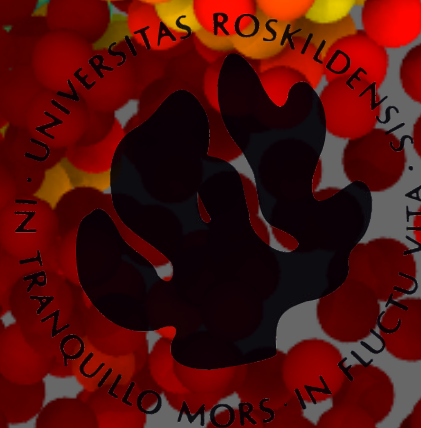


Kritiske Overgang til Klynger af Havfruepartikler

Udarbejdet af:
Mikkel Sakse Bennetsen

Vejleder:
Ulf Rørbæk Pedersen

Roskilde Universitet
Speciale i Fysik
14. november 2020



Abstract

Mermaid particles, also called SALR particles, are particles that attract each other when close, and repulse each other when farther away. Simulations made with this kind of interaction are usually made to explain the behaviour of clustering colloids. This work realises the SALR condition by combining a positive and a negative Yukawa potential and a Barker-Henderson corrected inverse power potential as the hard sphere. This potential is implemented into the molecular dynamics simulation software RUMD.

Simulations in equilibrium are used to obtain the size distribution of clusters while the density of free particles ρ_{fri} is measured in expansion and compression simulations. ρ_{fri} has a maxima which is used as a indicator of the critical clustering density. It is found that expansion simulations under estimate ρ_{fri} while compression simulations over estimate it. The maximas found in these simulations are then used as the range in which the true critical clustering density (CCD) must be in. In earlier work by Sweatman and Lue a phase diagram for the investigated pair potential was set up using thermodynamic calculations [1]. The phase diagram depicts A_a vs density, where A_a is the amplitude of the attractive Yukawa potential. According to the results in this work the CCD-line is found to be much lower then what is depicted in the phase diagram.

Resumé

Havfruepartikler, også kaldet SALR-partikler, er partikler som tiltrækker hinanden når de er tætte, og frastøder hinanden når de er længere væk. Simulering af systemer med denne slags interaktioner er som regel lavet for at forklare klyngedannende kolloider. Dette arbejde realiserer SALR forholdene ved at kombinere et positivt og et negativt Yukawa potentiale og et Barker-Henderson korrigeret invers potens potentiale som den hårde sfære. Dette potentiale bliver så implementeret i molekylær dynamik software pakken RUMD.

Ligevægtssimuleringer bliver brugt til at få størrelsesfordelingerne af klynger, mens densiteten af de frie partikler ρ_{fri} er målt i ekspansions- og kompressionssimuleringer. ρ_{fri} over ρ grafen har et maxima som kan bruges til at indikere den kritiske klyngedannelsesdensitet. Det blevet opdaget at ekspansions- og kompressionssimuleringer underestimerer ρ_{fri} , mens kompressionssimuleringer overestimerer den. Maksimaerne i disse simuleringer bliver så brugt som spændet hvori den sande klyngedannelsesdensitet må befinde sig i. I tidligere arbejde af Sweatman og Lue et er der tegnet et fasediagram for det undersøgte parpotentiale som er udarbejdet ud fra termodynamiske beregninger [1]. Fasediagrammet viser A_a , som er amplituden af den attraktive del af potentialet, på første-aksen og densitet på andenaksen. Resultaterne fra dette arbejde indikerer at linjen for den kritiske klyngedannelsesdensitet er væsentligt lavere end hvad der er vist på fasediagrammet.

Indhold

1	Problemformulering	4
2	Indledning	5
3	Baggrund	7
4	Simulering	11
4.1	Parpotentialet	11
4.2	Implementering af Langevin Dynamik i Leapfrog	16
5	Modeller	18
5.1	Densiteten af frie partikler	18
5.2	Størrelsesfordeling af Klynger	22
6	Resultater	26
6.1	Bestemmelse af Cutoff i Klyngeanalyse	26
6.2	Ligevægtsimuleringer	28
6.3	Simuleringer med hysteresse	31
7	Diskussion	37
8	Opsummering	38
	Litteratur	39
A	Kompressionssimulering	41
B	Ekspansionssimulering	44

1 Problemformulering

Hvordan kan den kritiske klynge densitet for havfruepartikler bedst estimeres? Kan dette gøres ved ekspansions- og kompressionssimuleringer med hysteresis?

2 Indledning

I dette teoretiske speciale undersøges såkaldte SALR potentialer. SALR står for: ‘**S**hort range **A**ttraction, **L**ong range **R**epulsion’, som betyder korttrækkende tiltrækning, langtrækkende frastødning. Det bliver desuden også kaldet et ‘Mermaid potential’ som betyder havfrue potentialer. Navnet kommer af at en havfrue har et tiltrækkende frontparti og et frastødende bagparti [2]. Hvis partikler har denne interaktion med hinanden giver det mulighed for klyngedannelse under de rette forhold. Her kommer havfrue sammenligningen til kort da man må forestille sig at havfruer ikke ser hinandens haler som frastødende.

Hvorfor er disse klynger så, så interessante? Jo det viser sig at det faktisk ikke er særlig svært at finde denne slags interaktioner i naturen. Et eksempel fra fysikken er protoner der kan binde sig til hinanden ved den stærke kernekraft, men frastødes også på grund af den elektrostatiske kraft. I dette tilfælde er klyngerne meget ustabile og kræver neutroner, der kun har den stærke kraft, for at facilitere klyngedannelsen. På nanometer længdeskalaen, findes polymerer og kolloider som kan have havfrue egenskaber. Proteiner der har en overordnet ladning har en skærmet coulomb interaktion til hinanden (Yukawa potentialet), men når de er tæt på hinanden vil der være Van der Waals interaktioner som kan holde dem sammen. Liquid-liquid (væske-væske) separation er en almindelig forklaringsmodel for membranløse organeller [3]. Det er dog blevet foreslået at den spontane klyngedannelse gennem SALR kan bruges til at forklare liquid-liquid separationen [4]. Peptider med et kombineret Van der Waals og Yukawa potentiale ville muligvis kunne danne viskøse dråber i cytoplasmaen. Lysozym er eksempelvis et protein der kan lave liquid-liquid separation og simuleringer af denne stemmer godt overens med eksperimentielle resultater [5,6]

En hypotese for hvordan livet startede er at det startede med den spontane dannelse af lipidmembraner, men det er blevet foreslået at en primitiv membranløs organel kunne være det der faciliterede de første katalytiske processer [4].

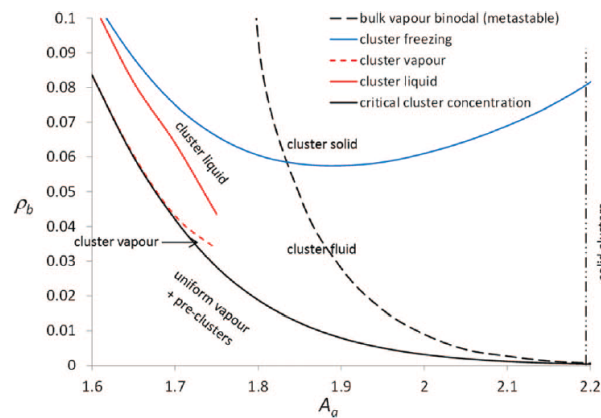
Globulare proteiner egner sig godt til molykylær dynamisk (MD) simulering da disse kan approksimeres som sfæriske kugler. Der er derfor lavet simule-

ringer med isotropiske sfæriske partikler med SALR parpotentialer. Dette speciale bygger videre på teoretiske undersøgelser med udgangspunkt i resultater fra Sweatman og Lue [1]. Der er tages udgangspunkt i potentialet de bruger. Til forskel for arbejdet af Sweatman og Lue bruger dette speciale dynamiske Langevin simuleringer og ikke Monte Carlo simuleringer. Til dette bruges software pakken RUMD som kan udfører særligt effektive beregninger ved brug af grafik kort [7]

3 Baggrund

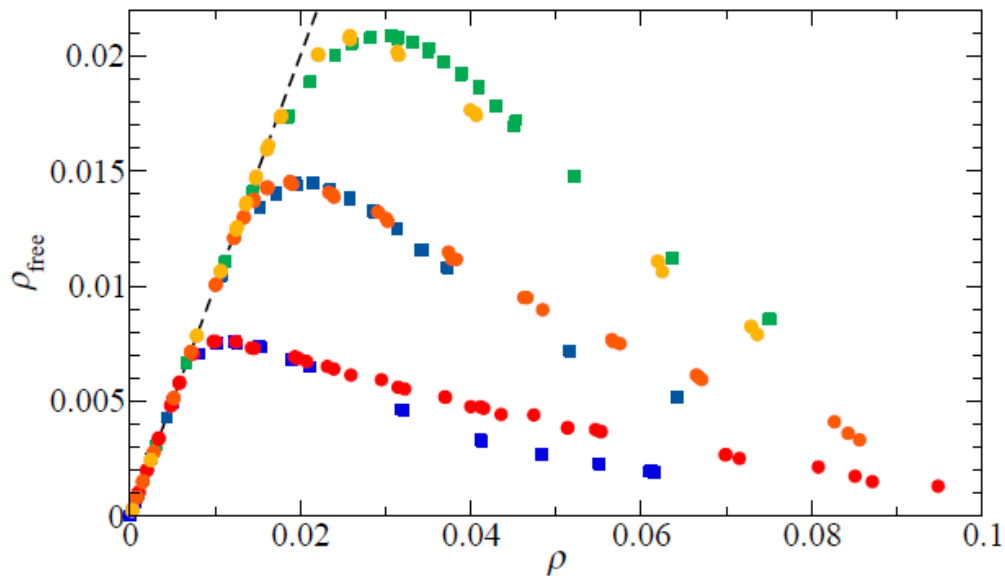
Som nævnt i indledningen tages der i dette arbejde udgangspunkt i Sweatman og Lue's arbejde. De har udarbejdet et fasediagram for SALR-partiklerne der arbejdes med. Fasediagrammet er lidt specielt. På andenaksen er der densiteten og på førsteaksen er der noget der hedder A_a . Dette er amplituden på det tiltrækkende potentiale. Denne afgør derfor hvor godt partiklerne kan klistre til hinanden.

