

Exam Questions - FYS4460

Emilie Fjørner

May 22, 2015

Molecular Dynamics

1. Molecular-dynamics algorithms

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

Potentials: I enhver molekylærdynamisk simulering har man behov for et potensiale som beskriver interaksjonene mellom molekylene man modellerer. Man kan da velge mellom en lang rekke potensialer med varierende grad av nøyaktighet når det gjelder fysikken de beskriver. Måten man beskriver det fysiske systemet på innen klassisk molekylærdynamikk krever at man gjør noen forenklinger fra den fulle kvantemekaniske beskrivelsen. Den første av disse forenklingene beskrives av den såkalte **Born-Oppenheimer** approksimasjonen. I denne tilnærmingen antar man at Hamiltonfunksjonen for et molekyl i systemet kan deles opp i to separate deler. Den ene beskriver da elektronene, og den andre nukleidene. Gjennom å løse den tidsuavhengige Schrödingerligningen kan man da finne en bølgeligning som beskriver elektronene, og en som beskriver resten av molekylet, som sammensatt gir den totale bølgeligningen for systemet. Bakgrunnen for denne tilnærmingen er at elektronenes vekselvirkninger skjer så raskt at man kan se på det som om de reagerer instantant til nukleidenes bevegelse. Den andre forenklingen går ut på at man kan behandle nukleidene som punktpartikler og beskrive dem ved hjelp av klassisk Newtonsk dynamikk. Dersom det er behov for finere detaljer kan man benytte seg av såkalte kvantemekaniske potensialer der kvantemekaniske effekter er inkludert i beskrivelsen av systemets interaksjoner.

Det enkleste “pair-potential” er **Lennard-Jones** potensialet. Et par-potensial er et potensial som kun tar for seg interaksjonen/samspeillet mellom to og to partikler. Lennard-Jones potensialet er gitt som

$$U(r) = 4\epsilon \left[\left(\frac{\sigma}{r} \right)^{12} - \left(\frac{\sigma}{r} \right)^6 \right]. \quad (1)$$

Her er r avstanden mellom partiklene/atomene, ϵ beskriver dybden på potensialbrønnen og σ angir den fineste avstanden hvor potensialet er null. Dette potensialet beskriver van der Waals vekselvirkningene mellom atomer og er derfor en god beskrivelse av systemer der disse dominerer over andre vekselvirkninger, slik som de som har å gjøre med kjemiske bindinger, som for eksempel for en edelgass. L-J er sterkt avstøtende for korte avstander (Pauli repulsion) og svakt tiltrekkende ved lengere tilstander (van der Waals attraction) (skissér). Kreftene mellom partiklene i systemet kan finnes ved derivasjon $F = -\frac{\partial U}{\partial r}$.

Ofte konstrueres potensialer spesifikt for systemet man ser på. **Weber-Stillinger** potensialet beskriver for eksempel silisium. I likevekt vil silisium forme en tetrahedral struktur med sterke bindinger. Her kan vi ikke bruke Lennard-Jones eller andre sfæriske symmetriske potensialer. W-S potensialet består av en two-body og en three-body del, hvor two-body delen likner på Lennard-Jones potensialet med en smooth cut-off. Three-body delen beskriver bending stiffness i systemet.

VKRE (Vashita, Kalia, Rino og Ebbsjö) potensialet er laget spesielt for SiO_2 og inneholder en two-body del som består av tre termer (Coulomb vekselvirkninger, sterisk frastøtning og charge-dipole interaction). reaxFF er et potensiale som også inkluderer fire-partikkel interaksjoner.

Integration: For å studere hvordan systemet utvikler seg i tid brukes integrasjonsalgoritmer. Disse er basert på finite difference metoder hvor tiden er en diskret variabel. I enhver slik algoritme vil det være diverse feilkilder, slik som trunkeringsfeil, i tillegg til avrundingsfeil grunnet maskinimplementasjon. I våre studier av molekylære systemer har vi brukt Verlet algoritmen til å integrere atomenes bevegelser. Denne algoritmen baserer seg på å regne ut hastigheten et halvt tidssteg fram i tid, og bruker så denne hastigheten til å regne ut

den nye posisjonen. Deretter regnes det ut nye krefter med de nye posisjonene. Denne nye kraften brukes for å regne ut den nye hastigheten ved neste tidsteg (med utgangspunkt i hastigheten ved vi fant et halvt tidsteg fram i tid).

$$v_i(t + \Delta t/2) = v_i(t) + \frac{F_i(t)}{2m} \Delta t \quad (2)$$

$$r_i(t + \Delta t) = r_i(t) + v_i(t + \Delta t/2) \Delta t \quad (3)$$

$$F_i(t + \Delta t) = -\nabla_i U_i(r(t + \Delta t)) \quad (4)$$

$$v_i(t + \Delta t) = v_i(t + \Delta t/2) + \frac{F_i(t + \Delta t)}{2m} \Delta t \quad (5)$$

Verlet algoritmen er forholdsvis enkel men likevel er den numerisk stabil og gir meget god energibevaring.

Periodic boundary conditions: Når en partikkel beveger seg gjennom veggen av systemet, dukker den opp på andre siden med samme hastighet. Kan tenke på det som uendelig mange kopier av systemet ved siden av hverandre. Når man skal regne ut kraften mellom to partikler, så er det uendelig mange versjoner. Velger den minste avstanden: minimal image convention.

Initialisering: Vi har studert et system bestående av argonatomer, en edelgass som kan beskrives på en god måte av L-J potensialet. Krystallstrukturen til Argon kan beskrives som bestående av atomer plassert i en struktur tilsvarende en face centered cubic lattice/gitter. Vi valgte å starte simuleringen vår fra denne strukturen. Med andre ord initierte vi simuleringen fra en tilstand der alle atomene var plassert i en gitterstruktur. Atomene i strukturen ble gitt en tilfeldig hastighet trukket fra enten en Boltzmann distribusjon, eller en uniform distribusjon (trekk fra ev drift, gjør totalt momentum lik null). Hastighetene vil uansett utvikle seg til å følge en Maxwell-Boltzmann distribusjon. Fra denne tilstanden lot vi systemet utvikle seg til det var termalisert. For å få et system med ønsket temperatur kan man da bruke en termostat. Hvis man vil studere en spesifikk tilstand, f.eks argongass ved en spesifikk temperatur er det lurt å lagre denne staten etter at man har nådd den, slik at man slipper å bruke tid på å simulere seg fra krystallstrukturen hver gang man vil kjøre en simulering. Når man setter opp systemet bør man også tenke på hvilken fase det er man vil studere, ettersom denne avhenger av partikkeltettheten til systemet. Så dersom man vil studere f.eks. en gass, der partikkeltettheten er lav, kan man sette opp systemet slik at det er tilsvarende god plass mellom partiklene.

Efficiency improvements and cut-off: L-J potensialet vi har brukt regner parvis ut kreftene mellom to og to atomer i systemet. Dette er en svært tidkrevende prosess og vi vil gjerne effektivisere simuleringen. En enkel forbedring er å bruke Newtons 3. lov (kraft=motkraft) da halverer man med en gang antallet krefter man må regne ut. En annen nyttig effektivisering oppnås ved å se nærmere på L-J potensialet. Dette faller raskt til null, slik at etter en avstand $r_{cut-off}$ vil kreftene mellom to partikler være så godt som null i dette potensialet. Fra dette ser vi at vi kan introdusere en **cut-off** avstand ($\approx 3\sigma$). For avstander større en dette trenger vi ikke regne på kreftene mellom atomer (men det blir et skift i potensiale og kraft). Dette kan oppnås ved å implementere såkalte cell lists. Man deler systemet inn i celler på størrelse med r_{cut} , og lar kun partikler i naboceller vekselvirke. Totalt må man da for en partikkel regne på kreftene mellom denne og partikler i totalt 27 celler (cellen den er i pluss 26 naboceller). Dette gir en markant forbedring i tidsbruken til simulering. Ønsker man en enda raskere simulering kan dette løses ved å parallelisere programmet. Når systemet er inndelt i celler er det svært paralleliserbart.

2. Molecular-dynamics in the micro-canonical ensemble

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

Initialisering: Se oppgave 1.

