Masteroppgave

NTNU
Norges teknisk-naturvitenskapelige universitet
Fakultet for naturvitenskap
Institutt for materialteknologi

Håvard Mo Fagersand

A method for describing systems of spherical droplets in vapor

Masteroppgave i Nanoteknologi
Veileder: Sondre Kvalvåg Schnell

Januar 2021

NTNU
Kunnskap for en bedre verden

Håvard Mo Fagersand

A method for describing systems of spherical droplets in vapor

Masteroppgave i Nanoteknologi
Veileder: Sondre Kvalvåg Schnell
Januar 2021

Norges teknisk-naturvitenskapelige universitet
Fakultet for naturvitenskap
Institutt for materialteknologi

NTNU
Kunnskap for en bedre verden

Abstract

Nanoscale materials have long been known to have unique properties not found in bulk materials. The unique behavior of nanoscale systems also means they can't be described by standard thermodynamic equations, as surface effects become prominent and normally extensive properties behave differently. Molecular dynamics simulation can be a great tool for studying small scale systems, and determining their thermodynamic properties and how they change with system size.

In this thesis, molecular dynamics simulations of a certain kind of small scale system consisting of a droplet surrounded by vapor were performed. A method for classifying droplet-vapor systems called the Voronoi density-neighbor method was utilized to define the liquid, vapor and surface phases of each system. Thermodynamic properties of each phase of each system were calculated, and their change with droplet size was investigated.

It was found that the thermodynamic properties of each system scale roughly linearly with inverse droplet size, in agreement with the small-scale hypothesis. The surface tension of the systems appears to be independent of surface curvature, and therefore obeys the Young-Laplace equation. It has been demonstrated that the Voronoi density-neighbor method can be used to calculate the number of droplets present in a multi-droplet system. Overall, the Voronoi density-neighbor method appears to give a good description of droplet-vapor systems.

Sammendrag

Det har lenge vært kjent at nanomaterialer har unike egenskaper som ikke observeres i materialer på større skala. Den unike oppførselen gjør også at nanosystemer ikke kan beskrives av de vanlige termodynamiske likningene, siden overflateeffekter dominerer og normalt ekstensive egenskaper oppfører seg annerledes. *Molecular dynamics*-simuleringer kan være viktige verktøy for å studere småskalasystemer, og se hvordan termodynamiske egenskaper varierer med systemstørrelse.

I denne oppgaven lagde vi molecular dynamics-simuleringer av en spesiell type småskalasystem, bestående av en væskedråpe i likevekt med en gassfase. En metode for å klassifisere dråpe-gassystemer, kalt *Voronoi density-neighbor*-metoden, ble brukt til å definere væske-, gass- og overflatefasene til systemene. Termodynamiske egenskaper for hver fase ble beregnet, og deres endring med forskjellig dråpestørrelse ble undersøkt.

Vi fant at de termodynamiske egenskapene til hvert system endres lineært med invers radius, som stemmer med småskalahypotesen. Overflatespenningen til systemene ser ut til å være uavhengig av overflatekrummingen, og kan beskrives av Young-Laplace-likningen. Vi har demonstrert at Voronoi density-neighbor-metoden kan brukes til å regne ut antall dråper tilstede i et flerdråpesystem. Alt i alt gir Voronoi density-neighbor-metoden en god beskrivelse av dråpe-gassystemer.

Contents

1 Introduction

It is a well-known phenomenon that on the nanoscale, the properties of materials can change drastically.[1] In thermodynamics, *small scale systems* are defined as systems where extensive thermodynamic properties are no longer directly proportional to the volume, but higher-order functions of both the volume and the shape of the system. [2] As a consequence of this, surface effects, which can be neglected in larger systems, become dominant at this scale.

As an illustration, one can consider a cube of side length 1m.[1] Its area is $6 \cdot (1\text{m})^2 = 6\text{m}^2$, and its volume is $(1\text{m})^3 = 1\text{m}^3$. Next, consider a cube of the same volume, but made out of many small cubes, each with side length $1 \cdot 10^{-9}$m. The total amount of such nano-cubes needed is the total volume divided by the volume of each individual cube:

$$\frac{(1\text{m})^3}{(1 \cdot 10^{-9}\text{m})^3} = 1 \cdot 10^{27} \tag{1}$$

And these nano-cubes have a total surface area of:

$$(6 \cdot (1 \cdot 10^{-9}\text{m})^2) \cdot 1 \cdot 10^{27} = 6 \cdot 10^9\text{m}^2 \tag{2}$$

Which is one billion times the surface area of the cube with side length 1m! In such a system, a significant number of the constituent particles will be surface particles, which is what makes surface effects so significant.[1]

Another consequence of small scale is that properties which are intensive or independent of size in bulk are no longer so in small scale systems. Extensive or additive properties are also found to behave differently from how they would in large scale systems. [1] For instance, in a bulk system, doubling the system size from N to $2N$ particles will double the internal energy of the system from U to $2U$, while the temperature will remain the same. In a small-scale system, however, doubling the system size results in an internal energy different from $2U$, and a change in system temperature.

The novel properties of nanomaterials have led to their use in a wide range of industries. The greatly increased surface area make nanomaterials natural for use in catalysts.[1] Nanoparticles of silver are used for their anti-microbial properties[3], and the cosmetics industry uses many different nanomaterials[4]. Carbon nanotubes, cylindrical structures with walls of one or more layers of graphene, have tensile strength 100 times that of steel and excellent thermal and electrical conductivity. [1] They provide excellent anodes for lithium batteries,[5] and could be used as scaffolding in tissue engineering.[6]

As a consequence of surface effects and the size dependence of thermodynamic properties, classical thermodynamic relations which are valid for large scale systems can not be used directly on small scale systems. The development of thermodynamic equations for describing small-scale systems - *nanothermodynamics* - began in the 1960s with the work of Hill[7], and is still ongoing. Bedeaux et al.[8] have put forth the *small-scale hypothesis*, which states that certain properties of small-scale systems can be defined in terms of an expansion in the inverse of the system's characteristic length. This expansion includes terms which depend on the shape of the system.[8]

Computer models can be excellent tools for studying small-scale systems. One such model is *molecular dynamics simulation*, a classical computational method which uses Newton's laws of motion to determine how a system changes over time. The system's thermodynamic properties are then determined from the average of the property's values calculated at different points in the simulation. [9] Computer simulation offers obvious benefits: For example, one can create models of as of yet un-synthesized materials, and select only the ones with the most promising properties to perform a potentially expensive and/or difficult synthesis of. [10] Impurities are always a potential problem in the laboratory, and especially so for nanoscale systems, while 'creating' an uncontaminated system in a computer simulation is trivial.