Fasediagrammet kan ses på Figur 3.1 [1]. Fasediagrammet har forskellige linjer der indikerer faseovergange. Den fuldtoptrukkede sorte linje indikerer den kritiske klyngedannelsesdensitet. Det er her at systemet går fra at være en gas med præklynger. Præklynger er små ustabile klynger der nogen gange bliver dannet i gassen. Det er denne førsteordens faseovergang der vil blive undersøgt her i arbejdet gennem simulering. Når der så er dan-



Figur 3.1: Her ses fasediagrammet for den partikel arbejdet her undersøger. De forskellige linjer indikerer faseovergange i systemet. Den sorte linje er den kritiske klyngedannelsesdensitet som funktion af A_a -parameteren. Den røde stiplede linje er klyngegasovergangen. Den hele røde linje er klyngevæskeovergangen. Den blå linje indikerer hvor der begynder at dannes klyngekrystaller. Den lodrette prikkede linje indikerer hvor partiklerne i klyngerne begynder at krystalisere. Figuren er fra Sweatman og Lue [1]

net stabile klynger sker der nogle mystiske faseovergange. Disse faser kaldes mesofaser. Når systemet skifter mesofase ændrer den individuelle partikels



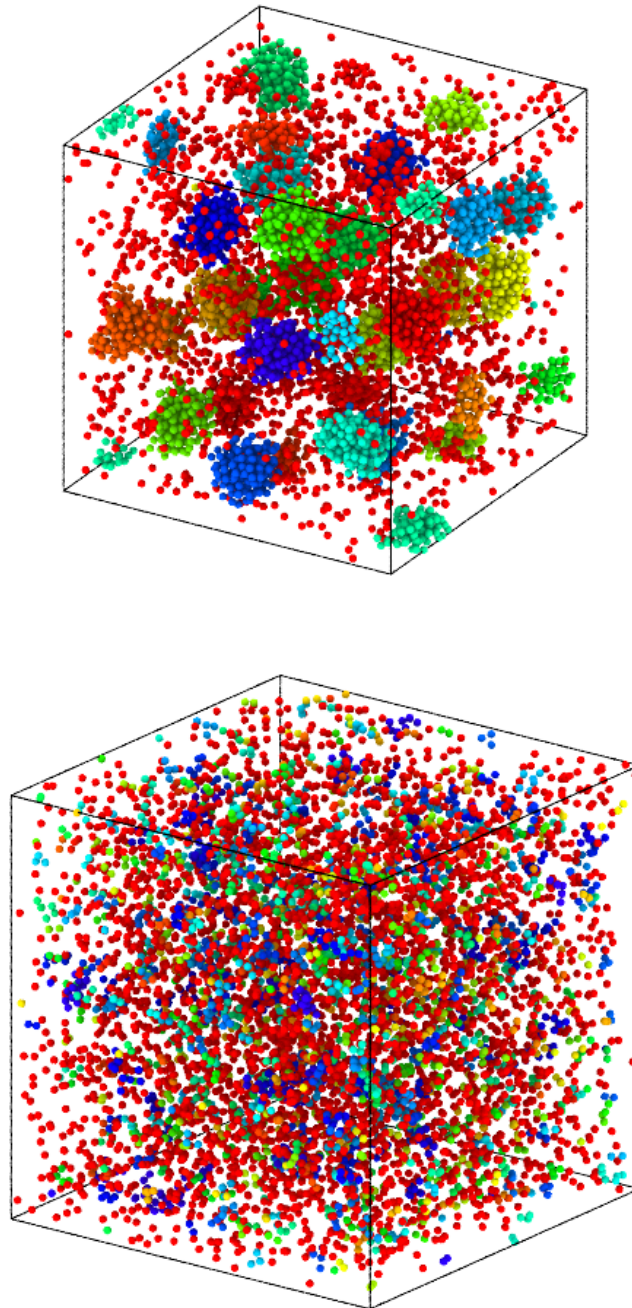
Figur 3.2: Figuren viser ρ_{free} for systemer med to forskellige potentialer og forskellige temperaturer. De røde til gule punkter har et potentiale hvor den tiltrækkende kraft er stærkere end den er hos de blå til grønne punkter. Toppunktet på disse grafer kan bruges som indikator for den kritiske klyngedannelsesdensitet.

adfærd sig ikke betydeligt, men det gør klyngerne til gengæld. Klyngerne kan faktisk opføre sig som om de var partikler. Det vil sige at der kan være en mesofase hvor klynger interagerer som var de partikler i en gas, som var de partikler i en væske og som var de partikler i en krytal. Der kan sågar være en koeksistens af disse mesofaser [8]. Den røde stiplede linje indikerer gas-koeksistensovergangen for mesofaserne. Den hele røde linje indikerer koeksistens-væskeovergangen. Den blå linje indikerer fluid-solidovergangen. Det er altså her at klyngerne begynder at lægge i gitter struktur. Den lodrette prikkede linje til højre indikerer der hvor partiklerne i klyngerne begynder at krystalisere.

Når den kritiske klyngedannelsesdensitet skal findes kan det være en god ide at kigge på antallet af frie partikler divideret med volumet. Denne størrelse kaldes i Santos et al. for ρ_{free} . Det der sker er nemlig at lige så snart at partikler begynder at samle sig i klynger falder ρ_{free} drastisk og det gør den faktisk monotonisk [9]. I Figur 3.2 er der resultater fra simuleringer med en anden

type SALR partikel. Resultatet viser her hvordan ρ_{free} stiger direkte proportionelt med den overordnede densitet, indtil den pludselig laver et knæk og herefter falder. I Santos et al. bruger de maksimummet på denne kurve som indikator på den kritiske klyngedannelsesdensitet.

Tidlige resultater viser at der bliver dannet fine klynger i de molekylær dynamiske simuleringer. for at læser kan få en fornemmelse for hvordan et system med havfrue klynger ser ud er der i Figur 3.3 vist billeder af et system uden klynger versus et system med klynger



Figur 3.3: Det øverste billede viser et SALR system med klynger. Det nederste billede viser et SALR system uden klynger. Farverne indikerer klyngeindeks.

4 Simulering

4.1 Parpotentialet

I en MD simulering har man en kollektiv positionsvektor der er defineret ved

$$\vec{R} = (\vec{r}_1, \vec{r}_2, \dots, \vec{r}_n) \quad (4.1)$$

Hvor \vec{R} er en $3N$ vektor, og \vec{r}_n er vektorer for hver partikel, der indeholder deres koordinater i det tredimensionale rum.

$$\vec{r}_n = (x_n, y_n, z_n) \quad (4.2)$$

Et parpotentiale er en funktion af afstand mellem to partikler som udtrykker den potentielle energi imellem dem. Det anvendte parpotentiale ser ud på denne måde

$$u(r) = u_{sr}(r) + u_{SALR}(r) \quad (4.3)$$

Her er $u(r)$ den potentielle energi som funktion af afstanden mellem to partikler r . $u_{sr}(r)$ og $u_{SALR}(r)$ er henholdsvis parpotentialet for en approksimeret hård kugle og et SALR parpotentiale. Når parpotentialet er defineret kan der opstilles en potentiale funktion for konfigurationen

$$U(\vec{R}) = \sum_{n=1}^{N-1} \sum_{m>n}^N u(|\vec{r}_m - \vec{r}_n|) \quad (4.4)$$

hvor $U(\vec{R})$ er en funktion der beskriver den potentielle energi i det tredimensionelle rum ud fra den kollektive positionsvektor og hvor N er det samlede antal partikler. Ud fra den potentielle energi kan kraftpåvirkningen beregnes som

$$\vec{F} = -\nabla U(\vec{R}) \quad (4.5)$$

hvor ∇ er gradienten på den $3N$ dimensionelle energy flade $U()$. Kraftvektoren for hver partikel kan så skrives op som

$$\vec{F}_n = -\nabla_n U(\vec{R}) \quad (4.6)$$

hvor ∇_n er det partielle differentiale for koordinaterne i vektoren r_n .

$$\nabla_n = \left(\frac{\partial}{\partial x_n}, \frac{\partial}{\partial y_n}, \frac{\partial}{\partial z_n} \right) \quad (4.7)$$

Den resulterende kraft for en individuel partikel kan også skrives op som en sum af kraftpåvirkningerne fra andre partikler.

$$\vec{F}_n = \sum_{n \neq m}^N \vec{F}_{nm} \quad (4.8)$$

hvor \vec{F}_{nm} kan findes med parpotentialet

$$\vec{F}_{nm} = -\frac{du(r_{nm})}{dr_{nm}} \hat{r}_{nm} \quad (4.9)$$

hvor r_{nm} er længden mellem to partikler, og \hat{r} er enhedsvektoren for $\vec{r}_m - \vec{r}_n$

$$\hat{r}_{nm} = \frac{\vec{r}_m - \vec{r}_n}{|\vec{r}_m - \vec{r}_n|} \quad (4.10)$$

Et SALR parpotentiale kan realiseres på mange forskellige måder. De to mest almindelige metoder er at kombinere Lennard-Jones og et Yukawa potentiale eller, som det bliver gjort i denne rapport, at kombinere et positivt og et negativt Yukawa-potentiale [10]. Så længe at det har et minima, der er mindre end nul ved korte afstande, efterfulgt af et maxima ved længere afstande, defineres den stadig som et SALR parpotentiale. I tidligere arbejde [4, 8], som dette arbejde tager udgangspunkt i, er det kombinationen af to Yukawa-potentialer der bliver benyttet

$$u_{\text{SALR}}(r) = -\frac{A_a}{r} e^{-(r-1)z_a} + \frac{A_r}{r} e^{-(r-1)z_r} \quad (4.11)$$

A_a og A_r er amplituder for henholdsvis det tiltrækkende og det frastødende potentiale og ved afstanden 1 bliver potentialet lig med $A_r - A_a$. z_a og z_r afgør henholdsvis det tiltrækkende og det frastødende potentiales aftagelse. dvs at de afgør hvor langtrækkende potentialerne er. Jo større z er jo mere korttrækkende bliver den respektive del af potentialet. For at denne model opfylder SALR definitionen kræver det at $A_a > A_r$ og at $z_a > z_r$. Hvis A_a var mindre end A_r ville det frastødende potentiale overdøve det tiltrækkende og parpotentialet ville derfor ikke være mindre potentiale når partikler er tætte på hinanden end når de var langt fra hinanden.

Dette potentiale går mod minus uendelig ved afstanden nul hvilket er vist med en stiplet linje på figur 4.2. Siden partiklerne ikke skal overlappe med

hinanden skal der tilføjes noget stærkt frastødende når afstanden bliver mindre end partiklernes diameter. Dette kan gøres med potentialet for den hårde kugle. I Sweatman's arbejde [4, 8] er dette blevet gjort ved at sige at når afstanden mellem to partikler kommer under d bliver potentialet uendeligt. Dette kan realiseres ved at tilføje en modificeret Heaviside trinfunktion

$$u_{\text{MC}}(R) = \left(\frac{1}{H(r-d)} - 1\right) + u_{\text{SALR}}(r) \quad (4.12)$$

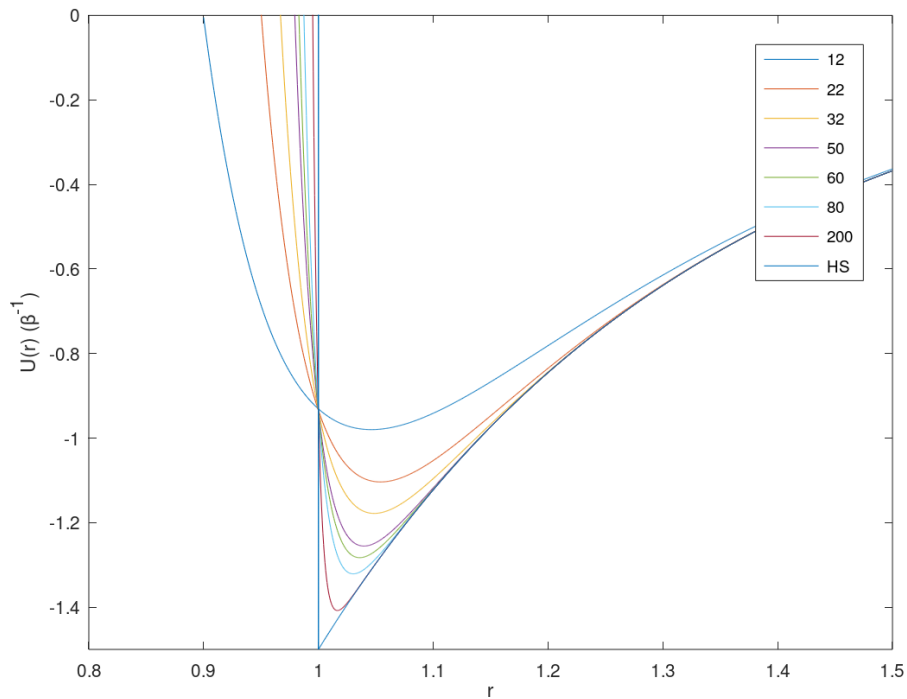
Dette er bekvemt når der benyttes Monte Carlo simuleringer (heraf subskriberet 'MC') hvilket Sweatman's beregninger er baseret på. Her bruges en kontinuerlig molekylær dynamik og algoritmen ville tillade partikler at overskride denne grænse hvilket ville medføre at to partikler ville støde fra hinanden med uendelig kraft. Dette problem er blevet løst ved at erstatte funktionen for den hårde kugle med en invers potensfunktion