Initialization effects: Vi har i prosjekt en og to studert systemer med Argonatomer. Disse danner en krystallstruktur bygget opp av face centerede kuber, en såkalt face centered cubic lattice, og vi valgte å starte simuleringene våre fra denne tilstanden. Det vil si at vi valgte initielle posisjoner slik at atomene formet en slik gitterstruktur. Dette er en mulighet for hvordan man kan sette opp systemet sitt når man vil kjøre en md simulering. Vi kunne også valgt å plassere ut atomene basert på andre regler, men dette kan føre til at partiklene får en uheldig konfigurasjon. Hvis man f.eks plasserer atomene tilfeldig ut i systemet havner kanskje noen for nærme hverandre. Dette gir voldsomt store krefter (og mye potensiell energi \rightarrow systemet eksploderer) og det er defor noe man vil unngå. Som en del av initialiseringen vil man også gi partiklene en starthastighet. Hvordan man velger starthastighetene vil ikke ha så mye å si, for hastighetsfordelingen vil utvikle seg til en Maxwell-Boltzmann distribusjon etterhvert som systemet går mot en likevektstilstand. Likevel vil det være lurt å velge en hastighetsdistribusjon nær den endelige likevektsdistribusjonen siden det da vil gå raskere å nå denne likevektstilstanden. Man bør også passe på at man ikke gir hele systemet en total hastighet (drift), gjennomsnittshastigheten må være null. I begynnelsen av simuleringen vil også energien fluktuere veldig før den når en omtrent konstant verdi. Før dette bør man ikke sample energien. Det er først når denne har blitt tilnærmet konstant at termisk likevekt er nådd. Selv etter dette vil det være små fluktuasjoner i energien grunnet at integreringsalgoritmen ikke bevarer energi perfekt. Generelt er vi interessert i å måle makroskopiske størrelser for systemet, og dette kan vi først gjøre når systemet har nådd en likevektstilstand.

Temperature: Ved å la systemet utvikle seg fritt etter initialiseringen vil temperaturen falle dramatisk før den når en likevektstilstand den fluktuerte rundt. I våre simuleringer så vi at den falt til omtrent halve den originale verdien. Dette er gjerne ikke det man ønsker, så for å holde temperaturen på et ønsket nivå kan man benytte seg av termostater i simuleringene. Disse bruker forskjellige metoder for å styre temperaturen mot et ønsket nivå, ved å simulere kontakt med et eksternt varmebad. f.eks. ved å multiplisere hastighetene med en skaleringsfaktor.

I klassiske md-simuleringer er energien, partikkeltallet og trykket konstant, med andre ord er det det mikrokanoniske ensemblet som simuleres. I det mikrokanoniske ensemblet kan det vises at temperaturen kan estimeres som $T = \frac{2\langle E_k \rangle}{3Nk_B}$. Med andre ord kan man finne den gjennomsnittelige temperaturen i systemet fra den gjennomsnittelige kinetiske energien. Den indre kinetiske energien finnes ved å summere opp bidragene fra alle partiklene. Dermed ser vi at temperaturen er styrt av enkeltpartiklenes hastigheter, og vi skjønner at en reskalering av disse kan gi ønsket temperatur. Dette kan gjøres på mange forskjellige måter (gange med skaleringsfaktor, simulere kollisjoner ved å gi ny hastighet trukket fra en Boltzmann dist). Ulempen er at termostater kan påvirke simuleringene på negativ måte (forstyrre dynamikken i gittervibrasjonene etc) \rightarrow ufysisk dynamikk i systemet.

3. Molecular-dynamics in the micro-canonical 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?

MD-simuleringer kan brukes til å måle forskjellige makroskopiske størrelser for et system, ved å måle forskjellige gjennomsnittelige størrelser i systemet som simuleres. En konsekvens av den ergodiske hypotesen (tiden et system har en gitt verdi for en observabel er proporsjonal til volumet i faserommet hvor observabelen har denne verdien) er at ensemble-gjennomsnittet og tids-gjennomsnittet for en variabel er det samme. Dette betyr at vi, ved å simulere (korrekt) over lange tider, kan måle makroskopiske størrelser i systemet vårt og dermed forutsi likevekts-størrelser for ekte systemer.

For å måle temperaturen i systemet er en løsning å bruke ekvipartisjonsprinsippet. Det sier at ved likevekt vil enhver frihetsgrad som er kvadratisk i Hamiltonfunksjonen (energien) har en gjennomsnittsenergi på $\frac{1}{2}kT$. For en ideell gass har vi $H_{kin} = \frac{1}{2}m(v_x^2 + v_y^2 + v_z^2)$, noe som vil si at vi har tre kvadratiske frihetsgrader per partikkel. Da vil gjennomsnittsenergien være $\langle E_{kin} \rangle = 3NkT/2$. Vi kan dermed måle temperaturen ved å finne den totale kinetiske energien $E_{kin} = \frac{1}{2}m(v_x^2 + v_y^2 + v_z^2)$, og regne ut $T = \frac{2E_{kin}}{3Nk}$. Hvis vi har mange partikler er dette en god tilnærming.

I et fleratomig system finnes det mange metoder for å måle trykket. I det mikrokanoniske ensemblet kan man bruke viriallikningen for å finne at det gjennomsnittelige trykket i et volum V med partikkeltetthet ρ er gitt av $P = \rho k_B T + \frac{1}{3V} \sum_{i < j} \vec{F}_{ij} \cdot \vec{r}_{ij}$ (uten vekselvirkninger reduseres denne til ideal gass lov). En annen metode er å måle trykket fra den radielle distribusjonsfunksjonen.

Når man vil måle makroskopiske størrelser for systemet må man tenke på hvilke størrelser som kan måles direkte; $N, V, v, \text{posisjon}$. Alle andre størrelser man vil måle må da kunne uttrykkes direkte fra disse.

Fasen til systemet bestemmes ikke bare av temperaturen, men også av partikkeltettheten ρ . Dette må man tenke på når man skal sette opp systemet (-> hvilken fase vil du studere?). Det er også veldig viktig at man venter med å måle til man har nådd en likevektstilstand.

+What can it be used for?

- forutsi likevekts-verdier for makroskopiske størrelser i virkelige systemer, få innsikt i statistiske egenskaper og fluktuasjoner for disse. Studere systemer det er vanskelig å studere eksperimentelt, måle størrelser det er vanskelig å måle eksperimentelt, studere systemer under forhold det er vanskelig å gjenskape eksperimentelt.

4. Measuring the diffusion constant in molecular-dynamics simulations

How to measure the diffusion constant in molecular dynamics simulations? -> limitations and challenges. Compare with methods and results from random walk modeling.

Diffusjon er en prosess som beskriver en partikkels netto bevegelse fra et område med høy konsentrasjon til et med lav konsentrasjon, altså beskriver det hvordan partikler sprer seg utover til de er uniformt fordelt. I et system som vårt har vi ikke noen konsentrasjonsgradient, så da er det isteden snakk om såkalt self-diffusion. Denne er da altså ikke knyttet til noen tetthetsforskjell, men beskriver partiklenes termiske bevegelse i systemet.

I en molekylærsimulering kan diffusjonskonstanten finnes ved å måle mean square displacement for partiklene i systemet. Mean square displacement karakteriserer partiklenes bevegelse i tid, og er gitt ved

$$\langle r^2(t) \rangle = \frac{1}{N} \sum_{i=1}^N (\vec{r}(t) - \vec{r}_{initial}). \quad (6)$$

For en virrevander vil mean square displacement være relatert til diffusjonskonstanten gjennom uttrykket

$$\langle r^2(t) \rangle = 2dDt. \quad (7)$$

Her er d dimensjonaliteten, som i vårt tilfelle da er lik 3, ettersom vi studerer et system i tre dimensjoner. Dette er et helt generelt resultat som gjelder når tiden går mot uendelig. Fra dette ser vi at dersom vi ser på partiklene som virrevandrere kan vi finne diffusjonskonstanten fra uttrykket over. Det viser seg også at bevegelsen til en virrevander i tre dimensjoner er en god tilnærming til et atoms bevegelse i en væske når tiden går mot uendelig og N er stor.

Når man måler msd er det veldig viktig at man tar spesielt hensyn til boundary conditions.

+ se Mathilde

5. Measuring the radial distribution in molecular-dynamics simulations

How can you measure the radial distribution function in molecular dynamics simulations. 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.

