentration gradient and thus no net flux. However, there is an internal microscopic motion of the particles, a self-diffusion in the system. If we look at a single particle, it will not tend to move in a general direction, so the expected value of the position is zero, $\langle r \rangle = 0$. Though the particle doesn't have a tendency to move in some direction, the actual distance travelled is not zero. Therefore the *mean displacement* is a measure of this quantity, given by $\langle r^2 \rangle(t)$.

The macroscopic diffusion can be predicted since it is the motion of all the individual particles in a gas. To model this, we use *random walkers*. A walker is a model of a single particle, discretized to take steps in some directions. For each step, it has a probability to move in some direction or another. The displacement of a single walker is then the sum of the displacements $\mathbf{x}_i$ of all the n steps,

$$\mathbf{X}_n = \sum_{i=1}^{n} \mathbf{x}_i \tag{8.7}$$

We can study this system both analytically and through Monte-Carlo simulations. For most probability distributions of x_i, we let it be symetrical, such that the probability for a step in any direction is equal. When the distribution is symmetric, it is easy to show that the expected displacement is zero and that the mean displacement is proportional to the number of steps,

$$\langle \mathbf{X}_n \rangle = 0, \quad \langle \mathbf{X}_n^2 \rangle \propto n \tag{8.8}$$

A walker that moves in d dimensions, for example a particle in a box, has a proportionality factor of $2d \cdot D$. In a Monte-Carlo simulation with a large number of random walkers moving in a box with a concentration gradient, we predict the diffusion from the random distribution.

The random walk model is of course different from our system, in that the particles do interact. Luckily this doesn't matter, and the interesting part is the mean displacement. The fact that particles interact is embedded into the measure of the displacement. Because the system consists of a large number of particles, we expect that the particles average (squared) displacement estimates the expectancy of a single particle (squared) displacement in a good way.

In a nanoporous system, we actually have the *same* result. The interactions affect the mean-square displacement, but it is only a mathematical fact of the connection between the microscopic mean displacement and the macroscopic diffusion. In our calculations we leave out both the solid matrix and the particles trapped in pores which aren't connected—and thus can't diffuse anywhere. Since the particles are allowed to move in a smaller area, the mean displacement is also smaller. This means that the diffusion constant must also be smaller, which is reasonable since it's harder to move through tunnels than open space. One can think of it as the dimension value d is less than 3, though we have three dimensions (thus the D value is still the same). This behavior is amplified with smaller pores. The diffusion might also vary in the dimensions. Though, if we randomly distribute the pores, these differences should disappear when the system size increases.

We trace the motion of every atom and can therefore estimate the diffusion constant from the mean displacement squared (in one dimension)[5] from all the particles by

$$\langle x^2(t)\rangle = \frac{1}{N}\sum_{i=1}^{N}(x_i(t) - x_i(0))^2 \tag{8.9}$$

By plotting the mean distance, we can find the slope by linear regression to estimate the diffusion constant D since they are proportional. Figure 4.1 shows us that the mean distance from the starting position of all walkers is proportional to $\sqrt{n}$. We will not see this behavior for only one walker—thus increasing the number of walkers will make the curve approximate $\sqrt{n}$.

8.2 Expectations

Studying precolation systems has given us a greater intuition for diffusion in nanoporous materials. We see that the systems can either have diffusion through the whole system or only in some parts, if the pores are closed off from each other. Diffusion through the whole system is also governed by the size and form of the pores.

The diffusion coefficient in a nanoporous system should always be smaller than in a bulk solid. Random motions in an 'open' area will disperse more since it's easier than maneuvering through tunnels. We must thus have an *effective* diffusion constant that takes into account the factors in a nanoporous material.

The first to take into account is the *transport-available* porosity ϵ_t: the porosity that actually leads from one side to the other—or in others words the porosity avilable for macroscopic transport of fluids. It is the total volume withough the solid matrix, the pores that are too small, dead ends and the pores that are sealed off from the rest of the system.

The second is the *constrictivity* δ. As our system consists of nanoscopic pores, there is a large contribution from surface tension along the walls. This results in a higher viscosity as the fluid has more trouble moving close to walls. The constrictivity is dependent on the size of the particles in the fluid and the pore size. It is a dimensionless scaling paramter and since the diffusion is less than in a bulk material, we have $\delta \leq 1$.

The final thing to consider for the effective diffusion coefficient is the curve-number of the pore tunnels. This is called the *tortuosity* τ, and is the number of turns a curve has, thus for a porous system we have $\tau > 1$. A system with many twists and turns will therefore have lower diffusion.

The effective diffusion coefficent can now be written

$$D_e = \frac{D\epsilon_t\delta}{\tau} \tag{8.10}$$

and we see that each of the terms lowers the diffusion independently of the others.

8.3 Comparison Between Bulk Liquid and Large-scale Porous Medium

The three systems of bulk liquid, larger-scale porous systems and nanoporous systems will all have diffusion. However the rate of diffusion will vary between the three. They all have a mean square displacement that will be proportional to time. The diffusion coefficient will be different, but they all respect Fick's law of mass transportation.

As explained earlier, we expect the diffusion coefficient of a bulk liquid to be greater than that of porous media, as we have $e_t < 1$, $\delta \leq 1$ and $\tau \geq 1$. Since downscaling a porous system to a nano-scale system will give less space in every pore, the constricitivity will be smaller, $\delta_{np} < \delta_p$. It is also possible that the tortuosity will be larger, as smaller pores could mean more tunnels and so we might have more turns, $\tau_{np} > \tau_p$.

[5] The mean displacement squared in three dimensions is of course $\langle r^2\rangle = \langle x^2\rangle + \langle y^2\rangle + \langle z^2\rangle$.

9. Flow in a nanoporous material

Discuss how to induce flow in a nanoporous material. How can you check your model, calculate the fluid viscosity and measure the permeability? What challenges do you expect?

9.1 Generating the Nanoporous Matrix

To induce flow in a nanoporous material, we must first create the matrix. We did this by creating a bulk Lennard-Jones fluid system, ‘froze’ all the particles and carved out pores of spheres obeying a random distribution, until we were satisfied with the porosity: the ratio between pore volume and the total system volume. We could now fill the matrix with a fluid, either of the same atoms as the solid, or any others. In our project we used the same Lennard-Jones fluid as the solid, but at half the density. Though the solid is frozen, the atoms contribute to the force calculations, but since they don’t move, they are ignord in the integration of the equations of motion.

9.2 Flow: inducement and measurements

The pores can either be connected or separated, but if they are connected, the fluid can move from one side to the other—a macroscopic flow through the system. In the microcanonical ensemble, we didn’t have a flow, as there was no pressure gradient in the system. By adding a global force to the system, we can create a pressure gradient and measure the induced flow.

By Darcy’s law, we have

$$U = \frac{k}{\mu}(\rho g - \nabla P) \tag{9.11}$$

U is the volume flux per area per time. k is the permeability, the ability for the fluid to flow through the system. μ is the viscosity, the ‘thickness’ of the fluid, or how well it resists changes. ρg is the gravitational term, which is the force that cause movement in the fluid. Finally we have the pressure gradient ∇P which usually comes from hydrostatic conditions[6].