$$u_{sr}(r) = d/r^n \quad (4.13)$$

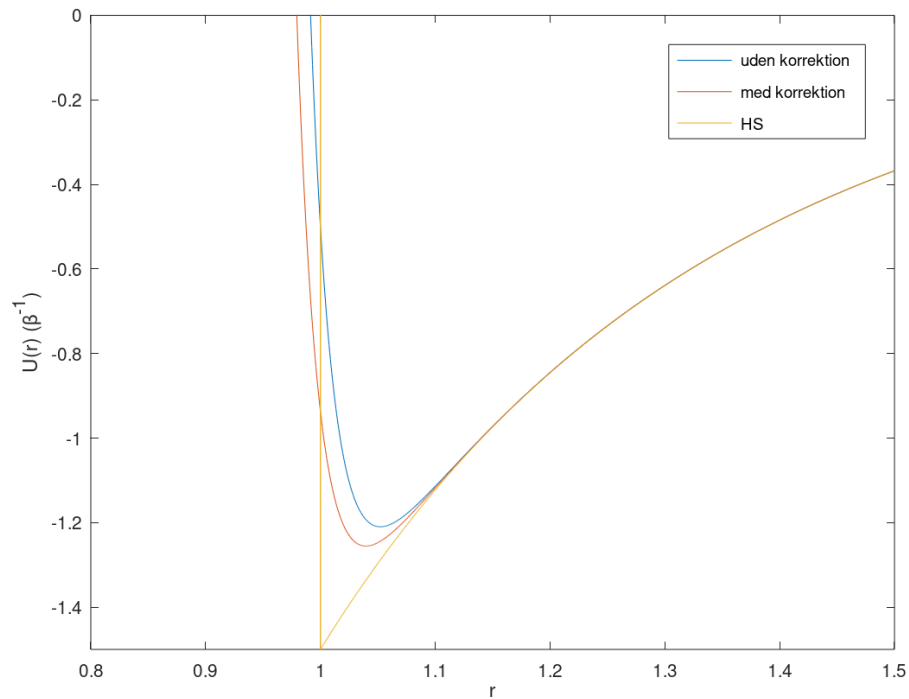
Hvor d er partiklens radius. Dette bliver lagt til potentialeudtrykket. Jo højere eksponenten er, des stejlere bliver den hårde skal og tilnærmer sig den hårde kugle. Dette kræver også mindre tidsskridt da sandsynligheden for at partikler overskrider hvad der er fysisk tilladt bliver større. En konsekvens af at bruge et invers potenspotentiale bliver at parpotentialet nær $r=d$ ikke ligner Sweatman et al. særlig meget. Det får ikke den samme spids og minimumet ligger højere som det ses på figur 4.1.

Et blødere potentiale er formentlig et mere realistisk parpotentiale, men det giver nok anderledes resultater end det der ses i Sweatmans arbejde.

Man kan tilnærme sig den hårde kugle med en Barker-Henderson korrektion [11] som skubber det inverse potens potentiale tættere på nul jo mindre potensen er. Dette gør at det samlede parpotentiales minima bliver mindre og det gør også at partiklerne kan komme tættere på hinanden. Figur 4.2 viser forskellen på par-potentialet uden Barker-Henderson korrektionen, med korrektionen og potentialet brugt i monte carlo simuleringer. Korrektionen gør ikke at potentialet kommer til at ligne monte carlo simuleringen mere, men tanken er at den ekstra plads partiklen kan bevæge sig på, kompenserer for den mindskelede bindingskraft.



Figur 4.1: Her ses det samme SALR parpotentiale, med parameterne $A_a = 2$, $A_r = 0.5$, $z_a = 1$, $z_r = 0.5$ og $d = 1$, kombineret med Barker-Henderson korrigeret invers potenspotentialer med potens af forskellige størrelser, som kan ses i ligning 4.11. Billedteksten indikerer hvilken potens er blevet brugt til hver graf. Den nederste graf viser funktionen i ligning 4.12 og viser parpotentialet for en SALR partikel med en hård kugle. Det ses at jo højere potensen bliver des mere lig potentialet for den hårde kugle bliver det



Figur 4.2: figuren viser henholdsvis parpotentialet med parametrene $A_a = 2.0$, $A_r = 0.5$, $z_a = 1.0$, $z_r = 0.5$ og $n = 50$, med og uden Barker-Henderson korrektion. Det sidste er parpotentialet med potentialet for den hårde kugle i stedet for et invers potens potentiale

4.2 Implementering af Langevin Dynamik i Leapfrog

RUMD benytter en Leapfrog algoritme. Hvad er Leapfrog? Det er en metode til at numerisk integrere differentialligninger der beskriver newtonske bevægelser. Det er en variation af Verlet integration som benytter at sekanten imellem to punkter er et godt approksimat for hældningen ved midterpunktet. I en simulering der benytter Leapfrog har hver partikel en hastighed. Denne hastighed ændrer sig alt efter hvad den resulterende kraft er. Denne kraft bruges til at beregne hvad den nye hastighed bliver efter et halvt tidsskridt.

$$\vec{v}_{i+1/2} = \vec{v}_{i-1/2} + \frac{\vec{F}_i}{m} dt \quad (4.14)$$

Hvor $\vec{v}_{i-1/2}$ og $\vec{v}_{i+1/2}$ henholdsvis er hastigheden et halvt tidsskridt før og efter tiden for partiklens position og m er partiklens masse. Med halvskridthastigheden bliver der så beregnet en ny position

$$\vec{r}_{i+1} = \vec{r}_i + \vec{v}_{i+1/2} dt \quad (4.15)$$

den simple udgave af Leapfrog algoritmen kan bruges til at simulere et deterministisk system hvor partiklers bevægelse udenlukkende er bestemt ud fra potentialet. For systemer der er mere stokastiske har RUMD implementeret Langevin dynamik ind i Leapfrog. Langevin fungerer som en model af et implicit opløsningsmiddel. Med implicit menes der at man ser på opløsningsmidlets overordnede effekt på partiklerne. Det står i kontrast til en explicit model der ville modellere opløsningsmidlet som individuelle partikler. implementeringen er gjort som beskrevet i Grønbech et al. [12].

Langevin dynamik kan i en dimension beskrives ved følgende ligning.

$$m\ddot{x}_i = F_{i,x} - \gamma\dot{x}_i + \sqrt{2\gamma k_B T} R(t) \quad (4.16)$$

hvor x er et element i den kollektive \vec{R} vektor, γ er friktionskoefficienten, k_B er Boltzmann konstanten, T er temperaturen, og $R(t)$ er en gaussisk proces der opfylder følgende krav

$$\langle R(t) \rangle = 0 \quad (4.17)$$

$$\langle R(t)R(t') \rangle = \delta(t - t') \quad (4.18)$$

Langevin ligningen har udover kraften givet ved potentialet en modstandskraft der er proportionel til hastigheden og et led der er der smider en

stokastisk kraft ind i systemet. Tanken er at dette skal modellere interaktionen med en solvent. Partikler der bevæger sig igennem et medie vil opleve en modstandskraft alt efter viskositeten som ved lave hastigheder er proportionel med hastigheden. Dette kaldes også Stokes' modstand. Udover det kan mediet også skubbe til partiklen i hvilken som helst retning som modelleres ved det sidste kraftled. størrelsen af denne tilfældige kraft afhænger af temperaturen, da dette har betydning for hvor hårdt mediet kan slå til partiklen, og det afhænger desuden af γ da dette har betydning for hvor meget partiklen er forbundet til varmebadet.

med Langevin dynamik kommer de numeriske beregninger for halvskridtshastigheden og positionen som udledt i Grønbech et al. til at se ud som følger

$$v_{i+1/2} = av_{i-1/2} + \frac{F_i}{m}dt + \frac{1}{2m}\sqrt{2\gamma k_B T}R_i \quad (4.19)$$

$$\vec{r}_{i+1} = \vec{r}_i + b \cdot dt \cdot v_{i+1/2} + \frac{bdt}{2m}\sqrt{2\gamma k_B T}R_{i+1} \quad (4.20)$$

hvor

$$a = \frac{1 - \frac{\gamma dt}{2m}}{1 + \frac{\gamma dt}{2m}} \text{ og } b = \frac{1}{1 + \frac{\gamma dt}{2m}}$$

Første halvskridtshastighed bliver beregnet ud fra ligning 4.14 først herefter bliver Langevin implementeret i skridtene. Når systemet skal evalueres bliver det bliver det nødvendigt at kigge på helskridtshastigheden, som regnes således

$$v_i = \frac{1}{2}(v_{i+1/2} + v_{i-1/2}) + \frac{1}{4m}\sqrt{2\gamma k_B T}R_i \quad (4.21)$$

denne hastighed kan bruges til at beregne den kinetiske energi.

5 Modeller

5.1 Densiteten af frie partikler

Her hvor grafen har toppunkt kan bruges til at pejle et kritisk punkt hvor vi har vores overgang [10]. Jeg vil forelå at et andet punkt bliver brugt som det kritiske punkt. grunden til dette burde blive åbenlyst efter at min model for koncentrationen af frie partikler bliver opstillet. modellen skal beskrive to udviklinger af koncentrationen og hvordan overgangen fra den ene til den anden sker. Under den kritiske koncentration er forventningen at alle partikler er frie. Det kan ske at der dannes en klynge som hurtigt fordamper igen, koncentrationen af frie partikler bør derfor være den samme som bulk koncentrationen. funktionen ser således ud.

$$\rho_{\text{fri}}(\rho) = \rho \quad (5.1)$$

dette er ret simpelt og intuitivt. Den anden del af modellen kræver mere argumentation og et par antagelser. En førsteordens faseovergang fra gas til vand kan gå igennem en koeksistens fase. Klyngefasen i vores system minder umiddelbart meget om en koeksistensfase hvor klyngerne er væskefasen og de frie partikler er gasfasen. Hvis densiteten ændres, i denne fase, mens at temperaturen holdes konstant, bliver densiteten for hver fase ikke ændret. Til gengæld går flere partikler i væskefasen hvis densiteten bliver større og flere partikler går i gasfasen hvis densiteten bliver mindre. Kun størrelsen af hver fase ændrer sig.

$$\rho_l = \frac{N_l}{V_l} \quad (5.2)$$

hvor ρ_l er densiteten af væskefasen, N_l er antal partikler der befinder sig i væskefasen, og V_l er volumen af væskefasen. herefter skrives de samme forhold op for gasfasen

$$\rho_g = \frac{N_g}{V_g} \quad (5.3)$$

hvor ρ_g er densiteten af gasfasen, N_g er antal partikler der befinder sig i gasfasen, og V_g er volumenet af gasfasen. Det samme forhold kan desuden sættes op for hele kassen

$$\rho = \frac{N}{V} \quad (5.4)$$

Det totale volumen og antal af partikler er lig med den ene fase lagt sammen med den anden

$$V = V_g + V_l \quad (5.5)$$

$$N = N_g + N_l \quad (5.6)$$

Ved at lægge formel 5.2, 5.3 og 5.6 opstilles følgende forhold.

$$N = V_g \rho_g + V_l \rho_l = V_g \rho_g + (V - V_g) \rho_l = V_g (\rho_g - \rho_l) + V \rho_l \quad (5.7)$$

først stilles forholdet op, derefter substitueres V_l med dens definition ifølge formel 5.5. nu kan vi substituere V_g med definitionen i formel 5.3.

$$N = N_g \frac{\rho_g - \rho_l}{\rho_g} + V \rho_l \quad (5.8)$$

nu isoleres N_g

$$\frac{\rho_g}{\rho_g - \rho_l} (N - V \rho_l) = \frac{\rho_g}{\rho_l - \rho_g} (V \rho_l - N) = N_g \quad (5.9)$$

for at forsimpler udtrykket kan vi k defineres.

$$k \equiv \frac{\rho_g}{\rho_l - \rho_g} \quad (5.10)$$

I dataen bruges en størrelse kaldet ρ_{fri} . dette er en lidt usædvanlig størrelse da det ikke er densiteten af gasfasen men antallet af frie partikler (gasparkler) divideret med det totale volumen.

$$\rho_{\text{fri}} = \frac{N_g}{V} \quad (5.11)$$

Formel 5.9 er et udtryk for antallet af gaspartikler så hvis vi dividerer med volumen får vi et udtryk for ρ_{fri} .

$$\rho_{\text{fri}} = \rho_l k - \frac{N}{V} k = \rho_l k - \rho k \quad (5.12)$$

Siden at k kun afhænger af ρ_g og ρ_l så beskriver formel 5.12 en lineær sammenhæng mellem ρ_{fri} og ρ så længe ρ_g og ρ_l er konstante.