Den radielle distribusjonsfunksjonen, eller pair correlation function, er et verktøy for å karakterisere den mikroskopiske strukturen til et fluid. Den tilsvarer den radielle sannsynligheten for å finne et annet atom i avstanden r fra et tilfeldig atom, eller (tilsvarende) atomtettheten i et sfærisk skall med radius r rundt et atom. Den radielle distribusjonsfunksjonen kan måles ved å telle antall partikler i et slikt sfærisk skall for økende radius.

Dette kan gjøres ved å sortere naboene rundt hvert atom inn i "distance bins" og ta gjennomsnittet av antallet partikler i hver enkelt bin over simuleringen. I denne prosessen er det viktig at vi husker på å dele på volumet av kuleskallet ettersom det er partikkeltettheten vi er interessert i. Fra den radielle distribusjonsfunksjonen kan man få informasjon om strukturen til stoffet og man kan også finne trykket P , og kompressibiliteten til fluiden.

Vi gjorde dette for et system med argonatomer. Disse atomene ble initielt plassert i en gitterstruktur bygget opp av atomer i såkalte face centered cubic lattices. Når vi målte den radielle distribusjonsfunksjonen for dette systemet fikk vi klare, definerte "spikes". Dette er en følge av at alle partiklene er plassert likt i forhold til hverandre, i en gjentakende gitterstruktur. Det er da svært lite variasjon i avstanden mellom to atomer med gitte plasser i strukturen, kun mindre vibreringer. (tegn ev cube, forklar første/høyeste spike.) Når temperaturen øker ser vi at vi fortsatt har den samme underliggende strukturen, og vi kjenner igjen de samme toppene, men de har nå blitt litt bredere. Dette kommer av at den økede temperaturen gir partiklene større kinetisk energi, og de beveger seg mer rundt slik at det blir mer variasjon i posisjonene. Der hvor spikesene var veldig tette glir de over i én tykkere spike. Øker vi temperaturen enda mer, slik at vi får en gass, vil den radielle distribusjonsfunksjonen miste mer av formen sin, og det blir vanskeligere å kjenne igjen toppene fra krystallstrukturen. Vi vil fortsatt

ha en relativt skarp topp i starten, ettersom partiklene er sterkt frastøtende ved korte avstander og ikke vil komme for nære. For større avstander blir distribusjonen mer og mer jevn. (Figurer/tegninger!)

Utfordringer når det gjelder å finne denne distribusjonen er blant annet at utregningen er computationally intensive siden vi regner på alle atomer. Man bør også passe på å ikke velge binsize for liten ettersom man da ender opp med å dele på et veldig lite volum, som kan føre til store feil pga maskinrepresentasjon.

For ukorrelerte partikler, slik som i en ideell gass, vil $g(r) = 1$ (for r over minsteavstanden) noe som tilsvarer en uniform sannsynlighetsfordeling. Ethvert avvik fra dette skyldes altså vekselvirkninger mellom atomene/partiklene.

+ compare to random walks: Kanskje det at tetthetsdistribusjonen for random walks glir utover i tid? Eller at random walkers er ukorrelerte -> som partikler i ideell gass.

6. Thermostats in molecular-dynamics simulations

Discuss the micro-canonical vs the canonical ensemble in molecular-dynamics 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 micro-canonical simulation?

Det mikrokanoniske ensemblet beskriver isolerte systemer. Det vil si systemer der antallet partikler, energien og volumet er konstant. I det kanoniske ensemblet kan systemer utveksle energi, mens det istedet er temperaturen som holdes konstant (altså N, T og V konstant). I en klassisk, standard md-simulering av et system vil det være energien, volumet og antallet partikler som holdes konstante. Da befinner vi oss i det mikrokanoniske ensemblet. Men vi har ofte interesse av å isteden simulere det kanoniske ensemblet. For å oppnå dette trenger vi da å holde temperaturen konstant i systemet vårt. Dette kan oppnås ved å simulere kontakt med et eksternt varmebad (selv om temperaturen vil være tilnærmet konstant for stor $N \rightarrow$ små fluktuasjoner). Dette implementeres ved hjelp av såkalte termostater. En god thermostat må fylle visse krav. Den bør være i stand til å holde systemtemperaturen på et relativt konstant nivå rundt varmebadstemperaturen. Den bør bevare dynamikken i systemet (ikke forstyrre) og sample faserommet tilsvarende det kanoniske ensemblet. I tillegg bør den være justerbar. Den thermostat som oppfyller disse kravene på best måte er Nose-Hoover thermostat, som samler det kanoniske ensemblet spesielt godt. Ettersom denne er relativt komplisert å implementere har vi isteden sett på to enklere termostater.

The Berendsen thermostat: Mange termostater simulerer interaksjon med et eksternt varmebad ved å reskalere hastighetene til alle atomene med en skaleringsfaktor γ . Blant disse finner vi **Berendsen thermostat**. For denne thermostat er skaleringsfaktoren gitt som

$$\gamma = \sqrt{1 + \frac{\Delta t}{\tau} \left(\frac{T_{bath}}{T} - 1 \right)} \quad (8)$$

τ kalles relaxation time og åpner for justering av (den svake) koblingen til varmebadet (større τ gir svakere kobling). Berendsentermostaten virker altså ved å dempe fluktuasjonene i systemets kinetiske energi. Hastighetskaleringen er slik at endringen i temperatur er proporsjonal med temperaturforskjellen mellom systemet og varmebadet.

$$\Delta T = \frac{\Delta t}{\tau} (T_0 - T(t)) \quad (9)$$

Termostaten gjensker ikke et korrekt kanonisk ensemble, men for store nok systemer vil de fleste resultatene være tilnærmet korrekte. Berendsen er en populær thermostat ettersom den termaliserer systemet til varmebadstemperaturen på en effektiv måte. Derfor brukes den ofte til å initialisere et systemet til en likevektstilstand, før man deretter tar i bruk Nosé-Hoover thermostat.

The Andersen thermostat: Andersen thermostat er en annen relativt enkel thermostat. Den forsøker å simulere det eksterne varmebadet ved å simulere (harde) kollisjoner mellom atomer i systemet og atomer i varmebadet. Dette gjøres ved at det for alle atomer i systemet trekkes et tilfeldig tall mellom 0 og 1 fra en uniform distribusjon. Dersom tallet som trekkes er mindre enn $\Delta t/\tau$ vil dette atomet være ett av de "kolliderende atomene" og atomet får tildelt en ny hastighet trukket fra en Boltzmann fordeling med standardavvik $\sqrt{k_B T_{bath}/m}$. τ kan her sees på som kollisjonstiden, og gjennom denne kan man jusere thermostat. Den angir styrken på koblingen til varmebadet. Andersen thermostat vil gi en kanonisk fordeling av mikrotillstander (hvis man antar ergodicity?). Algoritmen er ikke-deterministisk og dermed ikke tidsreversibel. Termostaten er nyttig

for å få systemet til en likevektstilstand, men den forstyrrer systemet på en uheldig måte ettersom den forstyrrer den naturlige dynamikken i systemet, blant annet ved å forstyrre vibrasjonene i krystallstrukturen.

(Wiki: Note: the Andersen thermostat should only be used for time-independent properties. Dynamic properties, such as the diffusion, should not be calculated if the system is thermostated using the Andersen algorithm.)

Fra prosjekt 1 så vi at vi fikk større fluktuasjoner for Andersen thermostaten enn for Berendsen thermostaten. Dersom vi ønsker å gjøre simuleringer i det mikrokanoniske ensemblet ved en ca temperatur T kan man bruke Berendsen thermostaten for å raskt nå denne temperaturen.

Advanced Molecular Dynamics

7. Generating a nano-porous material

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

Når vi har villet simulere et nanoporøst system har vi startet fra et termalisert system bestående av en væske med argonatomer beskrevet av L-J potensialet. Deretter har vi kunnet velge hvilke deler av systemet som skal være fast stoff og dermed danne den nanoporøse matrisen, mens resten av atomene da er væsken i systemet. For å lage denne matrisen lar man de utvalgte atomene ikke kunne bevege seg, men fortsatt vekselvirke med atomene rundt.