We are not interested in the gravitational forces, so we replace it with a global force by

$$\rho g = \frac{N}{V} mg = \frac{N}{V} F_g = nF_g \tag{9.12}$$

In our system, we don’t have hard walls, only periodic boundary conditions, so we can remove the hydrostatic build-up of pressure,

$$U = \frac{k}{\mu} nF_x \tag{9.13}$$

We can then let the force F_x affect the system over time and measure U. The measurement must be done when the flow is stationary, which means that the particles on average is not under acceleration, and the temperature only has small fluctuations. The remaining unknowns are permeability k and viscosity μ. We must find the ratio between these, as we can’t measure them independently.

9.3 Viscosity and Permeability Measurements

There are systems where we know the permeability. One of these is where we only have *one* cylindrical pore, with radius a, length L and pressure difference between each side ΔP. We expect this system to have the velocity profile

$$u(r) = \frac{\Delta P}{L}\frac{1}{4\mu}(a^2 - r^2) \tag{9.14}$$

[6] *Hydrostatics* is the study of incompresible fluids at rest (opposed to *fluid dynamics*).

at stationary flow. The pressure difference is $\Delta P = nFL$, so $\Delta P/L = nF$. This shows that the flow is fastest in the middle of the cylinder. Near the walls we have virtually no flow, due to the interactions between walls and fluid.

We measure the velocity profile at stationary flow by splitting the radii into bins, finding the mean velocity and then dividing by the number of particles in the respective bin. This is plotted in a histogram, and we can measure the viscosity μ.

If we now keep the same density ρ in our actual system, we should be able to use this μ in our calculations (since the viscosity is fluid dependent). Then we can find the permeability as

$$k = \frac{U\mu}{nF_x} \tag{9.15}$$

9.4 Challenges

The viscosity μ is as stated fluid dependent, and changing the density will therefore alter the viscosity. It can take a while to reach stationary flow (steady state), and this *must* be done in order to get correct results. The flux value U must be measured consistently. We can't in one moment measure the volume flux, and in the next measure the area and time flux.

C. PERCOLATION

10. Algorithms for percolation systems

How do we generate a percolation system for simulations? How to analyze and visualize the systems? How to find spanning clusters and measure the percolation probability?

10.1 Generating the Percolation System

We generate a percolation system numerically by creating a matrix lattice of size $L \times L$ (or a 1D system of length L) with elements randomly chosen between 0 and 1. We then check each *site* on the lattice and create a new matrix. If the points are larger than some probability p, the point is occupied and we set the value to 1. Otherwise we set the value to zero. The probability for a site to be unoccupied is $q = 1 - p$, and the process is uncorrelated between sites. For two sites to be connected, we use the 'nearest neighbor' rule which means that in two dimensions we have four candidates per site (except on the boundaries).

10.2 Analysis and Visualization

The system can be visualized by plotting a grid with colored (black and white) squares for each occupied or unoccupied state. For percolation to occur, we must have a path that leads from one side to the other. That is, there must be neighboring squares all the way across.

A set of neighboring squares is called a *cluster*, and a cluster that spans all the way across is a *spanning* cluster. The percolation *threshold* p_c is when we get a spanning cluster, or in other words: a percolationg system. For an infinite system, this value is set, but for finite systems, we have a new p_c for each realization.

To find a spanning cluster in a percolation system, we must first find the clusters in the system. This is done by checking all neighbors of a square and labeling them with a number. This can be done with automated methods in Python or MATLAB, but we can also write a brute-force algorithm.

When all sites are given a label, we can plot the matrix with each cluster having a distinct color. The colors should be randomized so it's easier to separate the clusters (color gradients are generally a bad idea). For larger systems we will get a lot of clusters and so it's even more important.

To check if a cluster is a spanning cluster, we could implement a *minimum bounding box* around the cluster. That is a rectangle (in two dimensions) that encloses all the sites. If either the width or height of the rectangle matches the length L, we have a spanning cluster. Fig 10.1 shows this.

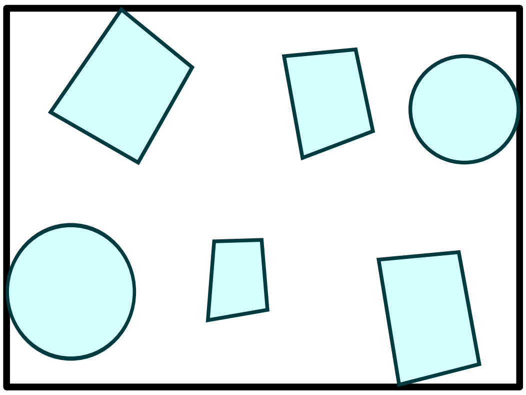


Figure 10.1: An example of a minimum bounding box.

The minimum bounding box is well-suited for $L \times L$ geometries, but not so much for more complicated ones. For these geometries it is better to look at the set of clusters that borders one side, and another set that borders the other. The intersection between these two sets is the set that contains our spanning clusters.

10.3 Percolation Probability

The probability of there being a spanning cluster in the system is given by the percolation probability $\Pi(p, L)$. If we generate one matrix, and checks if it percolates, we of course have the answer: yes or no. Therefore we must do many experiments to find out the percolation probability. We can find this probability in a set of finite systems by counting the number of times a set of random lattices percolate for different p values. That is:

1. Choose the number of experiments N.
2. Create a set of random lattices and check if they percolate for a set of values p_i.
3. N_i is the number of percolating systems we have for a given p_i.
4. Then we have $\Pi(p_i, L) = {}^{N_i}/_{N}$.

As the length $L \to \infty$, the percolation probability should become 1 after some $p = p_c$, the *percolation threshold*. For a 1D system, this is of course when $p = 1$, since we only have percolation when all sites are filled.

11. Percolation on small lattices

Discuss the percolation problem on a 2×2 lattice. Sketch $P(p, L)$ and $\Pi(p, L)$ for small L. Relate to your simulations. How do you calculate these quantities and how do you measure them in simulations?

11.1 Small Lattices

We are interested in finding the percolation probability $\Pi(p, L)$ and the cluster spanning density $P(p, L)$ for matrices of size $L \times L$. The smallest L-values give us matrices that we can easily draw and see what the probability and spanning cluster density is. The results are analytic solutions (closed-form), but for larger systems, it becomes difficult due to system size of $2^{L\times L}$. We want to study these small systems anyway as they give clues as to how the system sizes evolve, and thus we can relate them to results from numerical solutions for larger systems.

11.2 Configuration of 2×2 Matrix

A 2×2 lattice can have 1, 2, 3 or 4 occupied states. We can put the configurations into six groups like this:

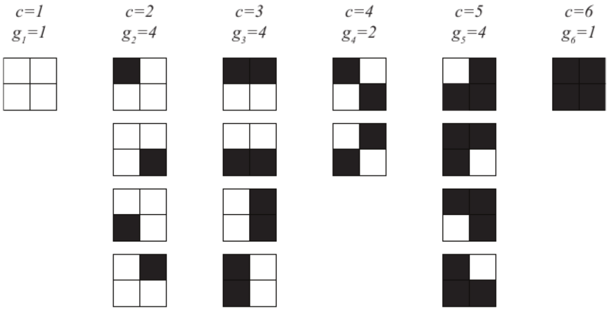


Fig. 1.8 The possible configurations for a $L = 2$ site percolation lattice in two-dimensions. The configurations are indexed using the cluster configuration number c.

If p is the probability of an occupied state, then for example $c = 0$ has the probability $P(c) = (1 - p)^4$. This is because we multiply the probability of there not being a occupied state together four times. The probabilities for all groups are given as:

$\mathbf{c}$	$\mathbf{g_c}$	$\mathbf{P(c)}$	$\mathbf{\Pi(p, L\vert c)}$
1	1	$p^0(1-p)^4$	0
2	4	$p^1(1-p)^3$	0
3	4	$p^2(1-p)^2$	1
4	2	$p^2(1-p)^2$	0
5	4	$p^3(1-p)^1$	1
6	1	$p^4(1-p)^0$	1

Table 1:

and the general expression is $p^x q^{4-x}$ where x is the number of occupied sites in the system.