Hvis k blev fundet til at være positiv og konstant giver det kun fysisk mening at dette gælder i intervallet $\rho_g < \rho < \rho_l$. Hvis ρ er mindre end ρ_g bliver

ρ_{fri} større end ρ og hvis ρ er større end ρ_l bliver ρ_{fri} negativ, hvis k forblev konstant.

I scopet af dette arbejde kommer der ikke til at blive set på overgangen fra koeksistens til 'væskefasen' eller der hvor at alt gassen er kondenseret, men i figur 5.1 er der opstillet en skitse for hvordan ρ_{fri} udvikler sig i de forskellige faser og i teorien hvis systemet bliver ved med at opføre sig som en klassisk koeksistens skulle ρ_{fri} falde linært endtil $\rho = \rho_l$. før det, starter ρ_{fri} med at følge udviklingen i formel 5.1 hvorefter den laver en overgang og følger udviklingen i formel 5.12

Så jeg har nu sat to modeller op der beskriver ρ_{fri} i to forskellige intervaller. Dette kan beskrives på følgende måde.

$$\frac{d\rho_{\text{fri}}}{d\rho} = \begin{cases} 1, & \rho < \rho_g \\ -k, & \rho > \rho_g \end{cases} \quad (5.13)$$

Hvis overgangen her var ideel kunne den afledte funktion fittes til en step-funktion, men virkeligheden er sjældent ideel. For at få et bedre fit kan en sigmoid funktion bruges da den tillader en blødere overgang. Jeg har valgt at bruge en logistisk funktion. det gælder at i grænsen hvor den logiske funktion har en uendelig vækstrate, er funktionen lig med en step-funktion

$$\lim_{b \rightarrow \infty} \frac{a}{1 + e^{-b(x-x_0)}} = \begin{cases} 0, & x < x_0 \\ a, & x > x_0 \end{cases} \quad (5.14)$$

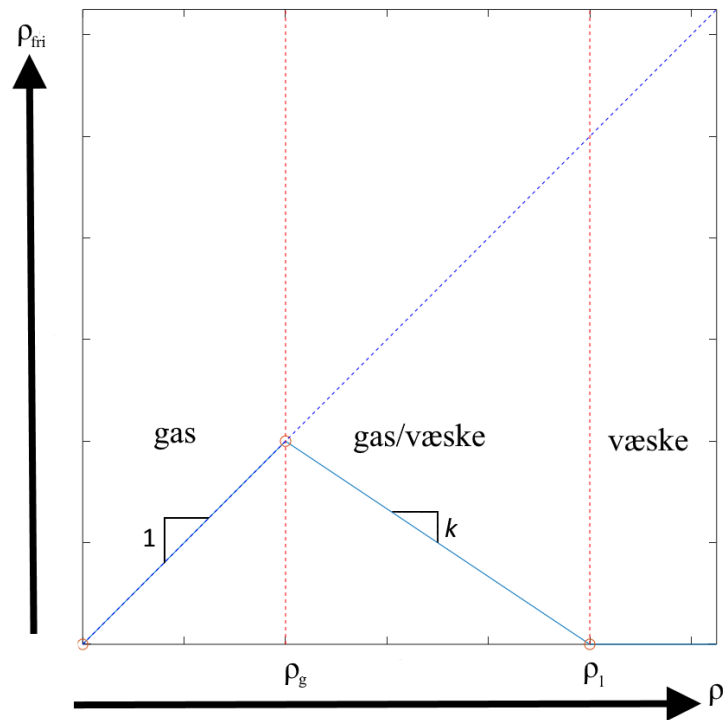
a beskriver både differencen mellem asymptoterne, b beskriver vækstraten, og x_0 beskriver hvornår overgangen sker, helt præcist der hvor den afledte af den logistiske funktion har et ekstremumpunkt. i vores model kan a defineres som

$$a = (-k) - 1 = -(k + 1) \quad (5.15)$$

vi bruger ρ som den uafhængige variable og som ses på figur 5.1 så er ρ_g der hvor overgangen finder sted

$$x = \rho \quad (5.16)$$

$$x_0 = \rho_g \quad (5.17)$$



Figur 5.1: skitse over hvordan ρ_{fri} som funktion af ρ i teorien skulle se ud hvis klyngefasen fulgte en klassisk koeksistensfase. Henad x-aksen aflæses den totale densitet, op af y-aksen aflæses densiteten af ρ_{fri} . Den stiplede diagonal viser hvordan ρ_{fri} ville udvikle sig hvis der ikke var en faseovergang. Den solide linje viser hvordan koncentrationen af frie partikler udvikler sig i henholdsvis gas-, koeksistens- og væskefasen. hældningen i gasfasen er lige med 1 fordi her antages det at $\rho_{\text{fri}} = \rho$ i koeksistensfasen antages det at hældningen er lig med k som er defineret i formel 5.10. I væskefasen ville hældningen være lig 0 da der ikke ville være flere frie partikler tilbage i systemet her. De lodrette stiplede linjer markerer hvor faseovergangen finder sted. ρ_g er desuden ifølge 5.12 densiteten af gasfasen i koeksistens fasen og ρ_l er densiteten i væskefasen og derfor også densiteten i klyngerne

funktionen starter som standard i nul men vi vil gerne have at den starter i 1 så der plusses 1 til funktionen så vi ender med at den logiske funktion ser ud således

$$\frac{d\rho_{\text{fri}}}{d\rho} \approx \frac{-(k+1)}{1+e^{-b(\rho-\rho_g)}} + 1 \quad (5.18)$$

Der er hermed komme en ny parameter b på.

Nu skal funktionen integreres således at der dannes et udtryk for ρ_{fri} .

$$\rho_{\text{fri}} = \int \left(\frac{-(k+1)}{1+e^{-b(\rho-\rho_g)}} + 1 \right) d\rho = -k\rho - \frac{(1+k) \cdot \log(1+e^{-b(\rho-\rho_g)})}{b} \quad (5.19)$$

dette bliver funktionen som bliver brugt som regressionsmodel når $\rho_{\text{fri}}(\rho)$ plottes.

Ud fra denne model kan der også findes et toppunkt for ρ_{fri} . Det er relevant at finde dette toppunkt da dette kan bruges som en definition på den kritiske klyngedannelsesdensitet. Hvis der kigges på formel 5.18 skal denne simpelthen bare sættes lige med nul, da dette er den afledte af ligning 5.19. Hvorefter ρ bliver isoleret. denne ρ kaldes ρ_t og er den densitet hvor at $\rho_{\text{fri}}(\rho)$ har sit toppunkt.

$$\frac{-(k+1)}{1+e^{-b(\rho_t-\rho_g)}} + 1 = 0 \quad (5.20)$$

$$\rho_t = \rho_g - \frac{\log(k)}{b} \quad (5.21)$$

5.2 Størrelsesfordeling af Klynger

Som nævnt i det tidligere afsnit så ses der to faser i de simulerede systemer. væskelignende Klynger og en omgivende gas. Med partikler hvor forskellen mellem det frastødende og det tiltrækkende potentiale er stor sker overgangen fra præklyngefasen til klyngefasen ved lave densiteter som kan ses på figur 3.1. A_a er parameteren der bliver skruet på i parpotentialet i Ligning 4.11. Der bliver undersøgt parpotentialer hvor A_a er i intervallet fra 1.50 til 1.60. Ved de laveste værdier bliver det meget svært at adskille gassen fra klynger. Dette skyldes at der i gassen kan findes hvad jeg vil kalde tekniske klynger. Tekniske klynger er mere lange strenge af partikler end tætte kugler, som vi er interesseret i. Hvis den eneste information der er for en given klynge er antallet af partikler den indeholder, kan der aldrig med sikkerhed adskille de

tekniske klynger fra fysiske klynger.

Et system der indeholder klynger vil som regel have en bimodal størrelsesfordeling. Dette kan udnyttes således at minimummet mellem de to toppe kan bruges som minimumsstørrelsen for en klynge bliver kaldt en klynge. Denne definitionsmetode har sine begrænsninger, men for at bestemme det totale antal klyngepartikler er der ikke nogen klar fejlkilde den ene eller den anden vej da der både bliver ekskluderet partikler fra klyngefordelingen og inkluderet partikler fra fordelingen af frie partikler.

Klyngestørrelsen som er minimummet mellem de to toppe kaldes oligomeren [9]. Det kan være svært at finde sandsynligheden for oligomeren. For at forklare dette så lad os kigge på et simpelt eksempel. Lad os forestille os at der er et system med 1000 partikler og sandsynligheden for at finde en partikel i oligomeren med en klyngestørrelse på 25 partikler teoretisk er 0.1%. Hvis oligomeren tilfældigvis blev fundet én gang i en konfiguration ville den empiriske sandsynlighed for oligomeren være $\frac{25}{1000}$, hvilket er højere end den teoretiske. Alternativt kunne oligomeren ikke være repræsenteret i en anden konfiguration og her ville den empiriske sandsynlighed være 0 hvilket er for lavt. Selvfølgelig hvis man kigger på nok konfigurationer ville gennemsnittet af de empiriske sandsynligheder nærme sig de 0.1%, men der vil altid være et minimum for hvor lav den gennemsnitlige empiriske værdi kan blive uden at være lig med 0. I praksis er dette et problem da antallet af konfigurationer bliver meget højt hvis man skal kunne ramme den lave sandsynlighed for oligomeren og klyngestørrelserne omkring denne. En løsning på dette er at gruppere usandsynlige klynger sammen.