Det finnes mange måter å velge ut hvilke atomer som skal være del av det faste stoffet på. En enkel metode er å la dette være tilfeldig, f.eks ved å la hvert atom trekke et tall, og hvis dette tallet er større/mindre en en gitt verdi blir dette atomet ubevegelig. Det viser seg at denne metoden gir svært små porer. En annen metode innebærer å fordele (muligens overlappende) kuler med tilfeldig størrelse og tilfeldig posisjon utover i systemet. Da kan man velge å la de partiklene som er innenfor en eller flere kuler være ubevegelige, eller motsatt. I prosjekt 2 valgte vi materialet innenfor kulene som fast stoff, og det som var utenfor som porer. Porøsiteten er den relative mengden porerom i det totale volumet. For å oppnå en gitt porøsitet kan man legge til kuler helt til man når ønsket porøsitet. En tilnærming til porøsiteten er forholdet mellom antall partikler innenfor kulene og antall partikler utenfor, $\phi = N_B/N$ ($\phi = 1 - V_{matrix}/V$) ettersom porøsiteten er den relative delen av det totale volumet som er pore-volum). Dette må måles rett etter man har valgt ut de ubevegelige partiklene. Ofte velger man å redusere tettheten av væsken i det nanoporøse systemet, og da blir det færre bevegelige partikler og porøsiteten blir mindre enn den egentlig er hvis man måler først da.

Væsken i det nanoporøse systemet og det nanoporøse mediet kan karakteriseres ved å måle flow i systemet og fra dette finne væskens viskositet μ og mediets permeabilitet k . Væskens flow gjennom det porøse mediet beskrives av Darcy's lov

$$U = \frac{k}{\mu}(\Delta P - \rho g). \quad (10)$$

Her er U volumfluksen (volum per areal og tid) og $\rho = \frac{Nm}{V}$ er massetettheten. Det siste leddet kan omskrives på følgende måte

$$\rho g = \frac{Nm}{V}g = \frac{N}{V}F_g = nF_g. \quad (11)$$

Her er $n = \frac{N}{V}$ number density og F_g er tyngdekraften. Denne kraften kan byttes ut med en annen kraft, og dette gjorde vi for systemet vårt, for å indusere flow. Dette er en måte å simulere trykkforskjell på. Uten trykkforskjell blir Darcy's lov på formen

$$U = \frac{k}{\mu}nF_x \quad (12)$$

og vi kan bruke den til å estimere μ og k .

Viskositeten μ forteller oss hvor tyktflytende en væske er, eller hvor "motstandsdyktig" væsken er mot endring (honning har større viskositet enn vann). Permeabiliteten k er et mål for evnen materialet har til å transportere væske, altså hvor lett væske strømmer gjennom materialet, og er uavhengig av hvilken væske man ser på. Lav permeabilitet vil si lite gjennomstrømming. For en sylinder med lengde L og radius a , forventer vi en hastighetsprofil $u(r) = \frac{nF_x}{4\mu}(a^2 - r^2)$ når vi har nådd en stasjonær tilstand. For å indusere flow i systemet å legger vi til en ekstern kraft i for eksempel x-retning (litt som tyngdekraft), F_x . For å måle μ lager vi et porøst

system der den eneste poren er en syllinder, med en eller annen radius a og setter på den ytre kraften. Deretter må vi la systemet utvikle seg til vi har nådd en stasjonær tilstand, som vil si at partiklene i gjennomsnitt ikke lenger blir akselerert, og temperaturen holder seg omtrent konstant. Dette kan ta lang tid. Nå måler vi hastighetsprofilen ved å dele radien inn i bins og finner gjennomsnittshastigheten for hver bin og deler på antall partikler i hver bin. Så kan man sammenlikne med $u(r)$ og på den måten finne μ . Når μ er funnet for en bestemt væske, kan vi måle permeabiliteten til forskjellige porøse materialer ved å måle væskefluksen for denne væsken.

$$k = \frac{U\mu}{nF_x}. \quad (13)$$

8. Diffusion a nano-porous 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

Diffusjonskonstanten kan finnes ved å måle atomenes forflytning $\langle r^2(t) \rangle$ på samme måte som i oppgave fire. Da har vi

$$\langle r^2(t) \rangle = 2dDt \quad (14)$$

og kan finne en lineær tilpasning til linja vi får for $\langle r^2(t) \rangle$. I et nanoporøst system har partiklene begrenset plass å bevege seg på. Da er det naturlig at forflytningen blir mindre enn for en væske som ikke er begrenset. Når forflytningen er mindre, vil diffusjonskonstanten bli mindre. Dette så vi i prosjektene. Vi kan tenke på det som at spektraldimensjonen til det nanoporøse systemet er mindre enn $d = 3$. Da vil diffusjonskonstanten være den samme, det er dimensjonen som har minnet. Vi forventer at denne oppførselen blir tydeligere med mindre porestørrelser. En annen forandring fra bulk er at i et porøst system kan diffusjonskonstanten gjerne variere med retningen, ettersom porene har en orientering og kan være "mindre" i f.eks x-retning enn y-retning. Hvis porene er generert på en tilfeldig måte, eller orienteringen deres er tilfeldig, vil dette nødvendigvis utlignes på større skala. For alle diffusjonsprosesser forventer vi at det oppnås en likevektstilstand etter en viss tid.

9. Flow in a nano-porous material

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

Se oppgave 7.

Væsken i det nanoporøse systemet og det nanoporøse mediet kan karakteriseres ved å måle flow i systemet og fra dette finne væskens viskositet μ og mediets permeabilitet k . Væskens flow gjennom det porøse mediet beskrives av Darcy's lov

$$U = \frac{k}{\mu}(\Delta P - \rho g). \quad (15)$$

Her er U volumfluksen (volum per areal og tid) og $\rho = \frac{Nm}{V}$ er massetettheten. Det siste leddet kan omskrives på følgende måte

$$\rho g = \frac{Nm}{V}g = \frac{N}{V}F_g = nF_g. \quad (16)$$

Her er $n = \frac{N}{V}$ number density og F_g er tyngdekraften. Denne kraften kan byttes ut med en annen kraft, og dette gjorde vi for systemet vårt, for å indusere flow. Flow kan med andre ord induseres ved å legge til en (fiktiv) kraft i en valgt retning. Dette har samme effekt som en trykkgradient i den retningen, men er lettere å implementere i simuleringen. I praksis lar man da alle partiklene få et ekstra ledd i kraften som virker på dem. Uten trykkforskjell blir Darcy's lov redusert til

$$U = \frac{k}{\mu}nF_x \quad (17)$$

og vi kan bruke den til å estimere μ og k .

Viskositeten μ forteller oss hvor tyktflytende en væske er, eller hvor "motstandsdyktig" væsken er mot endring (honning har større viskositet enn vann). Permeabiliteten k er et mål for evnen materialet har til å transportere væske, altså hvor lett væske strømmer gjennom materialet, og er uavhengig av hvilken væske man

ser på. Lav permeabilitet vil si lite gjennomstrømming. For en sylinder med lengde L og radius a , forventer vi en hastighetsprofil $u(r) = \frac{nF_x}{4\mu}(a^2 - r^2)$ når vi har nådd en stasjonær tilstand. For å indusere flow i systemet å legger vi til en ekstern kraft i for eksempel x-retning (litt som tyngdekraft), F_x . For å måle μ lager vi et porøst system der den eneste poren er en sylinder, med en eller annen radius a og setter på den ytre kraften. Deretter må vi la systemet utvikle seg til vi har nådd en stasjonær tilstand, som vil si at partiklene i gjennomsnitt ikke lenger blir akselerert, og temperaturen holder seg omtrent konstant. Dette kan ta lang tid. Nå måler vi hastighetsprofilen ved å dele radien inn i bins og finner gjennomsnittshastigheten for hver bin og deler på antall partikler i hver bin. Så kan man sammenlikne med $u(r)$ og på den måten finne μ . Når μ er funnet for en bestemt væske, kan vi måle permeabiliteten til forskjellige porøse materialer ved å måle væskefluksen for denne væsken.

$$k = \frac{U\mu}{nF_x}. \quad (18)$$

Utfordringer: Det er viktig at man måler på en steady state, altså at man har nådd en tilstand hvor temperaturen er konstant og systemet har sluttet å akselerere. Det kan ta lang tid.

1 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?

For å generere et perkolasjonssystem kan man starte med en matrise $L_x \times L_y$ hvor man lar hvert element i matrisen være et tall trukket fra en uniform sannsynlighetsfordeling (ukorrolerte tall). Vi velger så en sannsynlighet for at en site skal være besatt, p . Dette tallet angir systemets porøsitet. Alle sites/elementer i matrisen der det trukkede tallet er mindre enn p regnes som besatt (tallet settes til én), de andre er tomme sites (settes til null), og vi har fått dannet et perkolasjonssystem. "Matrisen" kan i teorien ha en hvilken som helst form.