In this project, we used molecular dynamics simulations to study a particular kind of small-scale system, consisting of a liquid droplet in equilibrium with a vapor phase. Several such systems of varying droplet size were studied, to see how the extensive thermodynamic properties of the droplet-vapor system changes with droplet size, and use this data to help test the small-scale hypothesis. Such a system could be used to, for example, model the formation of raindrops in rain clouds and study the thermodynamic properties of nucleating droplets of water. Properties which would be of great interest to determine for small-scale water droplets include their density, enthalpy and heat capacity.

To define the different phases present in the system, a novel method was developed and tested, which uses a characteristic volume of each particle as a criterion. This characteristic volume is calculated by a geometric procedure called Voronoi tessellation[11], and is referred to as the particle's Voronoi volume. The novel method is based on the work of Jared T. Fern et al.[12], who have described a method for obtaining single-phase diagrams from a two-phase system from each particle's Voronoi volume. Their method was expanded upon by introducing a step where the Voronoi volume of each particle's nearest neighbors is considered.

2 Theory and background

2.1 Nano-scale effects

Thermodynamically, nano-scale materials lie somewhere between the atomic world, where quantum mechanical effects dominate, and bulk materials which can be adequately described by classical thermodynamics. [1] Nano-scale systems can still be described to an extent by classical thermodynamics. [1] Another option is to employ so-called *small system thermodynamics.*

Hill [7] demonstrated that classical thermodynamic equations can be generalized by including terms which account for small-system effects. Hill's idea involves creating an ensemble of M distinguishable copies of the small-scale system. The ensemble can then be treated with standard thermodynamic relations, with an additional term representing the energy required to divide into another replica.[7]

Bedeaux et al.[8] have formulated the *small-system hypothesis*, based on the work of Hill.[7] It goes as follows: *The density of any additive thermodynamic property can be written as an expansion in the system's inverse characteristic size.* [8] The density of a property is the ratio of the property to the system volume For example, if A is an extensive property in a small three-dimensional system, the hypothesis states that its density $a = \frac{A}{V}$ can be expanded in $\frac{1}{L}$ as follows:

$$a = \alpha^{\infty} + \frac{\alpha^{8}}{L} + \frac{\alpha^{se}}{L^2} + \frac{\alpha^{sc}}{L^3} + ... \tag{3}$$

Where $L = \sqrt[3]{V}$ is the system's characteristic size, α^{∞} is the value of the density in the large system limit, and the other α-terms are corrections which depend on the shape of the system. In particular, the term α^{8}, which is proportional to $\frac{1}{L}$, is a correction based on the curvature of the system surface. [8]

2.2 Thermodynamics of two-phase systems

A two-phase system consisting of the same substance in different states can be described by a *phase diagram.* An example phase diagram for a vapor/liquid system is shown in Figure 1, with density plotted against temperature. The peak of the curve corresponds to the critical temperature T_c, above which the fluid is supercritical and distinct liquid and vapor phases do not exist. At temperatures below T_c, depending on the density, the system will either consist of a pure liquid phase, a pure vapor phase, or two distinct phases.

In Figure 1, a red line is drawn at the temperature $T_0 < T_c$. The line intersects the curve at $\rho = \rho_0$ and $\rho = \rho_1$. If $\rho < \rho_0$, the system is in the vapor region and exists in a single vapor phase with density ρ. Likewise, if $\rho > \rho_1$, the system will exist in a single liquid phase with density ρ. If $\rho_0 < \rho < \rho_1$, however, the system will consist of a distinct liquid phase, a distinct vapor phase, and an interface between them. In this case, the density of the vapor phase is given by ρ_0, and the density of the liquid phase is given by ρ_1. [13]

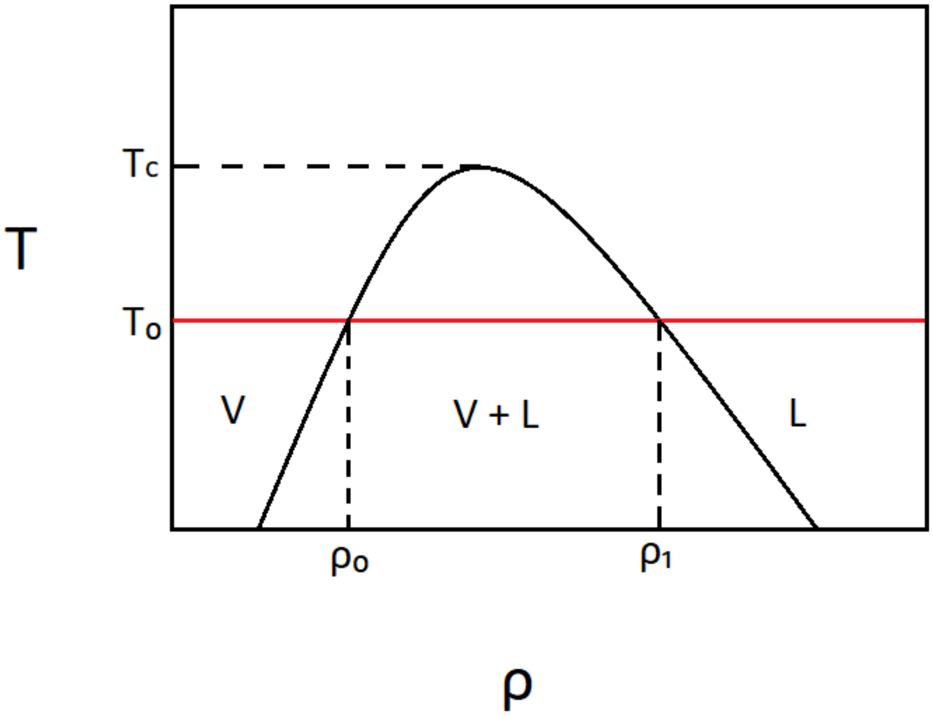


Figure 1: An example of a density-temperature phase diagram. T_c is the critical temperature, the red line represents a system at a given temperature T_0. and the points ρ_0 and ρ_1 are where the T_0-line intersects the phase boundaries. In the two-phase region at T_0, the density of the vapor phase is ρ_0 and the density of the liquid phase is ρ_1.

The surface tension is a force which arises from particle-particle interactions, and which tends to reduce the area of an interface. [14] In large systems, the surface tension is independent from the curvature of the surface. In this case, the relationship between the curvature of the surface and the difference in pressure between the two phases is given by the Young-Laplace equation:

$$\Delta p = \gamma(\frac{1}{r_1} + \frac{1}{r_2}) \tag{4}$$

Where Δp is the difference in pressure between the two phases, γ the surface tension of the interface, and r_1 and r_2 the radiuses of curvature of the interface. [14] If the interface is planar, $r_1 = r_2 = \infty$ and thus $\Delta p = 0$.