Der bliver brugt mindste kvadraters metode til at lave regression på disse fordelinger og en funktion af denne, er at høje værdier bliver vægtet mest. Da oligomeren er essentiel for at adskille klynger fra frie partikler, ville det foretrækkes at regressionen vægtede punkter omkring denne højere. Derfor log-transformeres data. Dette kræver dog at der ikke er nul-værdier i data. for at undgå dette, bliver der dannet klyngegrupperinger. Når der er et interval af klyngestørrelser hvori der ikke blev fundet partikler bliver disse lagt sammen hvortil de to klyngestørrelser hvor i der ér blevet fundet partikler også bliver inkluderet således at der nu er et interval som ikke er lig nul. Hvis der er én klyngestørrelse hvortil der er fundet partikler, men den er omgi-

vet af nulintervaller på begge sider bliver den inkluderet uden at afgrænse intervallet.

$$p(\vec{M}) = (\dots, p_n, p_{n+1}, p_{n+2}, p_{n+3}, p_{n+4}, p_{n+5}, p_{n+6}, \dots) \quad (5.22)$$

Hvor p_n er sandsynligheden for at en partikel befinder sig i en klynge af størrelsen M_n . hvis vi forestiller os at p_{n+2} bliver fundet til at være nul så kollapses intervallet så der kommer en ny værdi der er lig med

$$p' = \langle (p_{n+1}, p_{n+2}, p_{n+3}) \rangle \quad (5.23)$$

og den tilsvarende M' bliver

$$M' = \sum_{m=n+1}^{n+3} \frac{M_m p(M_m)}{\sum_{m=n+1}^{n+3} p_m} \quad (5.24)$$

Hvis både p_{n+2} og p_{n+4} blev fundet til at være lig med nul ville intervallet der skulle kollapses være $]n + 2, n + 5[$

En teknisk klynge opstår når en gruppe af partikler tilfældigvis befinder sig tæt nok på hinanden til at de bliver talt til at være i samme klynge. Erfaring viser at når der laves en klyngeanalyse af en idealgas uden klynger får man en monotonisk aftagende størrelsesfordeling hvor klynger med kun 1 partikel bliver fundet mest. Som at densiteten af gassen stiger bliver tekniske klynger mere almindelige.



Figur 5.2: Figuren her viser den største største klynge i en konfiguration. Dette viser en svaghed i en hierarkisk klyngeanalyse. I denne konfiguration befandt der sig sfæriske klynger men de var alle mindre end denne tilsyneladende tilfældigt sammensatte klynge.

6 Resultater

Der blev først lavet simuleringer hvor parpotentialparametrene $A_a = 1.60$, $A_r = 1.0$, $z_a = 1.0$ og $z_r = 0.5$ blev brugt. Der blev udført to simuleringer der hver startede i forskellige konfigurationer. I den ene startkonfiguration var alle partikler kondenseret i en stor klump der var dannet ved at køre en simulering uden det frastødende potentiale. Herefter blev simuleringen genstartet hvor det frastødende potentiale blev genaktiveret. Den anden simulering startede jævnt fordelt i et sc-gitter. Når begge disse simuleringer blev startet blev det tjekket om de begge nåede samme ligevægtstilstand. Der blev også tjekket om de nåede samme potentielle energi, men det viste sig at der kunne være et substantielt overlap mellem de to simuleringers fluktuationer i potentiel energi hvor hver havde distinkte konfigurationer. En værdi der ikke flukturerer i høj grad er antallet af klynger. I simuleringen der startede i en kondenseret tilstand startede der med at være mange klynger. I simuleringen der startede i sc-gitter var der ingen klynger i starten, hvorefter der blev dannet flere og flere. Kravet der blev stillet for at erklære at simuleringerne kunne nå i ligevægt var at der skulle være overlap i intervallet af antal klynger hver simulering kunne generere. Det viste sig at ved $A_a > 1.6$ og $\rho > 0.0700$ begyndte systemer at opføre sig metastabilt, dvs at de to simuleringer ikke nåede samme ligevægt. Dette er derfor den højeste A_a -værdi der bliver benyttet.

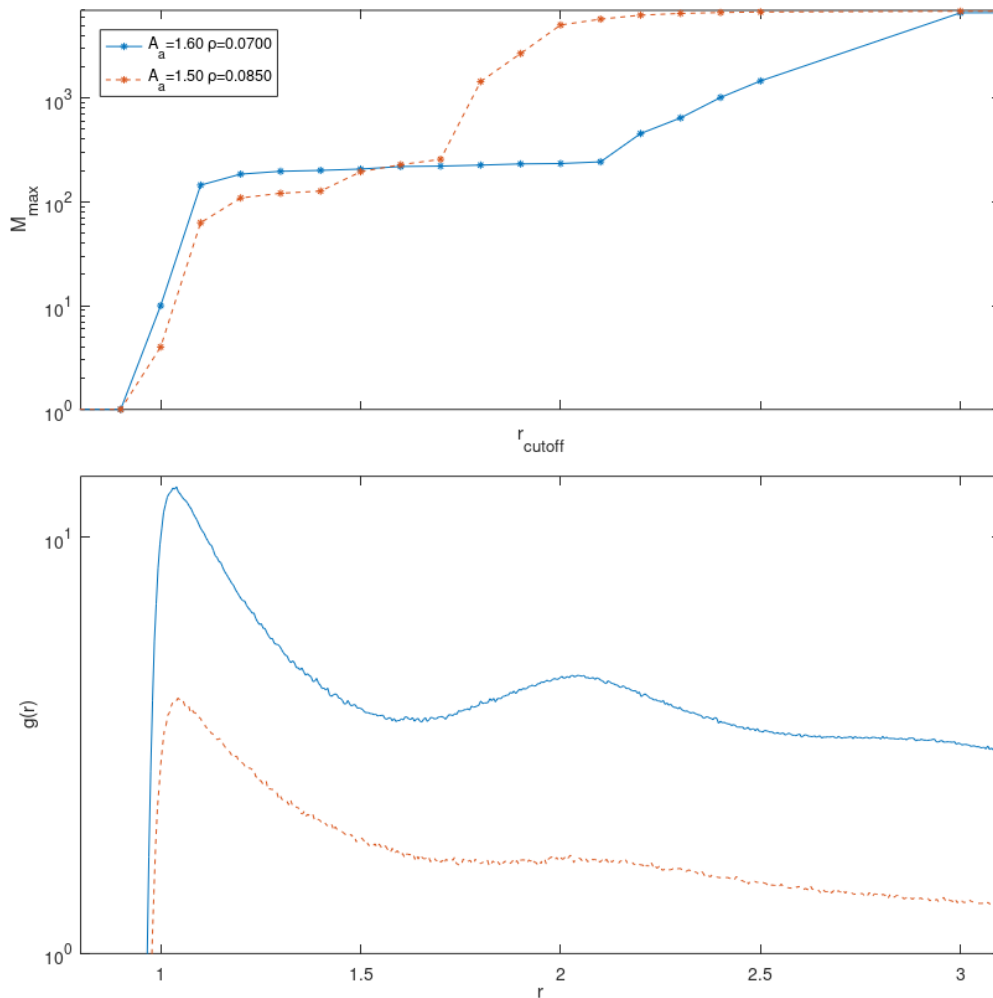
6.1 Bestemmelse af Cutoff i Klyngeanalyse

som beskrevet tidligere bruger klyngeanalysen en afskæringslængde en såkaldt cutoff. Denne længde afgør om to partikler tilhører samme klynge. Hvis distancen mellem to partikler er lavere end cutoff'en siges det at de tilhører samme klynge. For at definere en god cutoff opstilles en analyse hvor der kigges på hvordan størrelsen af klyngerne udvikler sig som at cutoff'en ændres.

Der er blevet lavet klyngeanalyse på en konfiguration fra en simulering kørt i ligevægt med parametrene $A_a = 1.60$ og $\rho = 0.0700$ og tilsvarende på en konfiguration med parametrene $A_a = 1.50$ og $\rho = 0.0850$. Dette er for at repræsentere den højeste og den laveste A_a -værdi. Densiteterne er valgt fordi

det vides med sikkerhed at der bliver dannet klynger ved disse.

Klyngeanalysen bliver foretaget med en række forskellige cutoff-værdier hvor-
 efter antallet af partikler i den største klynge, fremover kaldt M_{max} , bliver
 noteret. Der bliver kun benyttet den største klynge. Graferne for M_{max} kan
 ses i toppen af figur 6.1. Der kigges også på den radiale distributionsfunktion
 (RDF), som kan ses i bunden af figur 6.1. Det viser sig at M_{max} laver et



Figur 6.1: øverste graf viser hvordan den størrelsen af den største klynge i et system ændrer sig som funktion af cutoff'en i klyngeanalysen r_{cutoff} . Den nederste graf viser den radiale distributions funktion.

plateau omkring en cutoff på 1.1 for begge konfigurationer. I rdf'en kan det

ses at det er mellem 1.0 og 1.1 at rdf'en topper. Det er på dette plateau der skal udvælges et cutoff. Grafen for $A_a = 1.60$ holder dette plateau endtil 5.1 hvor det igen stiger indtil hele systemet er i samme klynge. For grafen til $A_a = 1.50$ begynder fremkommer et lille plateau inden den begynder at klynge hele systemet sammen. Dette skyldes formentlig at der har været to klynger der har været tætte nok på hinanden til at blive talt som en klynge. Det er kun den største klynge der bliver noteret så dette behøver ikke at være et problem da det ikke betyder meget hvis der en gang imellem bliver talt to klynger som en. hvis der kigges på rdf'en vil det give mening at vælge minimummet mellem første og anden top, så der sikres at man tæller de tætteste partikler med. Rdf'en for $A_a = 1.60$ har flere partikler i klynger så dens rdf beskriver i højere profilen fra klyngerne end den for $A_a = 1.50$ da denne i højere grad er overdøvet af de frie partikler. her er minimummet desuden så langt henne at hvis denne blev benyttet som cutoff stigningen mod det sidste plateau begyndt. der er derfor valgt at blive benyttet en cutoff på 1.4.

6.2 Ligevægtsimuleringer

Nu hvor der er defineret en passende cutoff længde kan der laves klyngeanalyse. Når denne analyse er udført bliver hver partikel tildelt et indeks der fortæller hvilken klynge den tilhører. Ved at se på hvor mange partikler der har samme indeks kan dette bruges som et udtryk for klyngens størrelse.

For at kunne adskille på klynger af frie partikler og store klynger ville det være ideelt at opstille en fordeling som er bimodal. Istedet for at kigge på fordelingen af klynger kigges der istedet på fordelingen af partikler over hvilke størrelse klynge de tilhører. Det kan skrives således

$$p(M) = \frac{N(M)M}{N_t} \quad (6.1)$$

Hvor M er størrelsen af en given klynge, $p(M)$ er sandsynligheden for at udvælge en partikel i en klynge af størrelsen M , $N(M)$ er antallet af klynger med størrelsen M og N_t er det totale antal af partikler.