The probability for a spanning cluster is[7]

$$\Pi(p, L) = \sum_c \Pi(p, L|c)P(c) \tag{11.1}$$

[7] From probability theory we have $P(A) = \sum_B P(A|B)P(B)$, which implies equation 15.32.

and the probability $P(c)$ of spanning, given a configuration c, is either 0 or 1, as we saw in table 2. This leads us to the general equations

$$\Pi(p, L) = 4p^2q^2 + 4p^3q + p^4 \tag{11.2}$$

$$P(p, L) = \frac{2}{4} \cdot 4p^2q^2 + \frac{3}{4} \cdot 4p^3q + \frac{4}{4} \cdot p^4 \tag{11.3}$$

For the spanning cluster density, we must multiply each term of the percolation probability by the size of the cluster for the given configuration. The spanning cluster density is generally $P = {}^{M_i}/_{L^2}$ where M_i is the 'mass' of the cluster.

Figure 11.2 shows us how Π and P behaves for $L = 2$. For the next few L-values, the curves are the same only (gradually) more distinctive, the Π curve is more S-shaped and the P curve is more curved.

Both curves increase from $p = 0$ to $p = 1$ and are smooth. P is however always below $y = p$. This is due to the fact that the spanning cluster density is the probability that a random site is part of the spanning cluster. The random site is not always occupied, so we must have $p \geq P$. As the $p \to 1$, the difference between the two vanishes. Of course, for $L = 2$, if there is a spanning cluster, all occupied sites are part of the cluster, but in larger systems we can have non-spanning and spanning clusters at the same time.

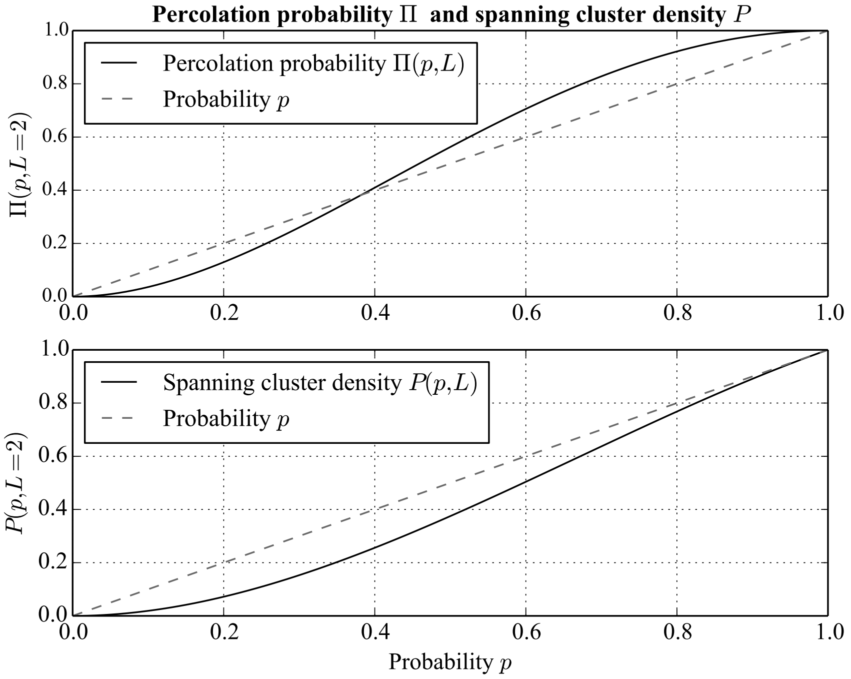


Figure 11.2: Sketch of probabilities and cluster densities for $L = 2$. The $y = p$ curve is the same as for $L = 1$ in both cases. For the next L-values (3 and 4) the curves are the same only more 'dramatic'

11.3 Numerical Solutions of Π and P

The size of a two-dimensional percolation system is $2^{L\times L}$. For larger systems, we can't manually find the the analytical solutions of Π and P. However, we can find them using Monte-Carlo methods. We generate N matrices of size $L \times L$ and check if they percolate or not. n is the number of matrices that do percolate. Each system has a mass of M_i, and thus we estimate Π and P as

$$\Pi(p, L) = \frac{n}{N}, \quad P(p, L) = \frac{\sum_i M_i}{NL^2} \tag{11.4}$$

12. Cluster number density in 1D percolation

Define the cluster number density for 1D percolation, and show how it can be measured. Discuss the behavior when $p \to p_c$. How does it relate to your simulations in two-dimensional systems?

12.1 Definition

A 1D system will only percolate if all sites are occupied, like water through a pipe will only if flow if there are no obstacles.

Imagine a cluster of size s in a 1D pipe. A specific site in the cluster could be the first site from the left. The probability for a site to belong to a specific site in a cluster of size s is called the *cluster number density*, given by $n(s, p)$.

The probability of a site being occupied is p. For all other sites in the cluster to be occupied, we have the probability p^{s-1}. Now we also need to make sure that these points are actually grouped together. The sites on each side of the cluster has a probability of $1 - p$ of being unoccupied. Thus the cluster number density is

$$n(s, p) = (1 - p)^2 p^s \tag{12.5}$$

These probabilities need to be summed the number of sites in the cluster, namely s. Thus, if we point at a specific site in a cluster in a lattice, the probability of that site to be part of a cluster of size s is

$$P(\text{site is part of cluster of size s}) = s \cdot n(s, p) \tag{12.6}$$

12.2 Measurements

Given a finite system, we can count the number of clusters of size s as N_s. The size of the system is L, so in a simulation we can measure the *estimated* cluster number density $\bar{n}$ as[8]

$$\overline{n(s, p)} = \frac{N_s}{L} \tag{12.7}$$

We can improve the estimate by generating M different systems and estimate the average

$$\overline{n(s, p)} = \frac{N_s}{LM} \tag{12.8}$$

Large systems yield very different s-values, but we can bin them and create a histogram. We should however create logarithmic bin sizes, so the bin-width increases as s increases. Else we would get a lot of empty bins. It is now very important to divide every bin with the correct bin size.

12.3 Convergence Behavior for $p \to p_c$

A one-dimensional pipe has to have all sites occupied in order to percolate. For an infinitely long pipe, the threeshold probability is thus $p_c = 1$.

To see the behavior when $p \to p_c$, we begin by creating the function

$$G(s) = \frac{n(s, p)}{(1 - p)^2} = p^s \tag{12.9}$$

If we plot this (see below), we see that as a function of s, the cut-off is increased when $p \to 1$.

[8] In d dimensions, we have $\overline{n(s, p)} = N_s / L^d$

Let us now visualize $G(s)$ in another way. We can rewrite

$$n(s,p) = (1-p)^2 e^{s \ln p} = (1-p)^2 e^{-\,{}^{s}/_{s_\xi}} \tag{12.10}$$

where we have defined $s_\xi = -{}^{1}/_{\ln p}$. The result is plotted below, and we now see that when we rescale the s axis, we align the graphs for different p values. We also notice that

$$G(s) \to \begin{cases} 1, & \text{if } s \ll s_\xi \\ 0, & \text{if } s \gg s_\xi \end{cases} \tag{12.11}$$

This means that our system should have clusters of size less than the characteristic cluster size, but generally few that are larger.