If the interface between the two phases is spherical, $r_1 = r_2 = r$ and Equation 4 reduces to:

$$\Delta p = \gamma(\frac{2}{r}) \tag{5}$$

If one were to plot Δp as a function of r^{-1}, the result will be a linear function which goes through zero and has a slope of 2γ.

2.3 Molecular dynamics

Molecular Dynamics (MD) simulation is a classical computational method for determining the thermodynamic properties of a system, by using Newton's laws of motion to calculate how the system changes with time. [9] There are several steps to a MD simulation, and here we will present each of them and their basis.

2.3.1 Initialization

The first step in a MD simulation is defining the system parameters. This includes the size and shape of the simulation box, the potential which will be used to calculate the forces between each pair of particles, the length of each time-step δt, how many time-steps the simulation will last, and the initial position of each particle. Each particle is also assigned an initial velocity at random, taken from a Maxwell-Boltzmann distribution at the given temperature. This velocity will be used for the initial force calculation.

Molecular dynamics simulations are typically performed with a small number of particles, and therefore it's important to avoid small scale effects where they aren't desired. For instance, if the walls of the simulation box were represented by an impassable, infinite potential, particle-wall interactions would have a significant effect on the system properties. [15] To solve this problem, periodic boundary conditions can be used instead of hard walls.

With periodic boundary conditions, the simulation box is treated as if it were part of an infinite grid of boxes. Each box in the grid is a copy of the original box and its contents, shifted integer multiples of the side length in the x, y, and/or z direction(s). In effect, each side, edge, and corner of the simulation box is in contact with its opposite. An illustration of a 2D simulation box with periodic boundary conditions is shown in Figure 2. The box in the center represents the 'real' system, which is surrounded by copies of itself. Only the first copy in each direction is shown. The black circles represent the particles in the system. The red arrow shows how that specific particle will move in the next time-step, the grey circle is its position after movement. As it moves, each of its corresponding copies move with it. When the particle leaves the original box by crossing a periodic boundary, its equivalent from the neighboring copy system enters at the opposite side. This keeps the total amount of particles in the system constant. [15]

2.3.2 Force calculation and potentials

MD is a classical method, and as such the Born-Oppenheimer approximation is used.[15] The motion of the electrons and the motion of the nuclei are assumed to be separable, and the energy of the system to be dependent only on the motion on the nuclei. The energy of the system can therefore be calculated based only on the nuclear positions and momenta. [10] The force between two particles is related to the potential energy between them by the equation:

$$\boldsymbol{F} = -\frac{dU}{d\boldsymbol{r}} \tag{6}$$

Where $\boldsymbol{F}$ is the force, U the potential energy and $\boldsymbol{r}$ the vector between them.[9]

The potential energy from particle-particle interactions depends on the potential used to model them. There exists many different models for describing the interactions between particles, of varying complexity. A common choice for spherically symmetrical particles is the Lennard-Jones model. In this model, the potential energy between each pair of particles is given by the Lennard-Jones (LJ) potential:

$$\phi(r) = 4\epsilon[(\frac{\sigma}{r})^{12} - (\frac{\sigma}{r})^{6}] \tag{7}$$

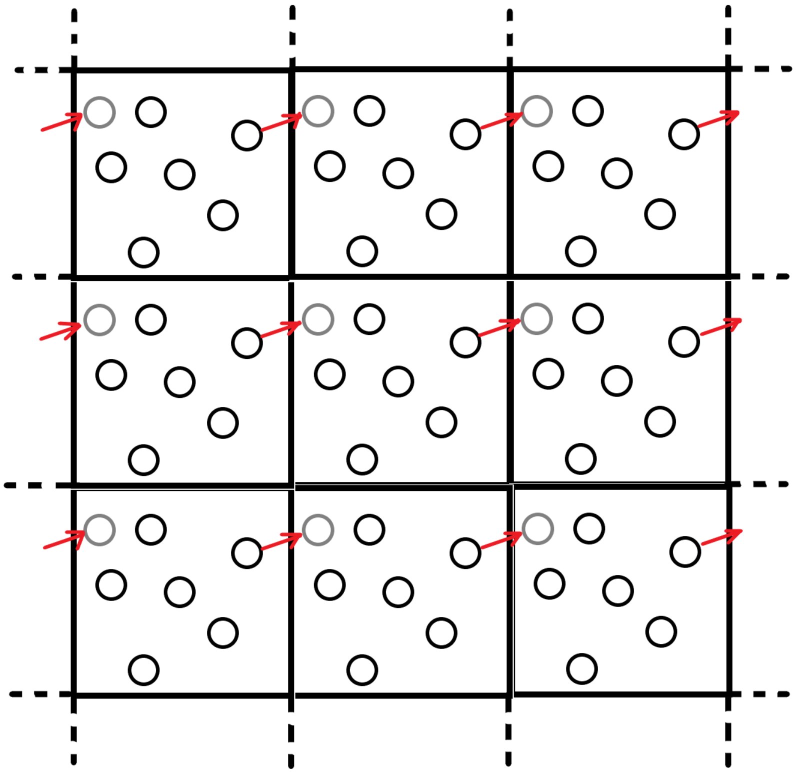


Figure 2: An illustration of a 2D system with periodic boundary conditions. The simulation box is treated as if it were in contact with shifted copies of itself. The black circles represent particles. The red arrow shows how one specific particle will move during the next time-step. As that particle crosses the periodic boundary, its copy enters from the opposite side, keeping the amount of particles in the simulation box constant.

Where r is the distance between the two particles, ϵ the absolute value of the potential between them at its greatest, and σ the distance at which the potential between the particles is zero. The term $(\frac{\sigma}{r})^{12}$ represents the Pauli repulsion, and the term $(\frac{\sigma}{r})^{6}$ represents the attractive Van der Waals forces. [10] Using the LJ potential, Equation 6 becomes:[15]

$$\boldsymbol{F} = \frac{\boldsymbol{r}}{|\boldsymbol{r}|}\frac{24\epsilon}{\sigma}\left[2(\frac{\sigma}{r})^{13} - (\frac{\sigma}{r})^{7}\right] \tag{8}$$

A plot of the LJ potential is shown in figure Figure 3. It should be noted that the LJ potential has a long, near-zero 'tail', and calculating all the very small interactions from this long tail can take up much processing time. Because of this, it's convenient to simply set the forces between a pair of particles to zero if they are far apart. With this change, the LJ potential becomes:

$$\Phi(r) = \begin{cases} \phi(r) & \text{if } r \leq r_c, \\ 0 & \text{otherwise.} \end{cases} \tag{9}$$

Where r_c is the cutoff distance. Equation 9 is called the truncated LJ potential. The truncated LJ potential is discontinuous both in force and in energy at $r = r_c$. The critical temperature and equilibrium density for a truncated LJ fluid is different from that of a fluid modelled by the full LJ potential. [16] It is possible to correct for the discontinuity in energy by adding a constant term, making the energy at $r = r_c$ zero. The constant term disappears under derivation and does not affect the force calculation. A potential with such a correction is called a *truncated and shifted LJ potential.*

The truncation of the potential affects the phase diagram of the simulated LJ system. This effect depends on the cutoff distance r_c, and whether the potential is shifted or not. [16] Figure 4 shows phase diagrams for both a truncated (dotted line) and a truncated and shifted LJ potential (dashed line), both with a cutoff distance $r_c = 2.5$.