Der er en betydelig mængde støj i fordelingerne, for at mindske denne støj er der blevet brugt en regressionsmodel på dem. Det viser sig $p(M)$ for de store klynger ser normalfordelt ud. Fordelingen af de små klynger er dog ikke

åbenlys. Med logaritmisk y-akse er der en skrå asymptote. Dette betyder at det bevæger sig mod at være eksponentiel. Følgende model har vist sig at fungere nogenlunde

$$p_{\text{fri}}(M) = a \cdot \exp(b \cdot \exp[-c \cdot M] - d \cdot M) \quad (6.2)$$

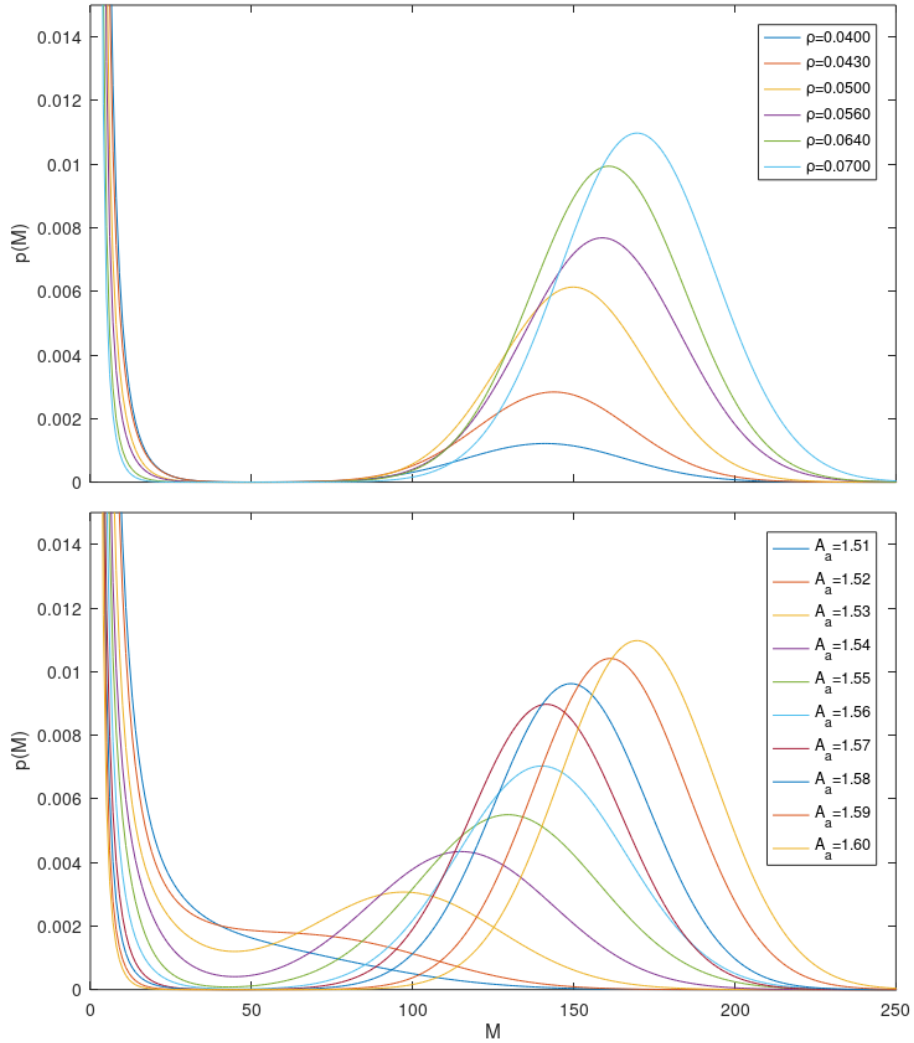
$p_{\text{fri}}(M)$ er fordelingen for små klynger der bliver dannet ud af gas. a , b , c , og d er parametre der kan skrues for optimere fittet. Fordelingen for de store klynger bliver fittet med Gauss funktionen. Det fulde udtryk bliver dermed

$$p(M) = p_{\text{fri}}(M) + \frac{A}{\sigma\sqrt{2\pi}} \exp\left(-\frac{(M - \mu)^2}{2\sigma^2}\right) \quad (6.3)$$

hvor A er amplituden, σ er spredningen og μ er gennemsnitsstørrelsen. Denne model er god når der er en klar bimodal fordeling. Som når de to fordelinger smelter sammen og der ikke er en distinkt top. I disse tilfælde er der blevet brugt et modificeret 3. grads polynomie til at beskrive fordelingen

$$\log[p(M)] = (aM^3 + bM^2 + cM + d)^e \quad (6.4)$$

Denne funktion er god til at finde ekstremumpunkter. Det er nyttigt til at finde oligomeren.



Figur 6.2: Figuren viser hvordan størrelsesfordelingen udvikler sig når densiteten øges med konstant parpotentiale (øverste grafer) og når der skrues op for A_a -parameteren (nederste grafer). I de øverste grafer er A_a sat til at være 1.60. I de nederste grafer er densiteten sat til 0.0700.

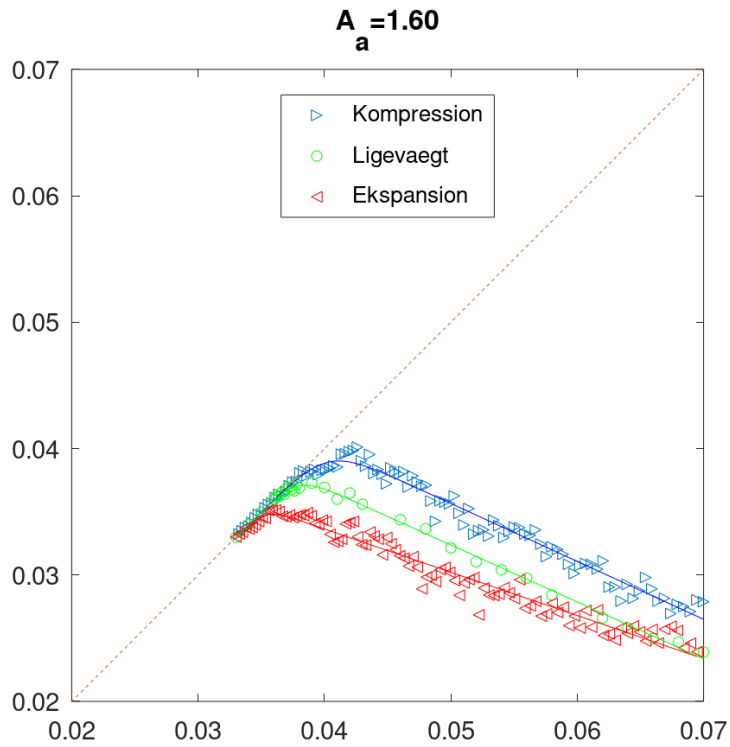
Der bliver skruet på to parametre i simuleringerne, A_a -værdien og densiteten. i Figur 6.2 kan størrelsesfordelingerne ses.

For den ene graf hvor der bliver skruet på densiteten kan det ses at i starten bliver gauss kurven pustet op således at amplituden og μ bliver større. Kurven for $\rho = 0.0700$ ser ud til at være skubbet til højre. Det samme ser ud til at ske fra $\rho = 0.0500$ til $\rho = 0.0560$. For alle fordelingerne for $A_a = 1.60$ ser der ikke ud til at ske en drastisk ændring i minima, fordi der er så få partikler der befinder i og omkring oligomeren betyder det ikke så meget for videre resultater om den bliver defineret lidt for højt eller lidt for lavt.

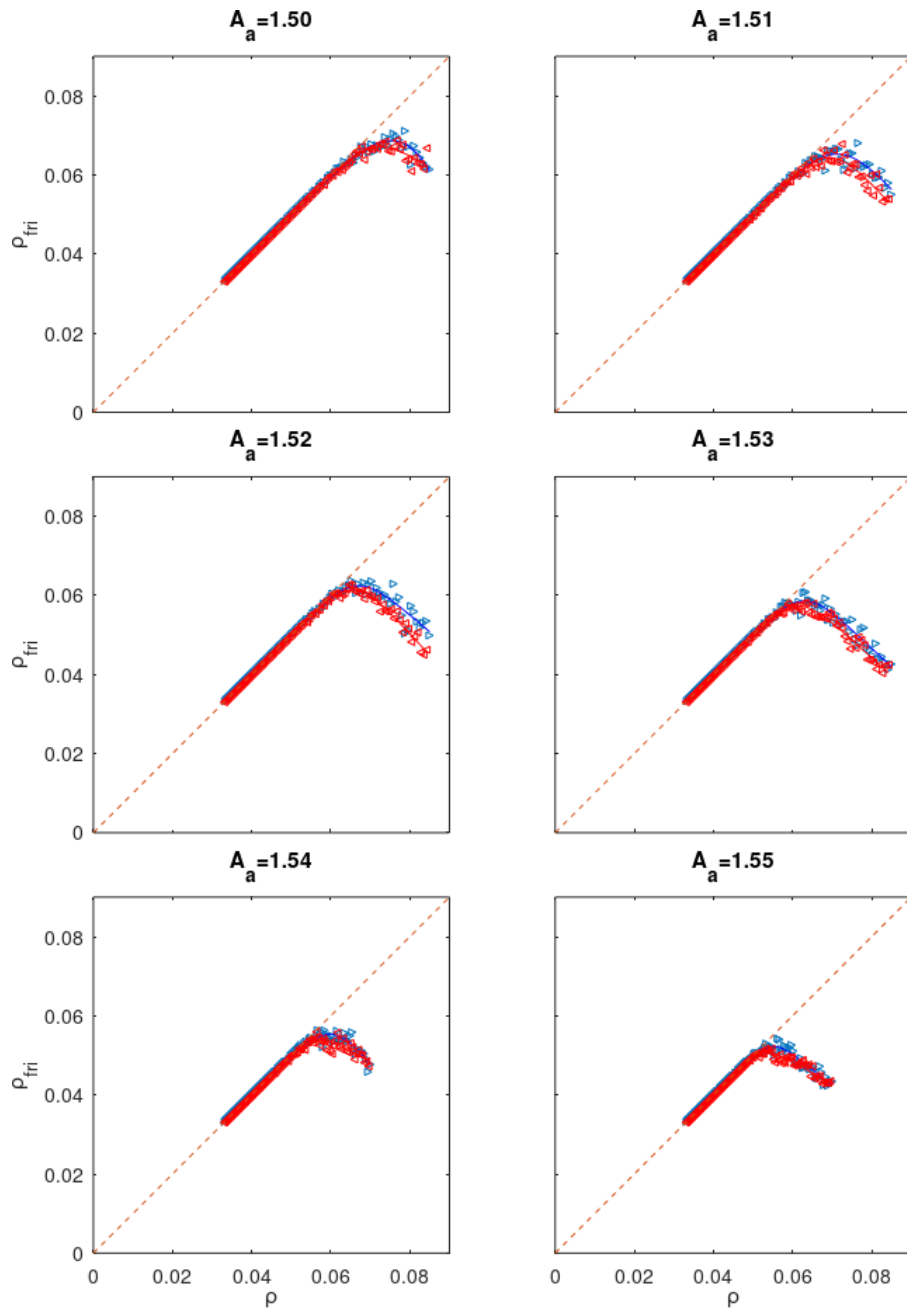
I den nederste graf i Figur 6.2 hvor der skrues på A_a -parameteren har kurverne for $A_a = 1.51$ og $A_a = 1.52$ ikke noget maksima eller minima. Fra $A_a = 1.52$ og frem er der et klart maksima og minima i kurverne. Det ser desuden ud til at minimaet ikke rykker sig betydeligt efter det først er dannet. Senere når ρ_{fri} skal findes, bruges $M = 40$ som oligomeren ud fra en kvalitativ vurdering af kurverne i Figur 6.2.