As $p \to p_c = 1$, $s_\xi \to \infty$. By using the Taylor expansion of $-{}^{1}/_{\ln p}$, we can approximate it as

$$s_\xi \simeq \frac{1}{1-p} \tag{12.12}$$

This means that the divergence of s_ξ is a power law with exponent -1 when $p \to 1$, and it is the case for all dimensions.

We can now write the characteristic cluster size on general form

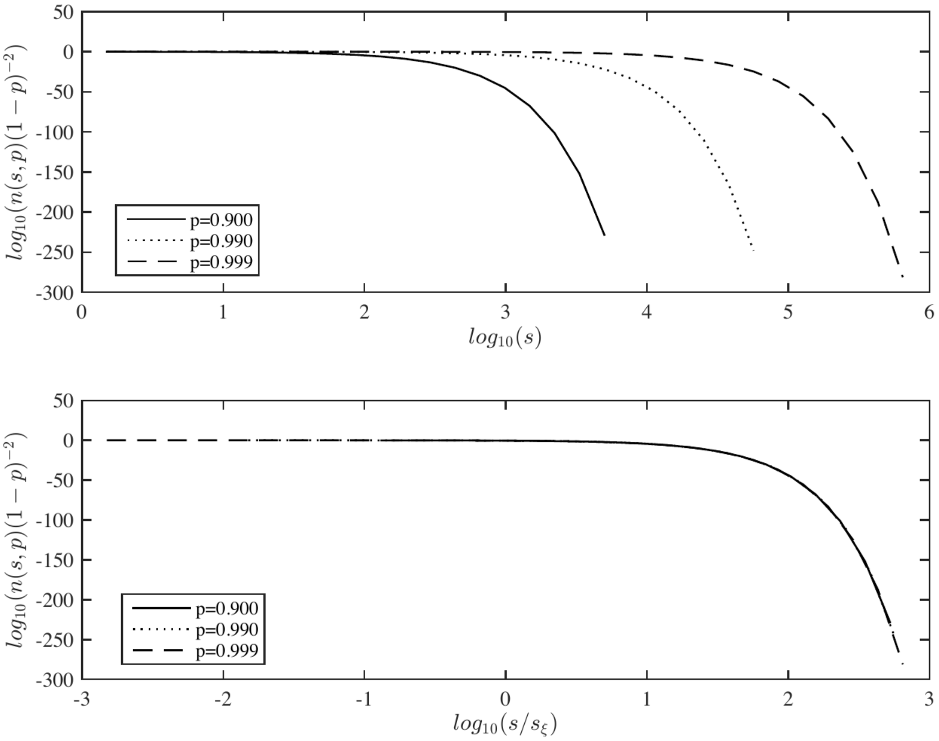
$$s_\xi \propto |p - p_c|^{-\,{}^{1}/_{\sigma}} \tag{12.13}$$

when $p \to p_c = 1$. In the one-dimensional case we have $\sigma = 1$. The divergence is a property of the infinite system.

As p approaches p_c, the infinite system is about to percolate, so the cluster sizes increases. At one point, though, we have a large (soon spanning) cluster that fuses with the other large clusters and then the characteristic cluster size drops in value again.

The σ value is dependent on the lattice dimensionality, and we can measure it by finding s_ξ for p-values close to p_c. We should see that it follows a power law. This has the challenge that our system is finite, so we could instead use the plot method from earlier. When we plot as a function of ${}^{s}/_{s_\xi}$ we see that all the lines fall onto a single curve. This is called a *data-collapse*.

In a two-dimensional case, we are able to distinguish between the characteristic size s_ξ (the mass) and the correlation length ξ.



13. Correlation length in 1D percolation

Define the correlation length ξ for 1D percolation. Discuss its behaivor when $p \to p_c$. How is it related to cluster geometry and your results for two-dimensional percoaltion?

13.1 The Correlation Length

The distance between two occupied sites a and b is denoted by r. These two sites are connected if all sites in between (there r are of them) are occupied. The probability for this is called the correlation function and in one dimension we have

$$g(r) = p^r = e^{-r/\xi} \tag{13.14}$$

where the *correlation length* is $\xi = -1/\ln p$. The correlation length can be seen as a cut-off variable for the correlation function

$$g(r) \to \begin{cases} 1, & \text{if } r \ll \xi \\ 0, & \text{if } r \gg \xi \end{cases} \tag{13.15}$$

In a given system we expect to find clusters of lengths up to ξ, but seldom clusters larger than ξ.

13.2 Divergence of the Correlation Length

When the probability $p \to p_c = 1$, the number of sites increases and thus ξ diverges.

To study this better, we rewrite the correlation function as

$$\xi = -\frac{1}{\ln p} = \frac{1}{1-p} = (p - p_c)^{\nu} \tag{13.16}$$

This is a general percolation result, where $\nu = 1$ in one dimension, and $\xi \to \infty$ and diverges as a power law when p approaches p_c. This is also the p-value where the spanning cluster forms.

13.3 Cluster Geometry

The correlation length ξ is different from the characteristic cluster size s_ξ, however this is only the case when we are beyond one dimension. In one dimension length is the same as 'mass'. In two dimensions and above, a cluster of size/mass s can have several different shapes, while in one dimension it is always a straight line. The correlation length gives us information about the extent of the system, which is important since the ratio ξ/L determines wheter the finite system affects the result.

When the correlation length is a lot smaller than the system size, $\xi \ll L$, the effects of having a finite system instead of an infinite is not noticeable. The reason is that we don't have any clusters that are large enough to "notice" that the system is finite. However, when the system size is a lot smaller than the correlation length, $\xi \gg L$ the systems behavior is dominated by the the size of the system. When $\xi \simeq L$, it is difficult to determine how close we are to p_c, but we can estimate given

$$L \simeq \xi \propto |p_c - p|^{-\nu} \Rightarrow |p_c - p| \propto L^{-1/\nu} \tag{13.17}$$

The characteristic cluster size in dimensions d is related to the correlation length by

$$s_\xi \propto \xi^D \tag{13.18}$$

where D is a fractal dimensionality constant $D < d$.

To quantify the geometry of clusters, we could look at the maximum distance between any two points in the cluster, or even the average distance between two points. However, we wish to look at the variance of the sites of the cluster to the centre of mass of the cluster. It is similar to the standard deviation We call this property the *radius of gyration*. For a given cluster i, with size s_i, we can find the radius of gyration by

$$R_i^2 = \frac{1}{s_i} \sum_{j=1}^{s_i} (\mathbf{r}_j - \mathbf{R}_i)^2 \tag{13.19}$$

R_i is the radius of gyration and j is just a label for a given site in the cluster. The cluster has a centre of mass $\mathbf{R}_i$ and for a site j, it's position in the cluster is $\mathbf{r}_j$.

The result is that a dense cluster of size s will have a smaller radius of gyration than a sparser cluster with the same size, and

$$s \propto R_s^D \tag{13.20}$$

14. Cluster Size in 1D Percolation

Introduce the characteristic cluster size for the 1D percolation problem, and discuss their behavior when $p \to p_c$. Relate to your simulations on two-dimensional percolation.

14.1 Definition

A 1D system will only percolate if all sites are occupied, like water through a pipe will only if flow if there are no obstacles.

Imagine a cluster of size s in a 1D pipe. A specific site in the cluster could be the first site from the left. The probability for a site to belong to a specific site in a cluster of size s is called the *cluster number density*, given by $n(s, p)$.