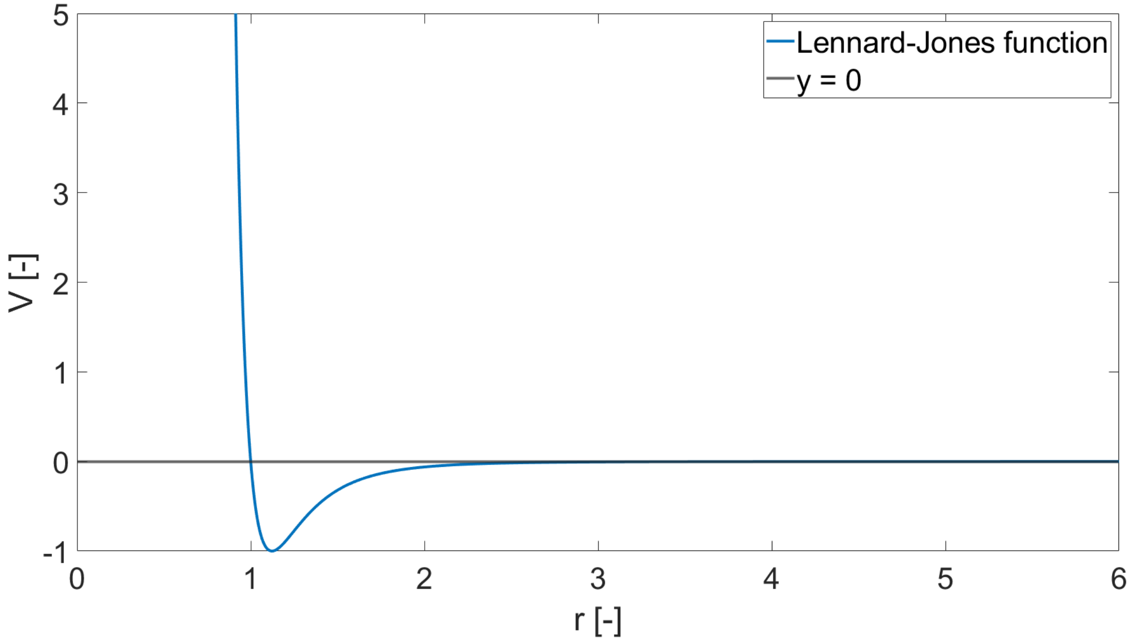


Figure 3: A plot of the Lennard-Jones function for $\epsilon = \sigma = 1$. r and V are given in arbitrary units. The Lennard-Jones function describes the sum of Pauli repulsion and Van der Waals attraction. The potential is zero for $x = \sigma$, and the lowest value of the potential is $-\epsilon$. The Lennard-Jones potential suddenly rises like a wall as the distance between particles gets small enough. For large distances, the potential is close to zero.

2.3.3 Reduced units

Molecular dynamics simulations utilizing the Lennard-Jones model are typically done using *reduced units*, also called Lennard-Jones units. Reduced units are defined by choosing fundamental units of energy, length, and mass, and all quantities are then expressed as functions of these fundamental units. [9] It is convenient to choose the fundamental unit of energy to be ϵ, the unit of length to be σ, and the unit of mass to be the particle mass m. For example, the unit of length r^* is defined as $\frac{r}{\sigma}$, where the asterisk superscript is used to denote a reduced quantity. A full list of the reduced units relevant for this project and their definitions can be found in Table 4 in the appendix.

Working with reduced units is convenient, as it avoids having to deal with extremely small or large numbers. In addition, using reduced units results in a more general system model. According to the law of corresponding states, there is universal behavior between different systems at the same state in reduced units. [13] To obtain results for a particular system in real units, for example for neon, the fundamental values ϵ, σ, and m are simply replaced with experimental values for neon.

2.3.4 Calculating the new velocities

From the forces acting on each particle, each particle's acceleration is determined using Newton's second law:

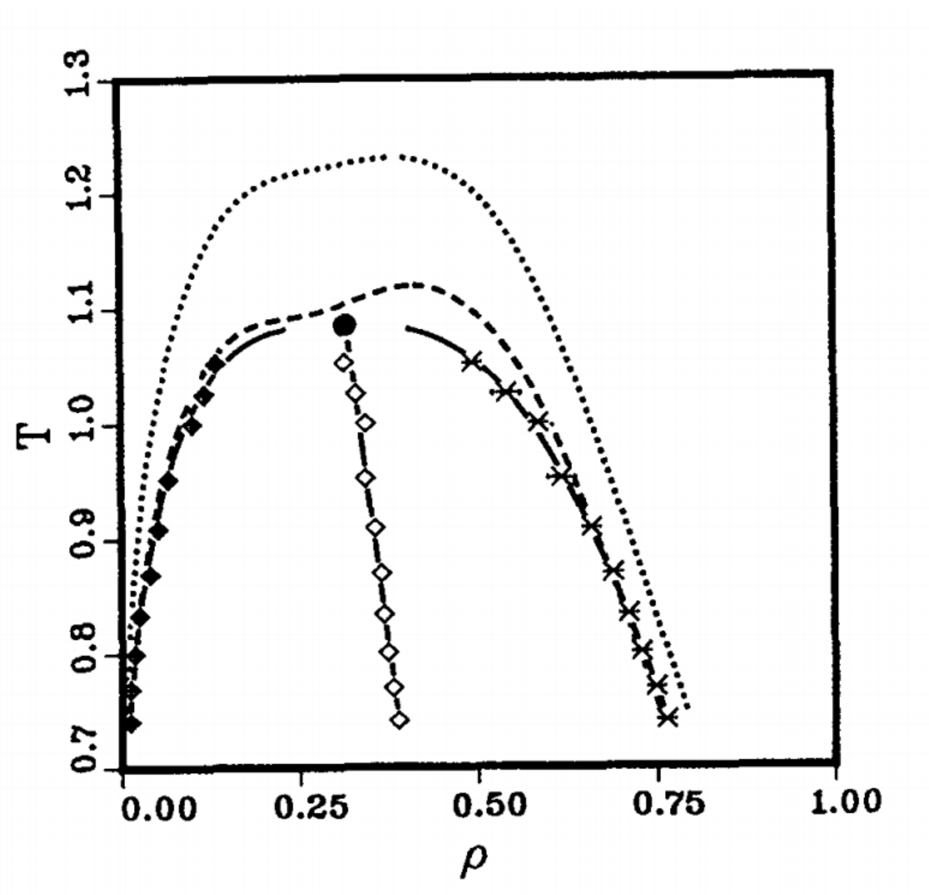


Figure 4: This image shows phase diagrams for a LJ fluid, where the potential is truncated at a distance of 2.5. The dotted line shows an estimated phase diagram for a truncated and not shifted LJ fluid. The solid line shows a phase diagram for a truncated and shifted LJ fluid, calculated from data obtained by Smit. [16] The dashed line represents the equation of state determined by J.J.Nicholas et al. [17] Reprinted from Phase diagrams of Lennard-Jones fluids by Smit[16], with the permission of AIP Publishing.

$$\frac{d^2\boldsymbol{r}_i}{dt^2} = \frac{\boldsymbol{F}_i}{m_i} \tag{10}$$

Where F_i is the force exerted on particle i, $\boldsymbol{r}_i$ the vector along which the force on particle i is directed, and m_i the mass of particle i.[15]

The next step is calculating each particle's position and velocity after an amount of time δt. This is done through numeric integration of the laws of motion. The algorithm used for the integration is called the *integrator*. In general, numeric integration algorithms rely on the assumption that particle positions and movement can be approximated by Taylor expansions.[15] For the position, this gives:

$$r(t + \delta t) = r(t) + \delta t v(t) + \frac{1}{2}(\delta t)^2 a(t) + ... \tag{11}$$

And for the velocity:

$$v(t + \delta t) = v(t) + \delta t a(t) + ... \tag{12}$$

A good integrator must fulfill two criteria: It must be *area-preserving*, and *time-reversible*. The Verlet family of integrators satisfies both criteria.[15] One of them, the *velocity Verlet* integrator, will be described further. The velocity Verlet algorithm uses the following two equations, based on the Taylor expansions shown in Equation 11 and Equation 12:[15]

$$r(t+\delta t) = r(t) + \delta t v(t) + \frac{1}{2}\delta t^2 a(t) \tag{13}$$

$$v(t+\delta t) = v(t) + \frac{1}{2}\delta t[a(t) + a(t+\delta t)] \tag{14}$$

Since equation 14 uses the acceleration at $t+\delta t$, the velocity Verlet method is implemented in three steps. First, equation 13 is used to calculate the positions at $t+\delta t$. Second, the velocities at time $t+\frac{1}{2}\delta t$ are calculated by:

$$v(t+\frac{1}{2}\delta t) = v(t) + \frac{1}{2}\delta t a(t) \tag{15}$$

Next, new forces are calculated from the positions at $t+\delta t$, which gives the accelerations at $t+\delta t$. The third and final step uses:

$$v(t+\delta t) = v(t+\frac{1}{2}\delta t) + \frac{1}{2}\delta t a(t+\delta t) \tag{16}$$

to calculate the velocities at $t+\delta t$. [15]

2.3.5 Calculating ensemble properties

The procedure of calculating new forces and new positions is repeated for as many time-steps as was specified during the initialization. Properties of the system can be measured at specified time intervals. For a general property A, its measured value at time t will be a function of the system particle positions and momenta:

$$A = A(\boldsymbol{p}^N(t), \boldsymbol{r}^N(t)) \tag{17}$$

Where N is the total amount of particles, $\boldsymbol{p}^N$ the particle momenta and $\boldsymbol{r}^N$ the particle positions.[15] The "true value" of the property, A_{avg}, is found in the limit where A is measured an infinite amount of times:[15]

$$A_{avg} = \lim_{M\to\infty} \frac{1}{M}\int_{t=0}^{M} A(\boldsymbol{p}^N(t), \boldsymbol{r}^N(t))dt \tag{18}$$

Where M is the number of times A is measured. By the ergodic hypothesis, the time average of A is equal to the ensemble average of A:[15]

$$A_{avg} = \langle A \rangle \tag{19}$$

In a molecular dynamics simulation, the ensemble average of a given property is calculated numerically:[15]

$$\langle A \rangle = \frac{1}{M}\sum_{i=1}^{M} A(\boldsymbol{p}^N, \boldsymbol{r}^N) \tag{20}$$

Ideally, each state measured should not be correlated with the previous one. For this reason, the measurements are not performed every time-step, but instead every X time-steps. For example, with $X = 100$, to obtain 10 000 snapshots of a system a simulation of 1 000 000 time-steps is needed. Here, there is a trade-off between total simulation time and correlation as a result of too short a time between measurements. There exists methods to deal with the effects of data point correlation, such as block analysis which is described in Section A.2.

In summary, a molecular dynamics simulation goes as follows. The system parameters are defined. Next, the force experienced by each particle is calculated. Using the new forces. the position and velocity of each particle after a time δt is calculated. This process of force- and position calculation is repeated for as many times as specified. At specified points in the simulation, specified properties of the system are measured. These measurements are used to calculate ensemble properties of the system.

2.3.6 Thermostats

The molecular dynamics scheme presented above has a constant number of particles and constant system volume, while the time average of the energy is equal to the ensemble average. As such it simulates the NVE ensemble. [9] It's also possible to simulate the NVT ensemble. First, we consider the temperature of the system, which is related to the average kinetic energy:

$$\langle E_k \rangle = \frac{k_B T}{2}(3N - N_c) \tag{21}$$

where k_B is the Boltzmann constant, T the temperature, N the total number of particles, N_c the number of constraints on the system, and E_k the kinetic energy, which is distributed according to a Boltzmann distribution.[15] In other words, were the temperature set to be strictly equal to a constant would also make the average kinetic energy exactly constant. This does not adequately represent the real NVT ensemble.[9] Instead, to simulate the NVT ensemble, the temperature - and thus also the average kinetic energy - of the system must be allowed to fluctuate around the assigned value.