Clustere er områder med sammenhengende sites. Det finnes flere alternativer for hva som teller som sammenhengende. I prosjektet vårt valgte vi å se på systemer der en site regnes som å ha fire naboer (sør, nord, øst og vest). To naboer som er besatte sites regnes som connected, og en gruppe connected sites er et cluster. Når et cluster rekker fra nord til sør eller fra øst til vest sier vi at det er et spanning cluster. Vi sier da at systemet er percolated ettersom det da går en "vei" fra nord/øst til sør/vest.

Generering og visualisering av et perkolasjonssystem:

For å analysere systemet kan vi studere en rekke størrelser, som for eksempel cluster number density, $n(p, L)$, sannsynligheten for at en tilfeldig site tilhører et perkoleringscluster, $P(p, L)$, eller perkolasjonssannsynligheten $\Pi(p, L)$.

Vi kan tenke oss at clustrene er hulrom i et 2-dimensjonalt materiale, og for at vi skal kunne ha noe gjennomstrømming (som er det vi ønsker å studere) må minst et av clusterne være såkalte spanning clusters. Perkolasjonssannsynligheten beskriver sannsynligheten for at det i et system med en gitt L og p finnes et perkoleringscluster. Verdien for p sammen med systemstørrelsen L avgjør altså sannsynligheten for å få et perkoleringscluster i systemet. Denne sannsynlighet er gitt av $\Pi(p, L)$. For å finne denne sannsynligheten gjør vi målinger på systemer med varierende verdier for L (kanskje 10). For hver L -verdi lager vi systemer for mange forskjellige p verdier. Mange slike eksperimenter gjennomføres for hver kombinasjon av p og L verdi, og i hvert tilfelle kontrolleres det om systemet har et spanning cluster. Man kan undersøke om det finnes et spanning cluster på flere måter, blant annet kan man sjekke om størrelsen på et av clusterene er lik systemstørrelsen, altså L_x i x-retning, eller L_y i y-retning. Ved å dele antallet forsøk som gir et spanning cluster på totalt antall forsøk for hvert sett med L og p verdier får vi et tilnærming for $\Pi(L, p)$. For et uendelig stort system vil $\Pi(p, L)$ være en step-funksjon som går fra null til en ved $p = p_c$ som er perkoleringsgrensen.

Det finnes mange metoder for å studere systemet nærmere. Med matlabs **regionprops** funksjon kan man få tilgang til mye informasjon om systemet slik som lengden på alle clusterene, størrelsen deres osv. Dette kan brukes til å studere systemers gjennomsnittelige clusterstr, cluster number density etc. Det viser seg at mange av funksjonene man kan bruke til å beskrive systemet følger såkalt power laws. Et eksempel er $P(p, L) \propto (p - p_c)^\beta$, for $p > p_c$. Vi kan da finne β ved først å gjøre eksperimenter for å finne $P(p, L)$ og finne β fra et log-log plot.

En annen metode som brukes mye for å analysere perkolasjonssystemer er det å generere et data-collapse plot. Hovedidéen er å plote en funksjon $F(G(x))$ som en funksjon av $G(x)$ for en rekke målinger for å finne x . Man prøver forskjellige verdier til man finner den x -verdien der kurvene legger seg oppå hverandre og danner en universalkurve.

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 (figur 1.3 s.7). Relate to your simulations. How do you calculate these quantities and how do you measure them in simulations?

For små systemstørrelser, slik som en 2×2 matrise, er det mulig å finne en eksakt løsning for $\Pi(L, p)$ såvel som $P(L, p)$. For et system på størrelse 1×1 er problemet veldig enkelt. Det finnes bare to tilstander; enten er siden tom eller så er den opptatt. Dette gir en porøsitet på henholdsvis 0 og 1, og dermed en perkolasjonssannsynlighet på henholdsvis 0 og 1. Så en grafisk representasjon av perkolasjonssannsynlighet gir en rett linje fra (0,0) til (1,1). $P(L, p)$ vil bli identisk. I tilfellet der systemstørrelsen øker til 2×2 , har vi fortsatt en begrenset mengde muligheter (se figur 1.2 s. 6). Systemet har totalt 6 forskjellige konfigurasjoner å velge mellom, med varierende grad av degenerasjon.

Tabell som viser Π og P for de forskjellige statesene, hvor p er sannsynligheten for at en gitt site er besatt:

c	1	2	3	4	5	6
Probability(c)	$(1-p)^4$	$p(1-p)^3$	$p^2(1-p)^2$	$p^2(1-p)^2$	$p^3(1-p)$	p^4
Degeneracy(c)	1	4	2	4	4	1
$\Pi(c)$	0	0	0	1	1	1
$P(c)$	0	0	$\frac{3}{4}$	0	$\frac{1}{2}$	1

Her er Π funnet ved bruk av formelen $P(A) = \sum_B P(A|B)P(B)$, som gir $\Pi(p) = \sum_c \Pi(p|c)P(c)$. Fra dette får vi

$$\Pi(p, L=2) = 4p^2(1-p)^2 + 4p^3(1-p) + p^4 \quad (19)$$

og

$$P(p, L=2) = 2p^2(1-p)^2 + 3p^3(1-p) + p^4. \quad (20)$$

Vi ser at for en finitt L vil $\Pi(L, p)$ kunne uttrykkes som et polynom av orden $o = L^2$. Samtidig ser vi at antallet konfigurasjoner for systemet er 2^o som vi da ser at øker svært raskt når L øker. Med andre ord er denne teknikken med å finne alle de mulige konfigurasjonene og deres degenerasjon ikke en teknikk som det er mulig å bruke for store L . Likevel er det mye å lære fra dette lille systemet. En ting man kan merke er at

$$\Pi(L, p) \simeq Lp^L + c_1p^{L+1} + \dots + c_n p^{L^2} \quad (21)$$

i grensen $p \ll 1$. Når $p \rightarrow 0$ vil det dominerende leddet være Lp^L . Når $p \rightarrow 1$ vil det dominerende leddet være omtrent lik $1 - (1-p)^L$ (?).

Måle $\Pi(p, L=2)$: Vi genererer systemer for mange forskjellige p verdier, og i hvert tilfelle kontrolleres det om systemet har et spanning cluster. Man kan undersøke om det finnes et spanning cluster på flere måter, blant annet kan man sjekke om størrelsen på et av clusterene er lik systemstørrelsen, altså L_x i x-retning, eller L_y i y-retning. Ved å dele antallet forsøk som gir et spanning cluster på totalt antall forsøk for hver p verdier får vi et tilnærming for $\Pi(L, p)$.

Måle $P(p, L=2)$: Tilsvarende som for Π , men vi ser kun på de tilfellene der vi har et spanning cluster, og måler da størrelsen på clusteret (som i vår tilfelle da er 2,3 eller 4). Størrelsen deles på den totale størrelsen (her 4). Dette gjøres mange ganger for mange forskjellige p -verdier, og tilslutt har vi en tilnærming til P .