The probability of a site being occupied is p. For all other sites in the cluster to be occupied, we have the probability p^{s-1}. Now we also need to make sure that these points are actually grouped together. The sites on each side of the cluster has a probability of $1 - p$ of being unoccupied. Thus the cluster number density is

$$n(s, p) = (1 - p)^2 p^s \tag{14.21}$$

These probabilities need to be summed the number of sites in the cluster, namely s. Thus, if we point at a specific site in a cluster in a lattice, the probability of that site to be part of a cluster of size s is

$$P(\text{site is part of cluster of size s}) = s \cdot n(s, p) \tag{14.22}$$

With a little trick, we rewrite

$$p^s = e^{s \ln p} = e^{-\,{}^{s}/_{s_\xi}} \tag{14.23}$$

where $s_\xi = -\,{}^{1}/_{\ln p}$ is the characteristic cluster size. It is the value of the largest cluster we can expect to find. $n(s, p)$ is almost constant when $s < s_\xi$ but dies out when $s > s_\xi$. The probability $s \cdot n(s, p)$ still dies out, since s is linear and $n(s, p)$ vanishes exponentially when $s > s_\xi$.

14.2 Convergence Behavior for $p \to p_c$

The characteristic cluster size s_ξ goes to infinity as p approaches 1. $p = p_c = 1$ is the percolation threshold for 1D percolation, and it is at this moment our system starts percolating. When $p \to p_c$, the clusters become bigger and bigger, but for an infinite system, it won't percolate until $p = p_c$.

14.3 Comparions with Two-dimensional Percolation

The characteristic cluster size s_ξ is the number of sites in a cluster. In 1D this is the effectively length of the cluster. When we proceed to two dimensions, it is instead meaningful to separate the length from the size (or the mass if you want). In one dimension the correlation length was found to diverge as a power law. We can now write the characteristic cluster size on general form

$$s_\xi \propto |p - p_c|^{-\,{}^{1}/_{\sigma}} \tag{14.24}$$

when $p \to p_c = 1$. In the one-dimensional case we have $\sigma = 1$, and for larger dimensions, $\sigma < 1$.

This means that the characteristic cluster size diverges faster in two dimensions than one. As for the one-dimensional case, we won't see any percolation until $p \geq p_c$. Before this, s_ξ approached infinity, but after the threshold isreached, it goes back down. This is due to the spanning cluster "eating up" the other large finite clusters as p increases past the threshold. When $p \to 1$, the characteristic size approaches zero. This is because s_ξ describes finite clusters, and the probability of finding a finite cluster is lower and lower. Here the power law doesn't hold, as it is only valid when $|p - p_c|$ is small.

14.4 Cluster Geometry

The correlation length ξ is different from the characteristic cluster size s_ξ, however this is only the case when we are beyond one dimension. In one dimension length is the same as 'mass'. In two dimensions and above, a cluster of size/mass s can have several different shapes, while in one dimension it is always a straight line. The correlation length gives us information about the extent of the system, which is important since the ratio ξ/L determines wheter the finite system affects the result.

When the correlation length is a lot smaller than the system size, $\xi \ll L$, the effects of having a finite system instead of an infinite is not noticeable. The reason is that we don't have any clusters that are large enough to "notice" that the system is finite. However, when the system size is a lot smaller than the correlation length, $\xi \gg L$ the systems behavior is dominated by the the size of the system. When $\xi \simeq L$, it is difficult to determine how close we are to p_c, but we can estimate given

$$L \simeq \xi \propto |p_c - p|^{-\nu} \Rightarrow |p_c - p| \propto L^{-1/\nu} \tag{14.25}$$

The characteristic cluster size in dimensions d is related to the correlation length by

$$s_\xi \propto \xi^D \tag{14.26}$$

where D is a fractal dimensionality constant $D < d$.

To quantify the geometry of clusters, we could look at the maximum distance between any two points in the cluster, or even the average distance between two points. However, we wish to look at the variance of the sites of the cluster to the centre of mass of the cluster. It is similar to the standard deviation We call this property the *radius of gyration*. For a given cluster i, with size s_i, we can find the radius of gyration by

$$R_i^2 = \frac{1}{s_i} \sum_{j=1}^{s_i} (\mathbf{r}_j - \mathbf{R}_i)^2 \tag{14.27}$$

R_i is the radius of gyration and j is just a label for a given site in the cluster. The cluster has a centre of mass $\mathbf{R}_i$ and for a site j, it's position in the cluster is $\mathbf{r}_j$.

The result is that a dense cluster of size s will have a smaller radius of gyration than a sparser cluster with the same size, and

$$s \propto R_s^D \tag{14.28}$$

14.5 Measurements

Given a finite system, we can count the number of clusters of size s as N_s. The size of the system is L, so in a simulation we can measure the *estimated* cluster number density $\bar{n}$ as[9]

$$\overline{n(s,p)} = \frac{N_s}{L} \tag{14.29}$$

We can improve the estimate by generating M different systems and estimate the average

$$\overline{n(s,p)} = \frac{N_s}{LM} \tag{14.30}$$

Large systems yield very different s-values, but we can bin them and create a histogram. We should however create logarithmic bin sizes, so the bin-width increases as s increases. Else we would get a lot of empty bins. It is now very important to divide every bin with the correct bin size.

[9] In d dimensions, we have $\overline{n(s,p)} = N_s/L^d$

15. Measurements and Behavior of $P(p, L)$ and $\Pi(p, L)$

Discuss the behavior of $P(p, L)$ and $\Pi(p, L)$ in a system with a finite system size L. How do you measure these quantities?

15.1 Percolation Probability Π and Cluster Spanning Density P

$\Pi(p, L)$ is the percolation probability: the probability that a randomly generated system of size L has atleast one spanning cluster. The spanning cluster density $P(p, L)$ is the probability a given site has of being a part of the spanning cluster. If we divide the average mass of a spanning cluster by the number of sites in the system, we get the spanning cluster density. These properties are different for systems in one and two dimensions.

15.2 Behavior of Π and P

For an infinite system, $L = \infty$, there is a (dimension-specific) percolation threshold p_c where the system *always* starts to percolate. For a one-dimensional system, which is basically a one dimensional pipe, this threshold is of course when $p_c = 1$. In other words: all sites must be occupied, and they are occupied with probability p. In two dimensions this threshold is $p_c = 0.59275$. The percolation probability $\Pi(p, L)$ is then a step function where p_c is the point where the probability of percolation goes from 0 to 1. In the smallest system, $L = 1$, we have linear percolation probabilites—that is $\Pi = p$ for both one and two dimensions.

For a small but finite L, we are able to find analytical (closed-form) solutions for the percolation probability. In one dimension, it is as easy as

$$\Pi(p, L) = p^L \tag{15.31}$$

since every site is occupied with probability p and all L sites must be occupied.

In two dimensions, we can't find a general expression for all L, but we can find it manually by listing all the configurations as long as L is small. The system size grows exponentially as $2^{L \times L}$.

15.3 Configuration of 2×2 Matrix

A 2×2 lattice can have 1, 2, 3 or 4 occupied states. We can put the configurations into six groups like this:

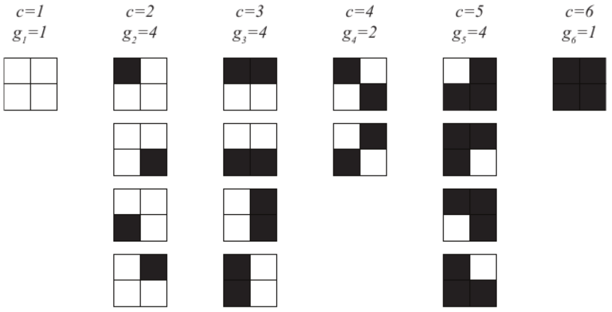


Fig. 1.8 The possible configurations for a $L = 2$ site percolation lattice in two-dimensions. The configurations are indexed using the cluster configuration number c.