6.3 Simuleringer med hysteresse

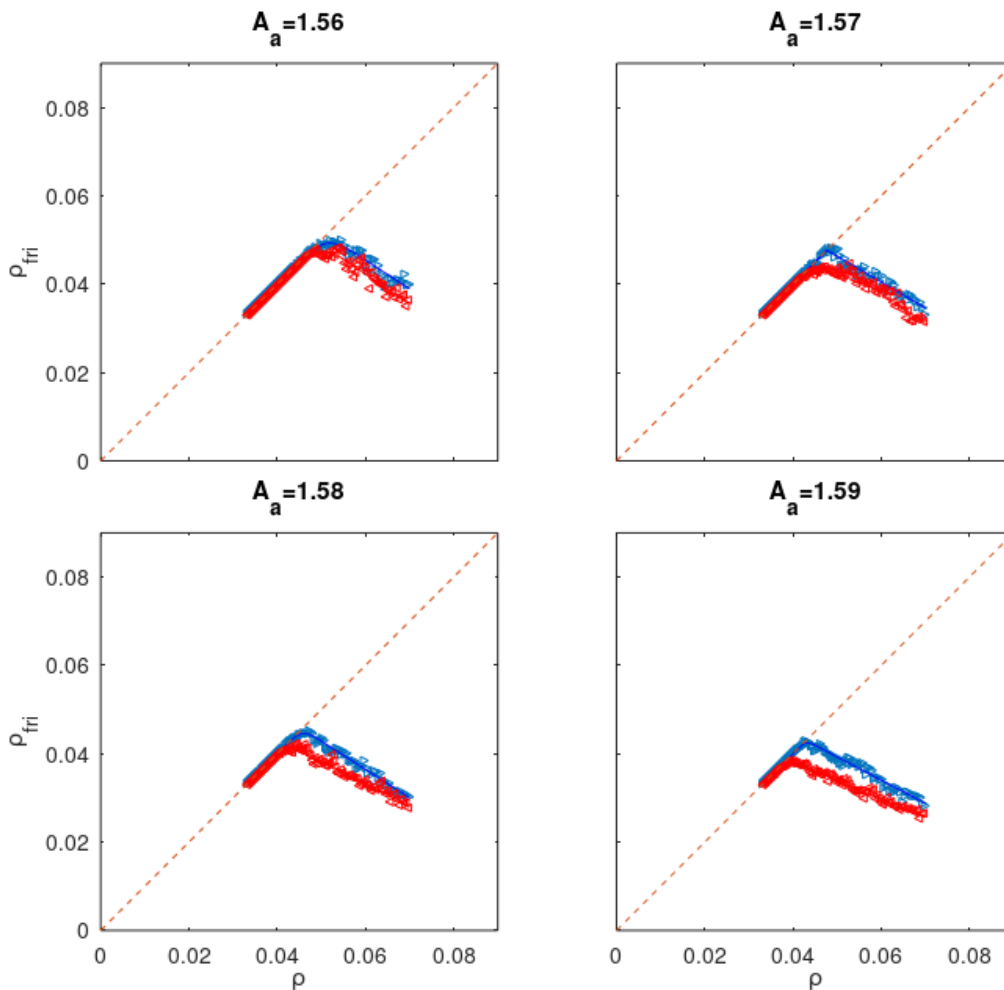
Kritiske klyngedannelsesdensitet bliver fundet ved at bruge modellen i ligning 5.19 som regressionsmodel til data der bliver dannet via simulering med RUMD.



Figur 6.3: De blå højrepegende trekkanter er densiteten af frie partikler for en simulering der startede med en bulk densitet på 0.0330 som blev forøget over simuleringstiden til systemet havde en bulk densitet på 0.0700. Hvert punkt er værdien for én konfiguration og det er derfor forventeligt med en del fluktuation. De grønne cirkler er gennemsnitsdensiteten af frie partikler for 144 konfigurationer af et system i ligevægt. De røde venstrepegende trekkanter er ρ_{fri} for en simulering der startede med en konfiguration der var kørt i ligevægt med en bulk densitet på 0.0700 hvorefter densiteten blev sænket over simuleringstiden indtil den var på 0.0330. Ligesom kompressionen, er hvert punkt i ekspansionen værdier for én konfiguration



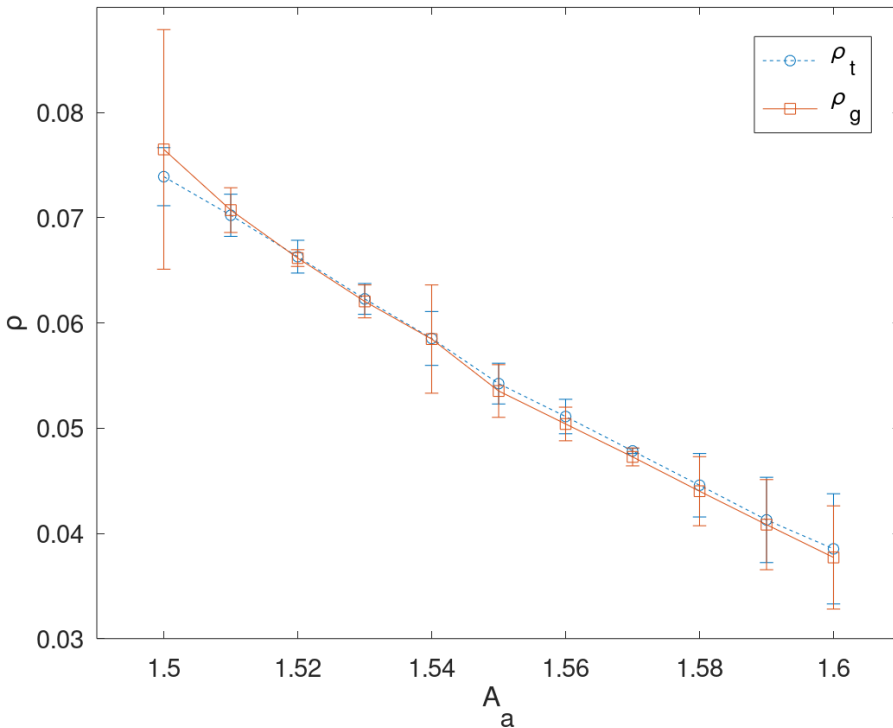
Figur 6.4: Graferne viser ρ_{fri} for kompressions- (vist med blå højrepegende trekanter) og ekspansionssimuleringer (vist med røde venstrepegende trekanter). A_a -parametre går fra 1.50 til 1.55. figurene $A_a = 1.50$ til $A_a = 1.53$ er lavet i densitetsintervallet 0.0330-0.0850, mens at resten er lavet i intervallet 0.0330-0.0700. ligning 5.19 er blevet brugt som regressionsmodel. Kurverne viser denne regression



Figur 6.5: Graferne viser ρ_{fri} for kompressions- (vist med blå højrepegende trekanter) og ekspansions- (vist med røde venstrepegende trekanter) simuleringer. A_a -parametre går fra 1.50 til 1.55. ligning 5.19 er blevet som regressionsmodel. Kurverne viser denne regression

Der blev først testet om hvorvidt der kunne bruges ekspansions- og kompressionssimuleringer til at finde overgangen, eller om hysteresen ville være for stor til at få noget meningsfuldt ud af det. Disse resultater blev derfor sammenlignet med hinanden og med resultater hvor simuleringen med sikkerhed er kørt i ligevægt. Resultaterne fra dette kan ses i figur 6.3. Det viser sig at ekspansions- og kompressionssimuleringerne generelt laver et underestimat af ρ_{fri} og kompressionssimuleringen generelt laver et overestimat. Selv med hysteresen ser det ud

til at ligning 5.19 stadig er et udmærket bud som regressionsmodel på hvert datasæt. Hver regression har en ρ_t og en ρ_g og det tyder på at disse to værdier for ligevægtskurven ligger et sted mellem dem der findes for ekspansions- og kompressionskurven. Herfra vil ekspansions- og kompressionssimuleringer be-



Figur 6.6: Figuren viser den kritiske klyngedannelsesdensitet ved forskellige A_a -værdier ud fra to definitioner. I den ene definition er det ρ_g som er der hvor ligning 5.1 og 5.12 krydser. Den anden beskriver toppunktet for funktionen i ligning 5.19, og bliver beregnet ved hjælp af ligning 5.21

nyttes til at finde den kritiske klyngedannelsesdensitet for andre A_a -værdier. ρ_t og ρ_g fra hver kurve vil blive brugt som usikkerhedsinterval for hvor disse værdier ville befinde sig for ligevægtskurven. Kurverne for $A_a = 1.50, 1.51, 1.52, 1.53, 1.54, 1.55, 1.56, 1.57, 1.58$ og 1.59 kan ses i figur 6.4 og 6.5.

Nu er der lavet regression på en række A_a -værdier ud fra disse regressioner kan der findes ρ_g og ρ_t som kan bruges til at tegne kritiske klyngedannelsesdensiteter i fasediagrammet. I figur 6.6 kan der ses 2 linjer. Den ene er den kritiske klyngedannelsesdensitet for forskellige A_a -værdier defineret ud

fra henholdsvis ρ_g , som er en given parameter i ligning 5.19, og ρ_t , som er defineret i ligning 5.21.

I dette arbejde er der ikke blevet fundet en analytisk model at holde disse empirisk data op imod. Det er dog blevet bekræftet at den kritiske klynge-dannelsesdensitet bliver lavere som at der skrues op for A_a . Dette betyder at jo lavere partiklernes tiltrækkende kraft er, jo sværere bliver det at danne klynger.

7 Diskussion

Området i fasediagrammet der er blevet undersøgt er ved lavere A_a -værdier end hvad der skulle forventes af fasediagrammet i Figur 3.1. Dette skyldes at 1.60 var den højeste A_a -værdi hvor der ikke blev fundet metastabile tilstande. Resultaterne viser også at den kritiske klyngedannelsesdensitet for $A_a = 1.60$ ligger omkring 0.0400 hvilket er meget lavere end hvad der bliver fremstillet i fasediagrammet i Figur 3.1. Hvorfor resultaterne afviger så meget er svært at sige. Det er muligt at de antagelser fasediagrammet er baseret på ikke holder.

På Figur 6.4 og Figur 6.5 ser det ud til at hældningen på kurven efter knækket bliver stejlere ved lavere A_a -værdier. Hvis tesen om at ρ_g og ρ_l er korrekt ville dette give mening, siden at der hvor grafen skærer første-aksen er densiteten af væsken. Der er ikke noget der tyder på at densiteten inde i klyngerne bliver højere ved lavere A_a -værdier, tværtimod, og overgangen sker ifølge Figur 6.6 ved højere densitet. Det betyder k må blive højere ved lave A_a -værdier. k går mod uendelig når ρ_g og ρ_l nærmer sig hinanden. Det giver mening da der her ikke længere er et koeksistens interval. Her bliver fluiden bare tættere. Ændringen af k gør at ρ_t ændrer position relativt til ρ_g . Når $k > 1$ er toppunktet højere end ρ_g og når $k < 1$ er ρ_t lavere end ρ_g . Jo skarpere overgangen er jo tættere ligger de på hinanden.

En fordel ved at bruge Ligning 5.19 til at fitte ρ_{fri} over ρ er at der egentlig ikke behøver at være resultater omkring den kritiske klyngedannelsesdensitet. Det er heller ikke nødvendigt at danne data for gasfasen, da modellen allerede antager at kurven ligger op af $\rho_{\text{fri}}(\rho) = \rho$ linjen.

I Figur 6.2 sker der noget sjovt når densiteten bliver forøget fra 0.0640 til 0.0700. Kurven bliver forskudt til højre hvilket tyder på at der ikke bliver dannet flere klynger. der bliver i stedet brugt partikler på at forøge klyngernes størrelse. Dette sker også når der skrues på A_a . Fra $A_a = 1.56$ til $A_a = 1.57$ bliver kurven pustet op uden at toppunktet ændrer sig. Det tyder på at der i denne overgang bliver brugt partikler til at danne nye klynger uden at forstørre dem. Disse særheder er muligvis tegn på overgange i mesofaserne.

8 Opsummering

Det er blevet fundet at en god cutoff til klyngeanalysen var 1.4. Det skyldes at der var et plateau i $M_{\max}(r_{\text{cutoff}})$ fra 1.1 til 1.4 hvorefter to klynger begyndte at blive talt som en.

Med ligevægtssimuleringer blev der lavet fordelingskurver der viste hvordan disse fordelinger ændrede sig som at der blev skruet på enten A_a -værdien og densiteten. Ud fra disse kurver kunne det ses at lige så snart at der var et minima og maksima blev oligomeren ikke ændret betydeligt. Ud fra kvalitative vurdering blev det vurderet at $M = 40$ ville være en god adskiller af frie partikler og partikler i klynger.

Til sidst blev der lavet ekspansions- og kompressionssimuleringer for at finde den kritiske klyngedannelsesdensitet. Ved at kigge på densiteter ved $A_a = 1.60$ blev det fundet at målinger af ρ_{fri} i ekspansionssimuleringen laver et underestimat, mens at målinger i kompressionssimuleringer laver et overestimat. Målingerne for disse to simuleringer kan desuden fittes til modellen i Ligning 5.19. ρ_g ρ_t fra disse fits kan bruges som spændet hvori vi ville finde den kritiske klyngedannelsesdensitet for det quasistatiske tilfælde.