The algorithms used for this are called *thermostats*. One such algorithm is the Nosé-Hoover thermostat.[18] It simulates a thermal reservoir in contact with the system, allowing energy to flow between the system and the reservoir, and making the temperature fluctuate. This reservoir is represented by introducing an additional degree of freedom s to the Hamiltonian of the system. [9]

2.4 Voronoi tessellation

Voronoi tessellation is a method for dividing a section of space into smaller, uniquely defined subsections. The procedure takes a set of points in an area or a volume, and computes a characteristic area or volume for each point, called a *Voronoi cell*. The Voronoi cell of any given point contains the space which is closer to that point than to any other. [11] The walls of the Voronoi cell are hyperplanes which are equally close to two particles, and the set of hyperplanes define the subsections.[19]

An illustration of a Voronoi tessellation performed in two dimensions is shown in Figure 5. The diagram was created using a program made by Frederik Brasz.[20] In this case, the hyperplanes which form the Voronoi cell walls are lines, and each line is shared with another Voronoi cell. Two particles which share Voronoi a cell wall in such a way are called *Voronoi neighbors*. For example, in the system shown in Figure 5, particles $B1$ through 5 are Voronoi neighbors with particle A, the Voronoi cell edges of which are colored red.

A line is drawn between the points marked C and E. This line is normal to the hyperplane between points C and E, and point D where the line and hyperplane intersects is equally close to C and E. Any points on the hyperplane will be equidistant from C and E. At its topmost end point, where it meets the hyperplanes between C and $B4$ and between E and $B4$, it is equally spaced between C, E and $B4$.

In three dimensions, the Voronoi cell walls will be planes which are equidistant from two specific points. If a group of points in a volume represent the centers of mass of a group of particles, the volumes of the resulting Voronoi cells can be interpreted as the volume occupied by each particle. This volume is called the particle's *Voronoi volume*.[12] The *Voronoi density* of a particle is the density calculated by using the particle's Voronoi volume:

$$\rho_v = \frac{m}{V_v} \tag{22}$$

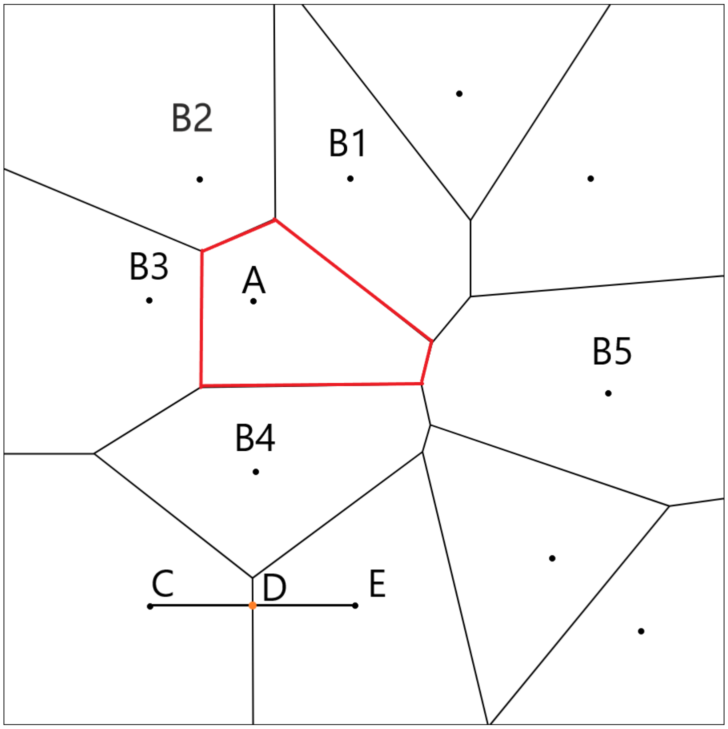


Figure 5: A Voronoi tessellation performed on twelve points in the plane. The lines separate the individual Voronoi cells. The space within a Voronoi cell is closer to the contained point than any other point, each point on the line separating two cells is equidistant. [11] The points $B1$ through 5 share a Voronoi cell wall with point A, making points $B1$ through 5 Voronoi neighbors with point A. D is the point where the line between points C and E intersects the hyperplane between them, which is exactly halfway between points C and E: $2CD = 2DE = CE$. [19]

Where m is the mass of the particle. In reduced units, 22 becomes simply:

$$\rho_v^* = \frac{1}{V_v^*} \tag{23}$$

In Figure 5, it should be noted that many of the Voronoi cells - like those containing points C and E - are not fully closed, but simply stop at the edges of the box. As with other boundary effects, however, this is not a problem if periodic boundary conditions apply. In that case, the Voronoi cells effectively stretch across the boundaries of the box instead of stopping there, making them continuous across the simulation box boundaries.

2.5 The Voronoi density-neighbor method for defining the phases of a two-phase system

2.5.1 The Fern method for determining phase densities from a two-phase system

A liquid-vapor system, at a temperature well below the critical temperature, can be considered to consist of three types of particle: Liquid particles, gas particles, and surface particles. Fern et al.[12] have developed a method for characterizing liquid-vapor systems using the Voronoi volume of each particle, as defined in Section 2.4. The Voronoi volumes can easily be obtained from a molecular dynamics simulation.

The molar volume of a particle in a system describes how much volume that particle tends to occupy. For pure substances the average molar volume is simply defined as $\frac{V}{N}$, where V is the total system volume and N the total amount of particles.[13] The molar volume of a particle can be approximated by the volume of that particle's Voronoi cell.[12] Since a vapor particle will have a greater molar volume than a liquid particle, it can be expected that a particle with a large Voronoi volume - and thus a small Voronoi density - is a vapor particle. Similarly, a particle with a small Voronoi volume (and large Voronoi density) is most likely a liquid particle.

In Figure 6, the Voronoi densities of particles in a planar liquid-vapor system are shown, with the Voronoi density defined as in Equation 23. The plot has two peaks, a smaller one at low Voronoi density and a larger one at It can be deduced that the small leftmost represents the gas particles and the large rightmost represents the liquid particles. The 'belt' in between represents surface particles, which are neither liquid nor vapor.[12]

The Fern method classifies a given particle as vapor if it has Voronoi density lower than a parameter X_{low}, as liquid if it has Voronoi density higher than a parameter X_{high}, and as surface if its Voronoi density is somewhere in between. [12] From Figure 6, one can already make a reasonable guess as to what those values should be for this system. The Fern method involves first guessing X_{low} and X_{high} from a plot of a density distribution like in Figure 6.[12] Next, the guesses are tested by comparing the variance of the average Voronoi density and with the variance of the average Voronoi density of a one-phase system of the same density, for both the liquid and the vapor phase. If these variances are equal, the cutoff parameter is good. Otherwise, another guess is made for X_{low} and/or X_{high} and the test is performed again. Another option is to compare the average Voronoi densities directly.