If p is the probability of an occupied state, then for example $c = 0$ has the probability $P(c) = (1 - p)^4$. This is because we multiply the probability of there not being a occupied state together four times. The probabilities for all groups are given as:

c	g_c	$P(c)$	$\Pi(p,L\|c)$
1	1	$p^0(1-p)^4$	0
2	4	$p^1(1-p)^3$	0
3	4	$p^2(1-p)^2$	1
4	2	$p^2(1-p)^2$	0
5	4	$p^3(1-p)^1$	1
6	1	$p^4(1-p)^0$	1

Table 2:

and the general expression is $p^x q^{4-x}$ where x is the number of occupied sites in the system.

The probability for a spanning cluster is[10]

$$\Pi(p,L) = \sum_c \Pi(p,L|c)P(c) \tag{15.32}$$

and the probability $P(c)$ of spanning, given a configuration c, is either 0 or 1, as we saw in table 2. This leads us to the general equations

$$\Pi(p,L) = 4p^2q^2 + 4p^3q + p^4 \tag{15.33}$$

$$P(p,L) = \frac{2}{4}\cdot 4p^2q^2 + \frac{3}{4}\cdot 4p^3q + \frac{4}{4}\cdot p^4 \tag{15.34}$$

For the spanning cluster density, we must multiply each term of the percolation probability by the size of the cluster for the given configuration. The spanning cluster density is generally $P = {}^{M_i}/_{L^2}$ where M_i is the 'mass' of the cluster.

Figure 11.2 shows us how Π and P behaves for $L = 2$. For the next few L-values, the curves are the same only (gradually) more distinctive, the Π curve is more S-shaped and the P curve is more curved.

Both curves increase from $p = 0$ to $p = 1$ and are smooth. P is however always below $y = p$. This is due to the fact that the spanning cluster density is the probability that a random site is part of the spanning cluster. The random site is not always occupied, so we must have $p \geq P$. As the $p \to 1$, the difference between the two vanishes. Of course, for $L = 2$, if there is a spanning cluster, all occupied sites are part of the cluster, but in larger systems we can have non-spanning and spanning clusters at the same time.

15.4 Numerical Solutions of Π and P

The size of a two-dimensional percolation system is $2^{L\times L}$. For larger systems, we can't manually find the the analytical solutions of Π and P. However, we can find them using Monte-Carlo methods. We generate N matrices of size $L \times L$ and check if they percolate or not. n is the number of matrices that do percolate. Each system has a mass of M_i, and thus we estimate Π and P as

$$\Pi(p,L) = \frac{n}{N}, \quad P(p,L) = \frac{\sum_i M_i}{NL^2} \tag{15.35}$$

[10] From probability theory we have $P(A) = \sum_B P(A|B)P(B)$, which implies equation 15.32.

16. The Cluster Number Density

Introduce the cluster number density and its applications. Definition, measurement, scaling and data-collapse.

16.1 Definition

A two dimensional percolation system of size $L \times L$ is represented as a matrix. Each element of the matrix is called a site. The site can be occupied with probability p. A set of sites that are connected to each (the nearest neighbor is mostly used) is called a cluster. For a big cluster, p will be the porosity. A spanning cluster is a cluster that extends from one side to another. In two dimensions there can none, one or serveral spanning clusters. A cluster is part of the infinite spanning cluster with probability $P(p)$. The *cluster number density* is the probability $n(s,p)$ for finite clusters that a site is a *particular* site in a cluster of size s. The normalization is given by

$$\sum_s s \cdot n(s,p) + P(p) = p \tag{16.36}$$

Here $s \cdot n(s,p)$ is the probability that the site is part of the cluster with size s.

16.2 Measurements

To do measurements on the system, we must first create the $L \times L$ matrix. Furthermore, we label each site in a cluster i and find their size. Without including any spanning clusters, we find how many clusters are of size s, N_s. We can then estimate the cluster number density as

$$\overline{n(s,p)} = \frac{N_s}{L^d} \tag{16.37}$$

where d is the dimensionality, and the results is thus valid for dimensions higher than one.

To improve the result, we should generate M such systems, count the N_s for all systems, and divide by M,

$$\overline{n(s,p)} = \frac{N_s}{L^d M} \tag{16.38}$$

Since there are a lot of different cluster sizes s in a large system, we should create bins with some width. Then we could create a histogram showing the probability density $n(s,p)$. If we create bins width uniform sizes, we will be at loss. The reason is that the distribution has many low values for s, but not so many for larger values. Therefore we do logarithmic binning, which means that the bin size increase as s increases. The large s-values are still more probable than they would be in an expoential distribution[11]. Here it is crucial that we divide by the *correct* bin size for each bin.

16.3 Scaling

The cluster number density in one dimension can be written analytically as

$$n(s,p) = (1-p)^2 p^s = (1-p)^2 e^{s \ln p} = (1-p)^2 e^{-\,{}^{s}/_{s_\xi}} \tag{16.39}$$

where we defined the characteristic cluster size as $s_\xi = -\,{}^{1}/_{\ln p}$

For a constant p, the first term is a normalization factor, and the second factor, p^s describes the scaling As long as $s \ll s_\xi$, the graph will be relatively flat and stable, but when $s > s_\xi$, it falls exponentially.

[11] Black Swan behavior of power laws

Using the Taylor expansion of $\ln|1-x|$, with $x = 1-p$, we can rewrite the characteristic cluster size,

$$s_\xi = -\frac{1}{\ln p} \simeq \frac{1}{1-p} = |p-p_c|^{1/\sigma} \tag{16.40}$$

This holds for all dimensions, and $\sigma = 1$ for one dimensional percolation. The characteristic cluster size diverges as a power law when $p \to p_c$. This indicates that the behavior of $n(s,p)$ when $p \neq p_c$ is only similar to that of $p = p_c$ in the special case of $s \ll s_\xi$. We describe this by introducing a cut-off function

$$n(s,p) = s^{-2}\left(\frac{s^2}{s_\xi^2}e^{-\,s/s_\xi}\right) = s^{-2}F\left(\frac{s}{s_\xi}\right) \tag{16.41}$$

The dependence on s_ξ is included in the cut-off function F, and we get

$$n(s,p) = n(s,p_c)F\left(\frac{s}{s_\xi}\right) \tag{16.42}$$

Which again is a general percolation results. We have

$$n(s,p_c) = Cs^{-\tau} \quad \text{and} \quad F\left(\frac{s}{s_\xi}\right) \to 0 \text{ for } s \gg s_\xi \tag{16.43}$$

16.4 Data-collapse

The dependence on p is included in the $F(s/s_\xi)$ through the characteristic cluster size. The cut-off function is the same for all p if we plot it as a function of s/s_ξ. This is the phenomenon known as *data-collapse*. We can see this more clearly if we first plot $n(s,p)$ as a function of s, then as a function of s/s_ξ. The first plot shows a set of graphs that cut off at different values of p. In the other, all the graphs coincide. There could of course be some small fluctuations due to noise, but if there are systematic deviations, there must be something wrong with the simulation.