12. Cluster number density in 1-d percolation

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

Perkolasjon kan studeres nøyaktig i to grenser; én dimensjon og uendelig dimensjon. I 1d er problemet noe forenklet, men det kan likevel gi oss innsikt i problemet, og mange av konseptene kan generaliseres til høyere dimensjoner. I et 1d system bestående av L sites vil vi kun ha et spanning cluster dersom alle sitesene er okkuperte. Dette gir oss med en gang perkolasjonssannsynligheten som $\Pi(p, L) = p^L$, og vi ser at dennes oppførsel er triviell når $L \rightarrow \infty$. Dermed ser vi at i dette problemet er perkoleringsgrensen $p_c = 1$. Cluster number density $n(s, p)$ angir sannsynligheten for at en site er en spesifikk site, for eksempel den lengst til venstre, i et cluster av størrelse s . Denne kan uttrykkes som

$$n(s, p) = (1-p)p^s(1-p) = (1-p)^2 p^s \quad (22)$$

ettersom et cluster av størrelse s i 1-d vil bestå av s okkuperte sites, hver med sannsynlighet p , med en tom site på hver side, hver med sannsynlighet $(1 - p)$. Sannsynligheten for at en tilfeldig site er del av et cluster av størrelse s blir da naturlig nok $sn(s, p)$, som angir distribusjonen av cluster-størrelser. Dette ser vi fordi $n(s, p)$ er sannsynligheten for at en site har en spesifikk posisjon i et cluster av størrelse s , og det finnes da s slike posisjoner. Vi er også interessert i å sjekke om $ns(p, s)$ faktisk er en normalisert tetthet

$$p = \sum_s sn(s, p) = \sum_{s=1}^{\infty} sp^s(1 - p)s = (1 - p)^2 p \sum_{s=1}^{\infty} sp^{s-1} \quad (23)$$

Ved hjelp av et lite triks ser vi

$$\sum_s sn(s, p) = (1 - p)^2 p \frac{d}{dp} \left(\sum p^s \right) = p. \quad (24)$$

For å måle cluster number density i et forsøk kan vi først telle antallet cluster av størrelse N_s i systemet. Det totale antallet okkuperte sites i systemet er gitt av $\tilde{N} = pL$, og da ser vi at vi bør ha

$$\tilde{N} = pL = \sum_s sN_s. \quad (25)$$

Sannsynligheten for at en gitt *okkupert* site er del av et cluster av størrelse s er da gitt av uttrykket $\frac{sN_s}{\tilde{N}}$. Sannsynligheten for at en tilfeldig site er del av et cluster av størrelse s finner vi da ved å multiplisere med p

$$sn(s, p) = \frac{psN_s}{\tilde{N}} = \frac{sN_s}{L}. \quad (26)$$

Fra dette finner vi cluster number density

$$n(s, p) = \frac{N_s}{L^d} \quad (27)$$

d'en her indikerer at resultatet vi har fått gjelder uansett dimensjon.

Vi kan omskrive cluster number density på følgende måte

$$n(s, p) = (1 - p)^2 p^s = (1 - p)^2 e^{s \ln p} = (1 - p)^2 e^{-s/s_\xi}. \quad (28)$$

Her har vi introdusert variabelen $s_\xi = -\frac{1}{\ln p}$ som beskriver en cut-off cluster størrelse. Sturerer man $n(s, p)$ vil man med andre ord se eksponentialkurver med karakteristisk lengde s_ξ (se fig 2.2 s.13). Disse vil divergere når p går mot p_c . Når p er 1 kan vi skrive

$$\ln p = \ln(1 - (1 - p)) \simeq -(1 - p). \quad (29)$$

Fra dette får vi dermed

$$s_\xi = \frac{1}{1 - p} = \frac{1}{p_c - p} = |p - p_c|^{-1/\sigma} \quad (30)$$

for den karakteristiske lengden. Avviket til s_ξ når $p \rightarrow p_c$ er en power law med eksponent -1, og dette gjelder generelt i perkolasjonsteori. Verdien til eksponenten σ vil avhenge av dimensjonalitet, men ikke detaljene for strukturen (slik som hvor mange naboer som regnes som connected).

13. Corrolation length in 1-d percolation

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

Korrelasjonsfunksjonen $g(r)$ beskriver den betingede sannsynligheten for at to okkuperte sites a og b som er separert av en avstand r hører til samme cluster. I 1d tilfellet må da alle sites mellom a og b være okkuperte sites. Ved å la r angi antallet sites mellom a og b får vi da for korrelasjonsfunksjonen

$$g(r) = p^r = e^{-r/\xi}. \quad (31)$$

Her har vi introdusert størrelsen $\xi = -\frac{1}{\ln p}$ som kalles korrelasjonslengden. Denne vil divergere når $p \rightarrow p_c$. Ved å bruke at p_c i det éndimensjonale tilfellet er 1, kan vi skrive

$$\ln p \simeq -(1-p), \quad (32)$$

noe som gir oss korrelasjonslengden $\xi = \xi_0(p_c - p)^{-\nu}$ med $\nu = 1$. Fra dette ser vi at korrelasjonslengden divergerer (blir veldig stor) som en powerlaw når $p \rightarrow p_c = 1$. Dette er et resultat som gjelder generelt i perkolasjonsteori, men verdien til ν vil variere med dimensjonaliteten. Korrelasjonslengden kan relateres til den finitte systemstørrelsen L . Når $\xi \ll L$ vil ikke effektene av at vi har et finitt system istedenfor et uendelig ett merkes. Dette er fordi vi da ikke vil ha noen clusterer som er store nok til å “merke” at systemet er finitt. Når $\xi \gg L$ vil systemets oppførsel domineres av systemstørrelsen. Når ξ er nærme L , $\xi \approx L$ er det vanskelig å avgjøre hvor nærme vi er p_c , men vi kan estimere. $L \approx \xi \propto |p - p_c|^{-\nu} \Rightarrow |p - p_c| \propto L^{-1/\nu}$.

14. Cluster size in 1-d percolation

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

clusterstørrelsen beskriver antallet besatte sites i et gitt cluster, s . I én dimensjon har vi at cluster number density er $n(s, p) = (1-p)^2 p^s$. Denne kan sees på som sannsynligheten for å finne clusterer av størrelse s i systemet, og hvis vi multipliserer denne størrelsen med clusterstørrelsen s får vi derfor sannsynligheten for at en site tilhører et cluster av størrelse s . $n(s, p)$ kan skrives om som

$$n(s, p) = (1-p)^2 p^s = (1-p)^2 e^{s \ln p} = (1-p)^2 e^{-s/s_\xi}. \quad (33)$$

Her har vi definert $s_\xi = 1/\ln p$. Denne størrelsen kan sees på som et mål på når det blir usannsynlig å finne større clusterer. s_ξ kalles gjerne for den karakteristiske clusterstørrelsen. I 1D er $p_c = 1$, og vi kan omskrive denne som

$$e_\xi = \frac{1}{p_c - p} = |p - p_c|^{-1/\sigma} \quad (34)$$

med $\sigma = 1$ for 1D. Vi ser at s_ξ vokser med p , og når $p \rightarrow p_c$ divergerer den.

I to dimensjoner ser vi en lignende oppførsel for den karakteristiske clusterstørrelsen som også her er på formen

$$e_\xi = |p - p_c|^{-1/\sigma} \quad (35)$$

Den karakteristiske cluster størrelsen s_ξ er definert i oppgave 12

15. Measurement and behaviour 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?

$P(p, L)$ beskriver tettheten til spanning clusteret, ettersom den gir sannsynligheten for at en site tilhører spanning clusteret. Vi har spanning cluster kun når $p \geq p_c$. I 1d er dette derfor ikke den mest interessante størrelsen. Den er 0 for $p < 1$ og 1 ellers. Det vi kan se er at ettersom sannsynligheten for at en site er besatt på være lik summen av sannsynligheten for at den er del av et finitt cluster, og sannsynligheten for at den er del av spanning clusteret, $P(p, L)$. Dermed får vi

$$P(p, L) = p - \sum_s sn(s, p). \quad (36)$$

16. The cluster number density

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

Cluster number density $n(s, p)$ angir sannsynligheten for at en site er en spesifikk site, for eksempel den lengst til venstre, i et cluster av størrelse s . Denne kan uttrykkes som

$$n(s, p) = (1-p)p^s(1-p) = (1-p)^2 p^s \quad (37)$$

ettersom et cluster av størrelse s i 1-d vil bestå av s okkuperte sites, hver med sannsynlighet p , med en tom site på hver side, hver med sannsynlighet $(1-p)$. Sannsynligheten for at en tilfeldig site er del av et cluster av størrelse s blir da naturlig nok $sn(s, p)$, som er distribusjonen av cluster-størrelser. Dette ser vi fordi $n(s, p)$

er sannsynligheten for at en site har en spesifikk posisjon i et cluster av størrelse s , og det finnes da s slike posisjoner.

For å måle cluster number density i et forsøk kan vi først telle antallet cluster av størrelse N_s i systemet. Det totale antallet okkuperte sites i systemet er gitt av $\tilde{N} = pL$, og da ser vi at vi bør ha

$$\tilde{N} = pL = \sum_s sN_s. \quad (38)$$

Sannsynligheten for at en gitt *okkupert* site er del av et cluster av størrelse s er da gitt av uttrykket $\frac{sN_s}{\tilde{N}}$. Sannsynligheten for at en tilfeldig site er del av et cluster av størrelse s finner vi da ved å multiplisere med p

$$sn(s, p) = \frac{psN_s}{\tilde{N}} = \frac{sN_s}{L}. \quad (39)$$

Fra dette finner vi cluster number density

$$n(s, p) = \frac{N_s}{L^d} \quad (40)$$

d'en her indikerer at resultatet vi har fått gjelder uansett dimensjon.