It was noted by Fern et al. that classifying a system according to each particle's Voronoi density results in 'surface' particles concentrated at the interface, but also spread throughout the system. [12] Figure 7 shows a visualization of a droplet-vapor system which was sorted by the Fern method. Particles labeled as vapor are red, particles labeled liquid are yellow and particles labeled surface are blue. It can be seen that there are many vapor particles which are mistakenly labeled as surface.

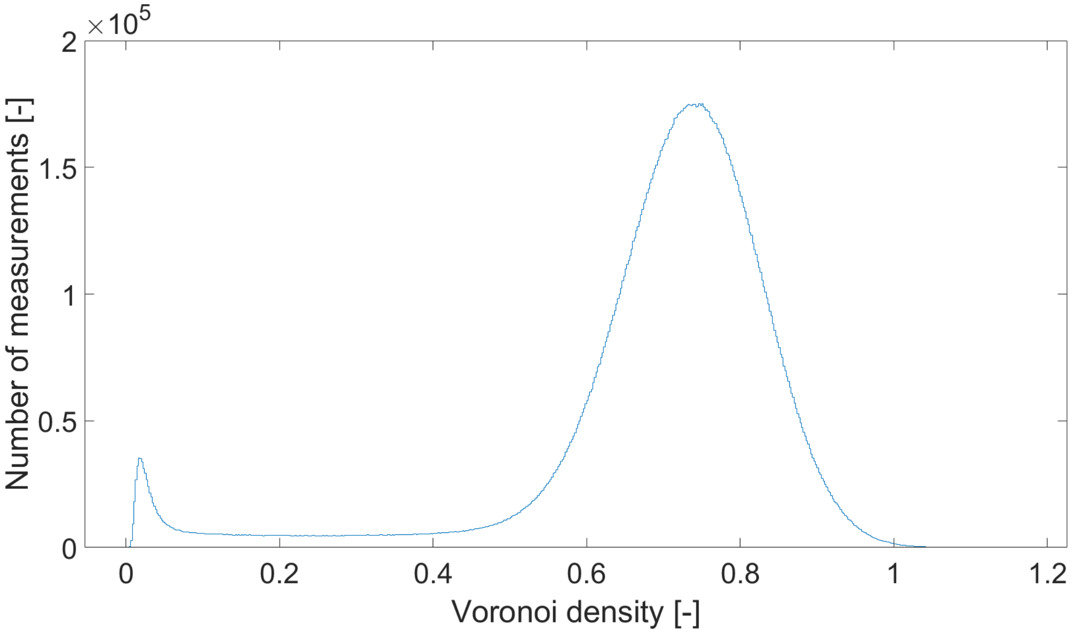


Figure 6: A histogram over the Voronoi densities of particles in a two-phase system with a planar interface. The leftmost peak corresponds to gas particles, the rightmost peak to liquid particles, with the surface particles located on the 'saddle' between them.

We have called our method for classifying droplet-vapor systems the *Voronoi density-neighbor method.* It consists of two steps. First the Fern method which was just described is applied, this is called the *density step*. The second step was devised to eliminate the falsely labeled particles. It is called the *neighbor step* and will be described in the following section.

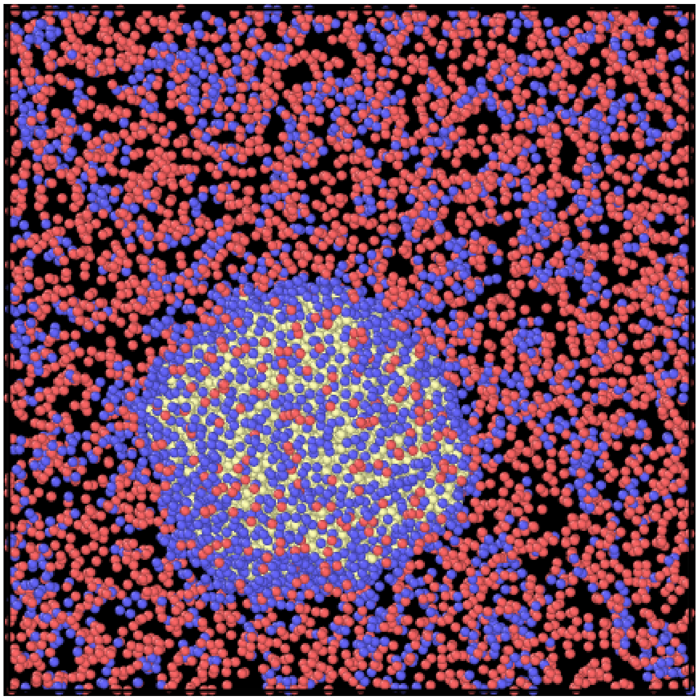


Figure 7: A visualization of a droplet-vapor system sorted by the Fern method. Vapor particles are labeled red, liquid particles yellow and surface particles blue. Many of the particles labeled as surface are in fact spread throughout the vapor phase. The visualization was created using OVITO.[21]

2.5.2 The neighbor step

As motivation for the neighbor step, we can consider three particles classed as 'surface' by the density step. The first particle $P1$ is actually a vapor particle, while the second particle $P2$ is a true surface particle, and the third particle $P3$ is really a liquid particle. A simple illustration of this situation is shown in Figure 8. The vapor phase is marked in red, the surface in blue and the liquid phase in green. Marked in purple and labeled are $P1$, $P2$ and $P3$. If we were to perform a Voronoi tessellation on such a system, the Voronoi cell of a mislabeled vapor particle like $P1$ would mostly share its cell walls with other vapor particles, and the Voronoi cell of a mislabeled liquid particle like $P3$ would share cell walls with mostly other liquid particles. A real surface particle like $P2$ will tend to share Voronoi cell walls with a mix of surface, liquid and vapor particles.

The idea is therefore to further examine all 'surface' labeled particles. For each, the *vapor neighbor proportion* and *liquid neighbor proportion* is calculated. A particle's vapor neighbor proportion is the area of its Voronoi cell walls which are shared with vapor particles, divided by the total area of its Voronoi cell walls. In the same manner, a particle's liquid neighbor proportion is the area of its Voronoi cell walls which are shared with liquid particles, divided by total Voronoi cell wall area. If a 'surface' labeled particle's vapor neighbor proportion is greater than a parameter Z_{vap}, then its label is changed to vapor. If its liquid neighbor proportion is greater than a parameter Z_{liq}, its label is changed to liquid. Otherwise, the particle remains labeled as surface.