17. Finite-size Scaling of $\Pi(p, L)$

Discuss the behavior of $\Pi(p, L)$ in a system with a finite system size L. How can we use this to find the scaling exponent μ, and the percolation threshold p_c?

To extract results from our numerical solutions, we must bridge the gap beween finite and infinite systems—the thermodynamical limit. Therefore we must be able to use the information from our finite system to make predictions about the infinite system. One way to achieve this is *finite-size scaling*, where we survey how the results scale with the system size. Then we use this information to predict the infinite system.

The correlation length ξ is a property that act as the expected maximum extent any cluster can have. That is, in some system with length L, we expect to find clusters of sizes up to this length, but not many larger ones. When $p \to p_c$, ξ diverges as a power law,

$$\xi \propto |p - p_c|^{-\nu} \tag{17.44}$$

The finite size of the system is not noticeable when the correlation length is much smaller than the system size, $\xi \ll L$. However, when $\xi \gg L$, the clusters in our system are supposed to be much larger. This will make the results very different.

17.1 The Percolation Probability

In the thermodynamic, when the system size goes to infinity, we know that the percolation probability is a step function at $p = p_c$. For small L-values, for example $L = 1$, it will be linear with p.

If we consider the affect of the finite size of the system, it is obvious that

$$\Pi(p, L) = \begin{cases} 1, & \text{if } \xi \gg L \\ 0, & \text{if } \xi \ll L \end{cases} \tag{17.45}$$

It should be obvious that the function scales as L/ξ. In the simple one-dimensional case it is possible to find the closed-form solution,

$$\Pi(p, L) = f\left(\frac{L}{\xi}\right) = e^{-L/\xi} \tag{17.46}$$

It is not possible find the analytic solution in two dimensions, but we just need the fact that there is a scaling relation of the ratio L/ξ. We find that

$$\Pi p, L = f\left(\frac{L}{(p - p_c)^{-\nu}}\right) = \Phi((p - p_c)L^{1/\nu}) \tag{17.47}$$

This is a relation of how the percolation probability should scale with p and L. If want to make something useful out of this, let us rather ask: What value must p take to be for us to obtain a wanted percolation probability, e.g. $\Pi(p, L) = 50\%$,

$$\Pi(p_{\pi=x}(L), L) = x \tag{17.48}$$

and using the relation above,

$$(p_{\pi=x} - p_c)L^{1/\nu} = \Phi^{-1}(x) \tag{17.49}$$

The form of Φ is unknown, but it is a smooth function, ranging from 1 to 0. Therefore an inverse function should exist. If we let it be a constant, C_x, we get

$$p_{\pi=x}(L) = p_c + C_x L^{-1/\nu} \tag{17.50}$$

To find the unknown terms in this equation, let us first look at difference between two p-values, Δp, and taking the logarithm,

$$\log \Delta p(L) = \log(C_{x_2} - C_{x_2}) - \frac{1}{\nu} \log L \tag{17.51}$$

We can then measure $\Delta p(L)$ numerically for different L-values and plot the log-log plot. The results is a straight line with slope $-{}^1\!/\!_\nu$. Furthermore we plot $p_{\pi=x}$ as a function of of $L^{-{}^1\!/\!_\nu}$, also a straight line,

$$p_{\pi=x}(L^{-{}^1\!/\!_\nu}) = p_c + C_x L^{-{}^1\!/\!_\nu} \tag{17.52}$$

The slope is now a function of C_x, but p_c is given by the intersection between the y-axis and p. We can finally find p_c by linear regression.

17.2 Summary

By generating a passel of systems for finite L, we can count the number of spanning clusters and easily find $\Pi(p, L)$ numerically. But we want to investigate the system in the thermodynamic limit $L \to \infty$, and it is not trivial to go from finite to infinite system sizes. Therefore we look at how $\Pi(p, L)$ scale as a function of ${}^\xi\!/\!_L$, and found a finite-size scaling-function that allowed us to predict the the percolation threshold p_c and the scaling parameter ν. Both properties of the thermodynamic limit.

18. Subsets of the Spanning Cluster

Introduce and discuss the scaling of subsets of the spanning cluster. How can we measure the singly-connected bonds, and how does it scale?

18.1 The Subsets

In a percolating system, there is at least one spanning cluster. To investigate the spanning cluster, we divide into parts called subsets. There are three of them: *the single-connected bonds*, *the backbone* and the *the danlging ends*.

If we assume that flow is running through the cluster, we can explain the subsets in the following way.

The Single-connected Bonds (SCB) These are the ‘bottlenecks’ of the system—the sites which are vital for the cluster to be spanning. That is, if removed, the cluster would seize to percolate. Consequently all flow go through these sites.

The Backbone (BB) All sites that the fluid is transported through is part of the backbone. In nanoporous materials this is called the *transport-available porosity*. A subset of the backbone is the single-connected bonds since these are part of the transport system.

The Dangling Ends (DE) The parts of the system that do not contribute to the percolation. Any ‘dead end’ which are connected through only one site is a dangling end. If the site is removed, the dangling end is cut off from the cluster. When we have stationary flow, flow can go into or out of the dangling end.

18.2 Scaling

The mass M of the spanning cluster follows a power law with respect to the size of the system.

$$M \propto L^D, \text{ where } D \leq d \tag{18.53}$$

D is the fractal dimensions of the spanning cluster. It should be a power of p. When $p \to 1$, we expect $D \to d$. The reason is that the spanning cluster would grow as fast as the system. For lower p-values, we expect $D < d$. This is because a fraction of all the sites added when the system size increases will be unoccupied sites, or occupied sites in clusters that are not spanning.

The mass M_i of the subsets all independently follow a power law, with different fractal dimension coefficients D_i,

$$M_i \propto L^{D_i}, \text{ where } i \in \{\text{SCD, BB, DE}\} \tag{18.54}$$

There are some obvious restrictions to these fractal dimension coefficients,

$$M_i \leq M \Rightarrow D_i \leq D \tag{18.55}$$

A *self-avoiding walk (SAW)* is basically a path through a lattice with no self-crossing—each point on the lattice is only visited once. If we have a system with the subsets discussed above, and let a large number of SAWs walk from one side of the spanning cluster to the other, there can be many different paths. Since the dangling ends are only connected through one site, a SAW entering one will be stuck. Looking at the SAWs that make it across, everyone will have visited the single-connected bonds. Thus we can extract the SCBs. From the SAWs that make it, we can also extract the backbone, since it is the union of all the paths.

The length of the paths the SAWs take can be different, so we denote the scaling of the maximum and minimum paths as $D_{\max}$ and $D_{\min}$, and we must have the following relations

$$D_{\text{SCB}} \leq D_{\min} \leq D_{\max} \leq D_{\text{BB}} \leq D \leq d \tag{18.56}$$

The results for the percolation threshold p_c in two dimensions are tabulated in table 3.

D_{SC}	$D_{\min}$	$D_{\max}$	D_{BB}	D	d
0.75	1.1	1.5	1.6	1.89	2

Table 3:

The dangling ends remain, and we know that their fraction must be $D_{\text{DE}} \leq D$.

The total mass of the spanning cluster is

$$M_{\text{SP}} = M_{\text{BB}} + M_{\text{DE}} \tag{18.57}$$

since the backbone contains the single-connected bonds, and the dangling ends are the rest of the system. The scaling of the system can then be written

$$aL^D = bL^{D_{\text{BB}}} + cL^{D_{\text{DE}}} \tag{18.58}$$

18.3 Measurements

Let us assume we have a two-dimensional spanning cluster. We can deploy just two walkers to find the single-connected bonds. We let walker A be *left-turning* and B be *right-turning*, that is A tries to go left at every opportunity, while B tries to go right. If the A can't go left, it tries forward, then right, and as a last solution backward. Similarly for B.