Litteratur

- [1] M. B. Sweatman, R. Fartaria, and L. Lue, “Cluster formation in fluids with competing short-range and long-range interactions,” *J. Chem. Phys.*, vol. 140, 2014.
- [2] C. P. Royall, “Hunting mermaids in real space: known knowns, known unknowns and unknown unknowns,” *Soft Matter*, vol. 14, pp. 4020–4028, 2018.
- [3] V. N. Uversky, “Intrinsically disordered proteins in overcrowded milieu: Membrane-less organelles, phase separation, and intrinsic disorder,” *Current Opinion in Structural Biology*, vol. 44, pp. 18 – 30, 2017. Carbohydrates: A feast of structural glycobiology • Sequences and topology: Computational studies of protein-protein interactions.
- [4] M. B. Sweatman and L. Lue, “The giant salt cluster fluid: a review,” *Advanced Theory and Simulations*, vol. 2, 7 2019.
- [5] J. Riest, G. Nägele, Y. Liu, N. J. Wagner, and P. D. Godfrin, “Short-time dynamics of lysozyme solutions with competing short-range attraction and long-range repulsion: Experiment and theory,” *The Journal of Chemical Physics*, vol. 148, no. 6, p. 065101, 2018.
- [6] J. Schulze, J. Möller, J. Weine, K. Julius, N. König, J. Nase, M. Paulus, M. Tolan, and R. Winter, “Phase behavior of lysozyme solutions in the liquid–liquid phase coexistence region at high hydrostatic pressures,” *Phys. Chem. Chem. Phys.*, vol. 18, pp. 14252–14256, 2016.
- [7] N. P. Bailey, T. S. Ingebrigtsen, J. S. Hansen, A. A. Veldhorst, L. Bøhling, C. A. Lemarchand, A. E. Olsen, A. K. Bacher, L. Costigliola, U. R. Pedersen, H. Larsen, J. C. Dyre, and T. B. Schrøder, “Rumd: A general purpose molecular dynamics package optimized to utilize gpu hardware down to a few thousand particles,” *SciPost Physics*, vol. 3, no. 038, 2017.
- [8] M. B. Sweatman and L. Lue, “Communication: The cluster vapor to cluster solid transition,” *J. Chem. Phys.*, vol. 144, may 2016.

- [9] A. P. Santos, J. Pkalski, and A. Z. Panagiotopoulos, “Thermodynamic signatures and cluster properties of self-assembly in systems with competing interactions,” *Soft Matter*, vol. 13, pp. 8055–8063, 2017.
- [10] S. Das, J. Riest, R. G. Winkler, G. Gompper, J. K. G. Dhont, and G. Nägele, “Clustering and dynamics of particles in dispersions with competing interactions: Theory and simulation,” *Soft Matter*, 2017.
- [11] H. J-P and M. I. R., *Theory of Simple Liquids: with Applications to Soft Matter*. New York: Academic, fourth ed., 2013.
- [12] N. Grønbech-Jensen, N. R. Hayre, and O. Farago, “Application of the g-jf discrete-time thermostat for fast and accurate molecular simulations,” *Computer Physics Communications*, vol. 185, pp. 524–527, feb 2014.

A Kompressionssimulering

```
#!/usr/bin/python3
# -*- coding: utf-8 -*-
"""
RUMD Simulation script

@author: urp
"""

import os
import sys
import numpy as np
import math
import rumd_clusters

rumdpath = "/net/debye/urp/rumd/rumd/"
py_rumd_lib = {2: rumdpath + "lib", 3: rumdpath+"lib_py3"}[sys.version_info[0]]
sys.path.insert(1, py_rumd_lib)
sys.path.insert(1, rumdpath+"Python")

import rumd
from rumd.Simulation import Simulation
import rumd.Tools
#import rumd.interactive_md as md
from rumd.Autotune import Autotune # FIXME Run autotuner
#at = Autotune()

sim = Simulation("start.xyz.gz")

#setup expansion data
blocksize= 2**18
nParticles = sim.GetNumberOfParticles()
```

```

vol = sim.GetVolume()
rho = nParticles/vol
rd = 0.0460

#Setup potential energy function
Aa = 1.60
Ar = 0.5
za = 1.0
zr = 0.5
rc = 15.0

sigr = 1.0 / zr
siga = 1.0 / za

exr = math.exp(1/sigr)
exa = math.exp(1/siga)

sig = 1.0      #d in the article
eps = 1.0
epsa = Aa / siga * exa
epsr = Ar / sigr * exr

# rowlinsons hard sphere diameter page 118 in theory of simple liquids by hansen and
n = 50
d = 1.0+0.5772/n

# Short ranged repulsive potential
u_sr = rumd.Pot_IPL_n(cutoff_method=rumd.ShiftedPotential, n=n)
u_sr.SetParams(0, 0, Sigma=sig/d, Epsilon=eps, Rcut=rc)
sim.SetPotential(u_sr) # FIXME update RUMD and change to AddPotential()
# Short ranged attractive potential
u_a = rumd.Pot_Yukawa(cutoff_method=rumd.ShiftedPotential)
u_a.SetParams(0, 0, Sigma=sig*siga, Epsilon=-eps*epsa, Rcut=rc)
sim.AddPotential(u_a)

```

```
# Long range repulsive potential
u_r = rumd.Pot_Yukawa(cutoff_method=rumd.ShiftedPotential)
u_r.SetParams(0, 0, Sigma=sig*sigr, Epsilon=eps*epsr, Rcut=rc)
sim.AddPotential(u_r)

T=1 # FIXME make langevin integrator
#itg = rumd.IntegratorNVT(timeStep=0.005, targetTemperature=T)
#itg = rumd.IntegratorNVE(timeStep=0.005)
# Setup NpT Langevin barostat
itg = rumd.IntegratorNPTLangevin(timeStep=0.005,
                                targetTemperature=T,
                                friction=0.1,
                                targetPressure=0, # Barostat not used
                                barostatFriction=100,
                                barostatMass=100)
itg.SetBarostatMode(rumd.Off) # Run NVT
sim.SetIntegrator(itg)

#sim.SetOutputParameters("trajectory","linear",interval=2000)
#sim.SetOutputParameters("energies","linear",interval=20)
#sim.SetBlockSize(200000)
sim.SetOutputMetaData("energies", volume=True)
#at.Tune(sim)

sim.SetOutputScheduling("energies","linear", interval=blocksize)
sim.SetOutputScheduling("trajectory","linear", interval=blocksize)

sim.SetBlockSize(blocksize)

class Compressor:
    def __init__(self, sample, scaleFactor):
        self.sample = sample
        self.scaleFactor = scaleFactor
    def Compress(self):
```

```
        self.sample.IsotropicScaleSystem(self.scaleFactor)
num_scale_intervals = 100
density_factor =rd/rho
scale_factor = math.exp(-math.log(density_factor)/3/num_scale_intervals)
comp = Compressor(sim.sample, scale_factor)
scale_interval=blocksize
sim.SetRuntimeAction("compress", comp.Compress, scale_interval)
sim.Run(scale_interval * num_scale_intervals)

#sim.Run(2**16*6*24)
#sim.sample.TerminateOutputManagers()

sim.WriteConf("end.xyz.gz")

rumd_clusters.main(['-i', 'end.xyz.gz', '-r', 1.6, '-m', 11])
```

B Ekspansionsssimulering

```
#!/usr/bin/python3
# -*- coding: utf-8 -*-
"""
RUMD Simulation script
"""

import os
import sys
import numpy as np
import math
import rumd_clusters

rumdpath = "/net/debye/urp/rumd/rumd/"
py_rumd_lib = {2: rumdpath + "lib", 3: rumdpath+"lib_py3"}[sys.version_info[0]]
sys.path.insert(1, py_rumd_lib)
sys.path.insert(1, rumdpath+"Python")
```

```
import rumd
from rumd.Simulation import Simulation
import rumd.Tools
import rumd.interactive_md as md
from rumd.Autotune import Autotune # FIXME Run autotuner
#at = Autotune()

sim = Simulation("start.xyz.gz")

#setup expansion data
blocksize= 2**18
nParticles = sim.GetNumberOfParticles()
vol = sim.GetVolume()
rho = nParticles/vol
rd = 0.0330

#Setup potential energy function
Aa = 1.60
Ar = 0.5
za = 1.0
zr = 0.5
rc = 15.0

sigr = 1.0 / zr
siga = 1.0 / za

exr = math.exp(1/sigr)
exa = math.exp(1/siga)

sig = 1.0 #d in the article
eps = 1.0
epsa = Aa / siga * exa
epsr = Ar / sigr * exr
```

```
# rowlinsons hard sphere diameter page 118 in theory of simple liquids by hansen and  
n = 50  
d = 1.0+0.5772/n  
  
# Short ranged repulsive potential  
u_sr = rumd.Pot_IPL_n(cutoff_method=rumd.ShiftedPotential, n=n)  
u_sr.SetParams(0, 0, Sigma=sig/d, Epsilon=eps, Rcut=rc)  
sim.SetPotential(u_sr) # FIXME update RUMD and change to AddPotential()  
# Short ranged attractive potential  
u_a = rumd.Pot_Yukawa(cutoff_method=rumd.ShiftedPotential)  
u_a.SetParams(0, 0, Sigma=sig*siga, Epsilon=-eps*epsa, Rcut=rc)  
sim.AddPotential(u_a)  
# Long range repulsive potential  
u_r = rumd.Pot_Yukawa(cutoff_method=rumd.ShiftedPotential)  
u_r.SetParams(0, 0, Sigma=sig*sigr, Epsilon=eps*epsr, Rcut=rc)  
sim.AddPotential(u_r)  
  
T=1 # FIXME make langevin integrator  
#itg = rumd.IntegratorNVT(timeStep=0.005, targetTemperature=T)  
#itg = rumd.IntegratorNVE(timeStep=0.005)  
# Setup NpT Langevin barostat  
itg = rumd.IntegratorNPTLangevin(timeStep=0.005,  
                                targetTemperature=T,  
                                friction=0.1,  
                                targetPressure=0, # Barostat not used  
                                barostatFriction=100,  
                                barostatMass=100)  
itg.SetBarostatMode(rumd.Off) # Run NVT  
sim.SetIntegrator(itg)  
  
#sim.SetOutputParameters("trajectory","linear",interval=2000)  
#sim.SetOutputParameters("energies","linear",interval=20)
```

```
#sim.SetBlockSize(200000)
sim.SetOutputMetaData("energies", volume=True)
#at.Tune(sim)

sim.SetOutputScheduling("energies","linear", interval=blocksize)
sim.SetOutputScheduling("trajectory","linear", interval=blocksize)

sim.SetBlockSize(blocksize)

class Compressor:
    def __init__(self, sample, scaleFactor):
        self.sample = sample
        self.scaleFactor = scaleFactor
    def Compress(self):
        self.sample.IsotropicScaleSystem(self.scaleFactor)

num_scale_intervals = 100
density_factor =rd/rho
scale_factor = math.exp(-math.log(density_factor)/3/num_scale_intervals)
comp = Compressor(sim.sample, scale_factor)
scale_interval=blocksize
sim.SetRuntimeAction("compress", comp.Compress, scale_interval)
sim.Run(scale_interval * num_scale_intervals)

#sim.Run(2**16*6*24)
#sim.sample.TerminateOutputManagers()

sim.WriteConf("end.xyz.gz")

rund_clusters.main(['-i','end.xyz.gz','-r',1.6,'-m',11])
```