For å studere $n(s, p)$ nærmere er vi nødt til å studere dens form nærmere, for eksempel ved å generere numeriske estimer for dens skaleringsoppførsel. Fra dette kan man foreslå en generell skaleringsform, og teste denne. Området vi er interessert i er der $p \rightarrow p_c$, noe som gir $|p - p_c| \ll 1$, $s \gg 1$. I denne grensen forventer vi at s_ξ markerer crossover mellom to forskjellige oppførseler. Det er en felles oppførsel for små s , opptil s_ξ .

Formen vi foreslår er

$$n(s, p) = n(s, p_c) F\left(\frac{s}{s_\xi}\right), \quad (41)$$

$$n(s, p_c) = Cs^{-\tau}, \quad (42)$$

$$s_\xi = s_0|p - p_c|^{-1/\sigma}. \quad (43)$$

Denne skaleringsformen forenkles ofte til

$$n(s, p) = s^{-\tau} F(s/s_\xi) = s^{-\tau} F((p - p_c)^{1/\sigma} s). \quad (44)$$

Ved å plote $s^\tau n(s, p)$ som funksjon av $s|p - p_c|^{1/\sigma}$ (log10 plot) forventer vi da å få ut skaleringsfunksjonen $F(x)$, som bør være en universell kurve (se fig 4.2 s.33). Denne metoden kalles en data-collapse siden alle kurvene "kollapser" til en universalkurve. (?)

Vi kan sjekke hvordan dette passer i 1d-tilfellet, der vi har en nøyaktig løsning

$$n(s, p) = (1 - p)^2 e^{-s/s_\xi}. \quad (45)$$

Ved å bruke tilnærmingen

$$s_\xi \simeq \frac{1}{1 - p} \quad (46)$$

når p er nære $p_c = 1$, ser vi at

$$(1 - p)^2 = \frac{1}{s_\xi^2}. \quad (47)$$

Dermed får vi

$$n(s, p) \simeq s_\xi^{-2} e^{-s/s_\xi}. \quad (48)$$

Dette kan skrives om til formen

$$n(s, p) \simeq s^{-2} \left(\frac{s_\xi}{s}\right)^{-2} e^{-s/s_\xi}. \quad (49)$$

Med $\tau = 2$ ser vi at dette passer med antagelsen vår som

$$n(s, p) = s^{-\tau} F\left(\frac{s}{s_\xi}\right) \quad (50)$$

hvor vi har funnet den eksakte formen på skaleringsfunksjonen som

$$F(x) = x^\tau e^{-x}. \quad (51)$$

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 ν , and the percolation threshold, p_c ?

$\Pi(p, L)$ er sannsynligheten for at det finnes et spanning cluster i et system. For uendelig stor L vil $\Pi(p, L)$ være en step-funksjon som blir 1 i $p = p_c$. For systemer med endelig L vil Π likne mer og mer på en step-funksjon for økende L (TEGN). Vi starter med antagelsen $\Pi(p, L) = f(L/\xi)$ hvor ξ er karakteristisk lengde som vi kan skrive som $\xi = \xi_0(p - p_c)^{-\nu}$. Vi kan nå skrive

$$\Pi(p, L) = f\left(\frac{L}{\xi_0(p - p_c)^{-\nu}}\right) = f(L(p - p_c)^\nu / \xi_0) = \tilde{f}([L^{1/\nu}(p - p_c)]^\nu) = \Phi[(p - p_c)L^{1/\nu}] \quad (52)$$

Vi bestemmer oss for å finne den p -verdien som gir $\Pi(p, L) = x$. Vi kaller den $p_{\Pi=x}(L)$. Skalerings-ansatzen gir oss da

$$x = \Phi((p_{\Pi=x}(L) - p_c)L^{1/\nu}), \quad (53)$$

som gir

$$(p_{\Pi=x}(L) - p_c)L^{1/\nu} = \Phi^{-1}(x) = C_x, \quad (54)$$

$$(p_{\Pi=x}(L) - p_c) = C_x L^{-1/\nu}. \quad (55)$$

Dette kan brukes til å finne både p_c og ν . Vi velger oss to verdier for x : x_1 og x_2 .

$$dp = p_{\Pi=x_2} - p_{\Pi=x_1} = (C_{x_1} - C_{x_2})L^{-1/\nu} = AL^{-1/\nu}. \quad (56)$$

Vi kan plotte $\log dp = 1/\nu \log L + \log A$ som funksjon av $\log L$. Vi finner en tilpasset lineær funksjon og leser av stigningstallet a . Et estimat for ν blir da $\nu = -1/a$.

Nå som vi har funnet ν kan vi bruke at $p_{\Pi=x} = p_c + C_x L^{-1/\nu}$. Vi plotter denne som en funksjon av $L^{-1/\nu}$, tegner en rett linje gjennom måledataene og ser hvor denne linja krysser y -aksen og finner dermed p_c .

18. Subset 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?

Spanning clusteret kan deles inn i flere typiske subsets. Singly connected bonds beskriver de sitesene som har den egenskapen at dersom de fjernes, så vil ikke lenger clusteret være et spanning cluster (figur 8.1, s.88). Ser man på fluid-strømning er dette sites der all fluiden må passere. Et annet subset beskriver unionen av alle self avoiding walks (SAWs) som spenner fra den ene enden av clusteret til den andre. For et fysisk system av porøst materiale tilsvarende dette subsettet de hulrommene hvor en væske vil flyte fra den ene siden til den andre dersom man setter opp en trykkforskjell. Dette subsettet kalles gjerne "backbone". Resten av sitesene er da såkalte dangling ends. Dette subsettet beskriver altså de stiene som er blindveier. Clusteret vårt = dangling ends + blobs (several parallel paths) + singly connected points connecting the blobs.

Vi har sett at massen til spanning clusteret kan skrives som $M = L^D$ hvor D er fraktal-dimensjonen til spanning klusteret. Subset av spanning klusteret vil også følge slik skalering. Vi forventer at massen til de singly connected bondsene vil følge $M_{SC} = L^{D_{SC}}$. Vi vet at $M_{SC} \leq M$ som betyr $D_{SC} \leq D$. Tilsvarende også for backbone og dangling ends: $M_{BB} \propto L^{D_{BB}}$ og $M_{DE} \propto L^{D_{DE}}$.

Når vi vil måle singly connected bonds antar at vi har generert et spanning cluster og at systemet vårt kun har dette clusteret (ev fjernet de andre). Vi setter ned to walkere på den ene enden av systemet. Den ene walkern velger neste steg etter følgene regler (i prioritert rekkefølge):

- til venstre

- rett fram
- høyre
- bakover

Den andre walkeren følger samme system, der høyre og venstre er byttet om. På denne måten vil walkerne følge de ytre veiene fra den ene kanten til den andre, som vil si at alle veier vil være innenfor veiene til walkerne. Dersom begge walkerne går gjennom en site betyr det at alle veier må gå gjennom denne site. Dermed er denne site en singly connected bond. Hvis vi fjerner et singly connected bond vil ikke clusteret lenger være spanning. For å finne singly connected bonds sjekker vi om de to walkerne har vært på samme site.

Vi har sett litt på skalering i dette systemet. Vi vet at massen til spanning clusteret må tilsvare summen av massen til backbone og massen til dangling ends

$$M_{Sp} = M_{DE} + M_{BB}. \quad (57)$$

Dette kan vi fra skaleringene over skrive som

$$aL^D = bL^{D_{DE}} + cL^{D_{BB}} \quad (58)$$

19. Random walks/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.

Når man studerer konduktivitet ser man tradisjonelt på tilfeldige transistor netverk. Her måler man konduktansen, G ($I = GV$), som avhenger av systemets konduktivitet σ , en materialkonstant. for en L^d prøve i et d-dimensjonalt er da konduktansen gitt som

$$G = L^{d-1} \sigma / L. \quad (59)$$

Vi er interessert i å se på flow gjennom et porøst materiale. For en prøve av lengde L og areal A , gir Darcys lov relasjonen

$$\Phi = \frac{kA}{\eta L} \Delta p \quad (60)$$

mellom den totale fluksen, Φ og trykkforskjellen over prøven, Δp . Her er k materialets permeabilitet, og η er fluidets viskositet. Denne loven kan vi generalisere til å gjelde for et d-dimensjonalt system på følgende måte

$$\Phi = \frac{kL^{d-1}}{\eta L} \Delta p = L^{d-2} \frac{k}{\eta} \Delta p. \quad (61)$$

Fra dette ser vi at de to problemene, elektrisk konduktivitet og Darcy-flow permeabilitet, essensielt er det samme problemet.