The intersection of the sites A and B visit are the single-connected bonds. It is important to distinguish these walkers from the SAWs, as the walkers will go back through where they came from if they are in a dangling end.

Doing this we find that the scaling of the SCB is $^1/_\nu = {}^3/_4$ (in two dimensions). In project 3 we did this with good results.

It is also possible to use renormalization theory to make theoretical arguments about the critical exponents and therefore the scaling. The reason is that the spanning cluster is a self-fractal.

19. Flow in a Disordered System

How do you measure the conductivity of the spanning cluster? Discuss the scaling theory for the conductivity $\sigma(p, L)$ when $p > p_c$. Relate the results to permeability in a nanoporous system.

19.1 Conductivity of the Spanning Cluster

Conductance G is the inverse of resistance R in a circuit component. It measures how willing the component is to let flow/current (I) pass through it. From Ohm's law

$$I = \frac{V}{R} = GV \tag{19.59}$$

A similar property to conductance is *conductivity*. While the conductance is a property of the component, conductivity is a property of the material. The element Copper thus has conductivity, but a Coppwer wire has conductance. In a two dimensional quadratic system these properties are equal,

$$G = \sigma\frac{A}{L} = \sigma\frac{L^{d-1}}{L} = \sigma \tag{19.60}$$

Since current is just flow of electricity, we use this for flowing fluids, which is described by Darcy's law,

$$\Phi = \frac{k}{\mu}\frac{A}{L}\Delta P \tag{19.61}$$

where we would have $\sigma = {}^{k}\!/\!_{\mu}$ and the pressure differential ΔP is similar to voltage V. Here k is the permeability and μ is the viscosity of the fluid.

Applying a pressure differential to the system, then measuring the flow through it, gives us conductivity/conductance (?). The current-laws of Kirchhoff lets us find the flow through the system. From the first law we know that the total flow into and out of a site must be zero. Otherwise we would build pressure inside the site, which can't be true for stationary flow. For a site i we find the sum of the flow from the neighbors,

$$\Phi_i = \sum_j g_{i,j}(p_j - p_i) \tag{19.62}$$

The flow flux Φ_i is only non-zero on the system bounds. The conductance between sites follow the Kronecker delta δ_{ij}, and it is 1 between connected sites and 0 elsewise. The flow between sites is proportional to the pressure difference between them. These flow equations for all sites in a spanning cluster is a set of linear differential equations, which we can write as the matrix equation

$$Ap = c \tag{19.63}$$

where p is the unknown pressure vector for all the sites, and c is the vector containing boundary information, as the flow elsewhere is zero. Our matrix A describes the spanning cluster. It can be set up in a clever way such that it becomes a sparse matrix. Solving the equation for the pressure p lets us set up a map of the flow through the system. From this map we can see which sites are part of the backbone, dangling ends and which are single-connected bonds. The conductance can be calculated as

$$G = \sigma = \frac{\Phi}{\Delta P} \tag{19.64}$$

19.2 Scaling Theory of Conductivity

We can assume that the conductance should increase with the pressure p, since an increasing volume becomes transport-available.

We could believe that σ would grow as the density of the cluster P, but it turns out that it grows faster. This means that the transport-available porosity grows faster than the porosity itself. The reason is that as $p \to p_c$ all the dangling ends become transport-available. The relations are

$$P \propto (p - p_c)^{\beta}, \ \beta < 1 \quad \text{and} \quad \sigma \propto (p - p_c)^{\mu}, \ \mu > 1 \tag{19.65}$$

Let us now look at the size dependence and the conductance as a function of correlation length and system size, $G(\xi, L)$. When $\xi \ll \xi$ our system behaves like it does when $p = p_c$. In the case where $\xi \gg \xi$, we start by dividing the system into subsystems of size ξ^d. The reason is that all the subsystems should look identical. Now every subsystem appears to behave like $p = p_c$, thus every subsystem will have the conductivity $G(\xi, \xi)$. The total conductance is

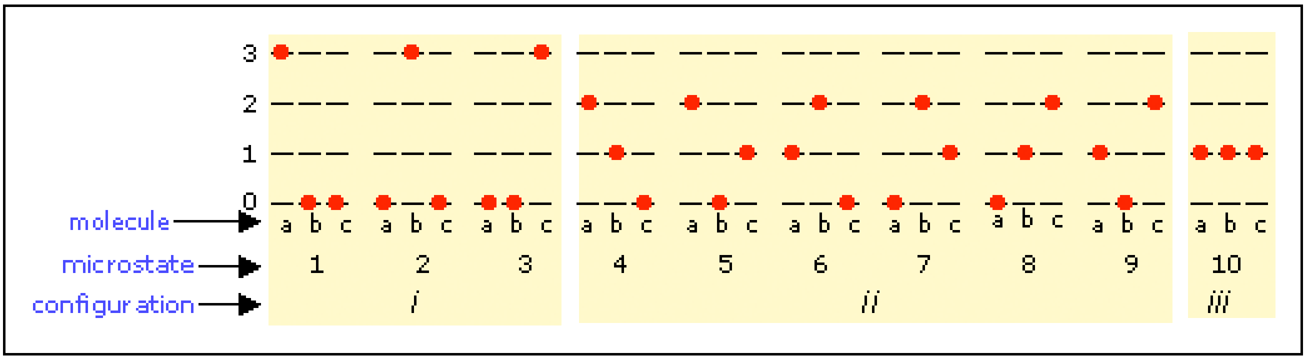
$$G(\xi, L) = \sigma \frac{A}{L} = G(\xi, \xi) \left(\frac{L}{\xi} \right)^{d-2} \tag{19.66}$$

This means that for the two-dimensional case, $G(\xi, L)$ remains constant while L increases much more than ξ. In the three-dimensional case, the conductance increases with increasing L indefinitely. The reason is that the area increases quadratically and the length increases linearly.

D. BASIC CONCEPTS

A What is a Microstate?

Imagine we have three ‘points’ we can distribute freely among three molecules a, b and c. Let us give 3 points to a, 0 to b and 0 to c. Now we have a *microstate*! So there are three microstates where *one* molecule gets 3 points, and the others get 0. These three microstates make up the *configuration* i. We could also give 2 points to one of the molecules and 1 to another. The set of all these microstates would be configuration ii. The last microstate possible is where each molecule has 1 point each. This is the final configuration iii, containing only one microstate. See figure below.



The figure above is an *ensemble*. It holds consists of tem microstates in three configurations. An ensemble is a set of microstates under some conditions. The conditions above is that we are only allowed to allocate three points to the molecules.

B Degrees of Freedom

A particle in three dimensions has the three spatial coordinates x, y and z. These are crucial in describing the system. A *degree of freedom* is an independent parameter needed to describe a system. Thus the three position coordinates, and also the three velocities, are degrees of freedom for a system of one particle. A set of independent degrees of freedom X_i holds $E = \sum_i E_i(X_i)$. Where the energy function E_i is a function of only one variable.

A *quadratic degree of freedom* is a degree of freedom that contributes quadratically to the energy. An example of a system of two degrees of freedom is $E = X_1^2 + X_1X_2 - X_2^2$.

A system of both quadratic and independent degrees of freedom must hold $E = \sum_i \alpha_i X_i^2$.