Project 4

Som fjerde prosjekt i dette kurset har jeg valgt å studere systemet med argonatomer med interaksjoner styrt av L-J potensialet nærmere. Jeg har i denne forbindelse fokusert på termostater, og i den forbindelse sammenlignet tre forskjellige termostater.

Når vi gjør en standard, klassisk md-simulering vil systemet vi studerer tilhøre det mikrokanoniske ensemblet. Dette kommer av måten md-simuleringer gjøres på; det er energien, volumet og partikkelantallet som holdes konstant. Ofte er vi heller interesserte i å simulere et system som sampler fra det kanoniske ensemblet. I dette ensemblet er det temperaturen istedenfor energien som er en bevart størrelse, i tillegg til volum og partikkelstørrelse. Dette er variable det er enkelt å holde konstant i et fysisk eksperiment, og dermed vil vi også gjerne kunne gjøre simuleringer i dette ensemblet slik at resultatene kan sammenlignes med eksperimentelle resultater. Termostater brukes også generelt for å oppnå en ønsket temperatur i systemet, f.eks hvis man vil måle temperaturavhengige størrelser i systemet, eller en spesifikk fase.

for å kunne simulere systemer fra det kanoniske ensemblet kan vi altså ta i bruk termostater. En termostat algoritme i md-simuleringer er rett og slett en modifisering av det klassiske md-schemet som har som mål å

føre til at temperaturen i systemet holdes konstant. Dette gjøres gjerne ved å simulere kontakt med et eksternt varmebad.

En god termostat bør oppfylle visse krav. Den bør være i stand til å holde temperaturen ved et tilnærmet konstant nivå rund varmebadstemperaturen. Den bør bevare dynamikken i systemet på en god måte, og den bør føre til at systemet samler faserommet tilsvarende det kanoniske ensemblet. I tillegg bør den være justerbar.

Mange termostater simulerer interaksjonen med et varmebad ved å reskalere partikkelhastighetene på en eller annen måte. De to første termostatene vi skal se på fungerer på denne måten. Den første av disse, Berendsentermostaten, fungerer på denne måten. Ved hvert tidssteg multipliserer den hastigheten til alle partiklene med en skaleringsfaktor

$$\gamma = \sqrt{1 + \frac{\Delta t}{\tau} \left(\frac{T_{bath}}{T} - 1 \right)} \quad (62)$$

hvor τ kalles relaxation time, og det er denne som lar oss justere koblingen til varmebadet. En stor verdi for τ gir en svak kobling. Hastighetsskaleringen er dermed slik at temperaturendringen er proporsjonal med temperaturforskjellen mellom systemet og varmebadet

$$\Delta T = \frac{\Delta t}{\tau} (T_{bath} - T(t)). \quad (63)$$

Det viser seg at Berendsen termostaten ikke gir en korrekt samplig av det kanoniske ensemblet. For store nok systemer vil likevel de fleste resultatene være tilnærmet korrekte. Berendsen er likevel en populær termostat ettersom den gir en rask termalisering av systemet, og kan dermed brukes i starten av simuleringen for å få systemet termalisert til en likevektstilstand med ønsket temperatur, før man bytter til en mer korrekt termostat.

Andersentermostaten simulerer kontakt med et eksternt varmebad ved å simulere harde kollisjoner mellom partiklene i systemet og partikler i varmebadet. Dette gjøres ved at det for alle partikler trekkes en tilfeldig tall mellom 0 og 1, og hvis dette tallet er mindre enn grenseverdi $\Delta t/\tau$ vil denne partikkelen få tildelt en ny hastighet trukket fra en boltzmannfordeling med standardavvik bestemt av varmebadstemperaturen som $\sqrt{kT_{bath}/m}$. τ kan her sees på som kollisjonstiden og åpner også for denne termostaten for justering av koblingsnivået med varmebadet. Andersentermostaten vil gi en kanonisk fordeling av mikrotilstander, men er ikke-deterministisk og dermed ikke tids-ireversibel. Den har også en tendens til å forstyrre den naturlige dynamikken i systemet.

I prosjekt 4 har målet vært å implementere en mer komplisert men også bedre termostat, nemlig Nosé-Hoover termostaten og sammenligne den med de to andre. Denne termostaten simulerer effektene fra varmebadet ved å gjøre det til en del av systemet. Dette gjøres ved å inkludere det i systemets Lagrangian. Dette gjøres ved å ta i bruk en kunstig variabel \tilde{s} som knyttes til en kunstig masse $Q > 0$, og hastighet $\dot{\tilde{s}}$. For denne termostaten er det størrelsen på Q som avgjør styrken på koblingen med varmebadet. Den utvidede Lagrangianen blir

$$\mathcal{L} = \sum_i \frac{m_i}{2} \dot{\tilde{s}}^2 \dot{\mathbf{r}}_i^2 - U(\mathbf{r}) + \frac{1}{2} Q \dot{\tilde{s}}^2 - g k_b T_0 \ln \tilde{s}. \quad (64)$$

De første to termene i denne representerer henholdsvis den kinetiske og den potensielle energien til det reelle systemet. De to gjenstående leddene representerer den kinetiske og potensielle energien for \tilde{s} , og er valgt på en sånn måte at vi tilslutt oppnår at der reelle systemet samler fra det kanoniske ensemblet.

Ved å løse denne Lagrangian finner man equations of motion for systemet

$$\ddot{\mathbf{r}}_i = \frac{\tilde{\mathbf{F}}_i}{m_i \tilde{s}^2} - \frac{2 \dot{\tilde{s}} \dot{\mathbf{r}}_i}{\tilde{s}} \quad (65)$$

$$\ddot{\tilde{s}} = \frac{1}{Q \tilde{s}} \left(\sum_i m_i \tilde{s}^2 \dot{\mathbf{r}}_i^2 - g k_b T_0 \right). \quad (66)$$

Disse er tids-reversible, glatte og deterministiske. De kan uttrykkes ved systemvariablene for det reelle systemet gjennom substitusjonene

$$s = \tilde{s}, \quad \dot{s} = \dot{\tilde{s}}, \quad \ddot{s} = \ddot{\tilde{s}} + \frac{\dot{\tilde{s}}^2}{\tilde{s}} \quad (67)$$

$$\mathbf{r} = \tilde{\mathbf{r}}, \quad \dot{\mathbf{r}} = \dot{\tilde{\mathbf{r}}}, \quad \ddot{\mathbf{r}} = \ddot{\tilde{\mathbf{r}}} + \frac{\dot{\tilde{s}}^2}{\tilde{s}} \tilde{\mathbf{r}}. \quad (68)$$

Det gir

$$\ddot{\mathbf{r}}_i = \frac{\mathbf{F}_i}{m_i} - \gamma \dot{\mathbf{r}}_i \quad (69)$$

$$\dot{\gamma} = \frac{-3Nk_B}{Q} T(t) \left(\frac{T_0}{T(t)} - 1 \right) \quad (70)$$

hvor $\gamma = \frac{\dot{s}}{s}$.

Fra dette ser vi at denne termostaten kan simuleres ved å legge til leddet $-\gamma \dot{\mathbf{r}}_i$ i akselerasjonen til partikkel i . Da trenger man γ og denne kan finnes fra uttrykket for $\dot{\gamma}$ ver f.eks en finite difference.

Når man forsøker å implementere dette i verlet algoritmen ser man fort at man får et problem. dette er fordi man i utregningen av farten ved neste tidssteg trenger nettopp farten ved neste tidssteg gjennom leddet $-\gamma \dot{\mathbf{r}}_i$. Dette kan løses på flere måter. Jeg valgte å heller bruke farten et hlat tidssteg fram, istedenfor et helt som tilnærming. Denne tilnærmingen kan forbedres ved f.eks å kjøre noen picarditerasjoner.