Figure 9 shows visualizations of a droplet-vapor system before and after the neighbor step was applied. Figure 9a shows the before case while Figure 9b shows the after case. It can be seen that after the neigh-

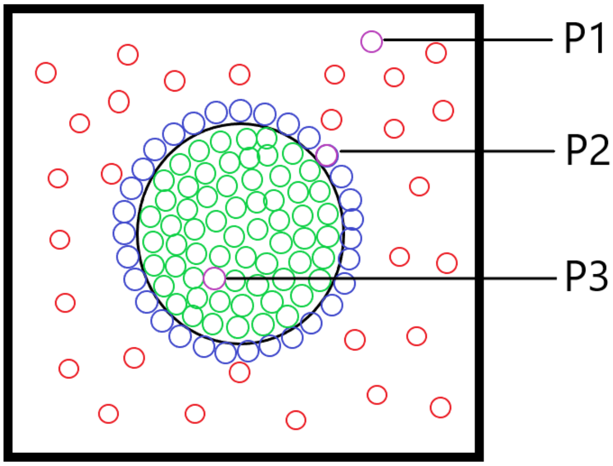
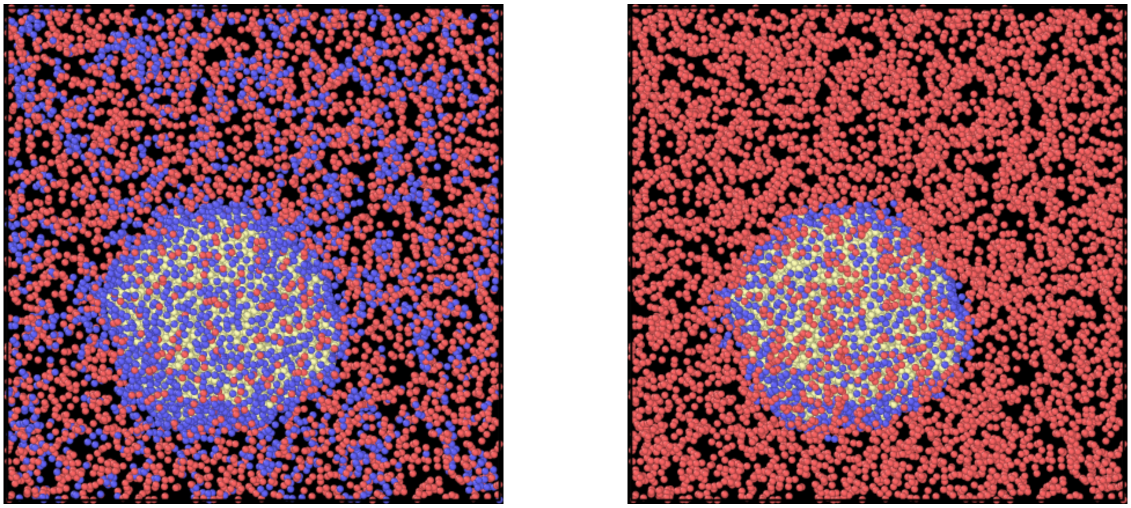


Figure 8: An illustration of a 2D cross-section of a droplet-vapor system. The red circles represent vapor particles, the blue circles surface particles and the green circles liquid particles. The purple circles represent three particles which are classified as 'surface' by the Fern method. Particle P1 is actually in the vapor phase and neighbors mostly other vapor particles. Particle P2 is located in the surface and neighbors a mixture of vapor, liquid and surface particles. Particle P3 is actually in the liquid phase and neighbors mostly other liquid particles.

bor step, the 'surface' particles in the vapor phase have disappeared; they are now correctly labeled as vapor.

It is possible to perform the neighbor step multiple times, each time updating the lists of liquid and vapor particles. This means there are three parameters in total to select for the neighbor step: the proportion limits Z_{vap} and Z_{liq}, and the number of step iterations N. A summary of the neighbor step is shown in Figure 10.



(a) Sorted by Fern method

(b) Sorted by Fern method and neighbor sorting

Figure 9: Visualizations of System 4 sorted in two different ways: With only the Fern method, and with the Fern method and the neighbor step. In the latter case, virtually all the clearly mislabeled vapor particles are now correctly labeled as vapor. The visualizations were created using OVITO.[21]

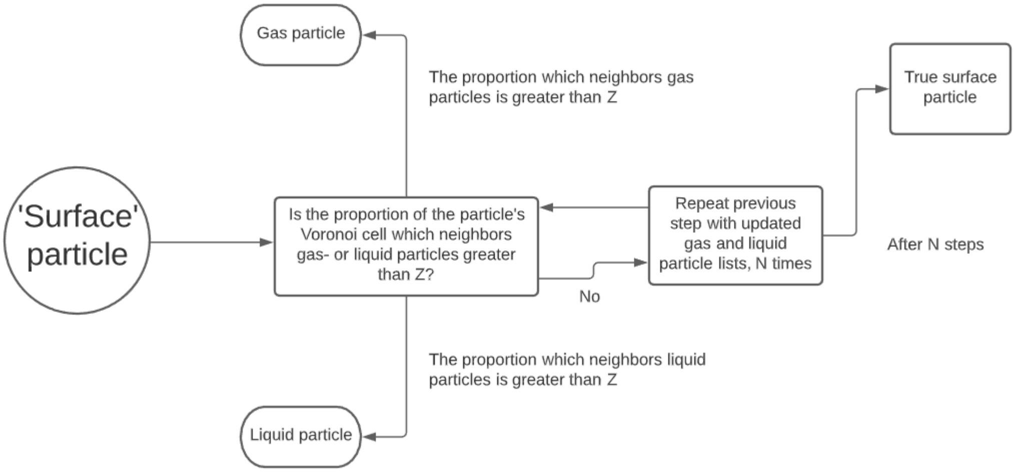

Gas particle
The proportion which neighbors gas particles is greater than Z
True surface particle
'Surface' particle
Is the proportion of the particle's Voronoi cell which neighbors gas- or liquid particles greater than Z?
Repeat previous step with updated gas and liquid particle lists, N times
After N steps
No
The proportion which neighbors liquid particles is greater than Z
Liquid particle

Figure 10: A schematic of the Voronoi neighbor sorting method.

3 Process

3.1 Common parameters

All the molecular dynamics simulations were performed with LAMMPS[22], an open source software. LAMMPS is available at http://lammps.sandia.gov. System visualizations were created using the open-source software OVITO[21].

Unless otherwise specified, all named quantities are in Lennard-Jones units, as defined in Table 4. Each time-step is of length 0.002. The potential between each pair of particles is modeled by a truncated Lennard-Jones potential with a cutoff distance of 2.5, as described in Section 2.3.2.

The scripts used in this project are provided at https://github.com/haavarfa/droplet_vapor_project.

3.2 Creating and equilibrating the droplet-vapor systems

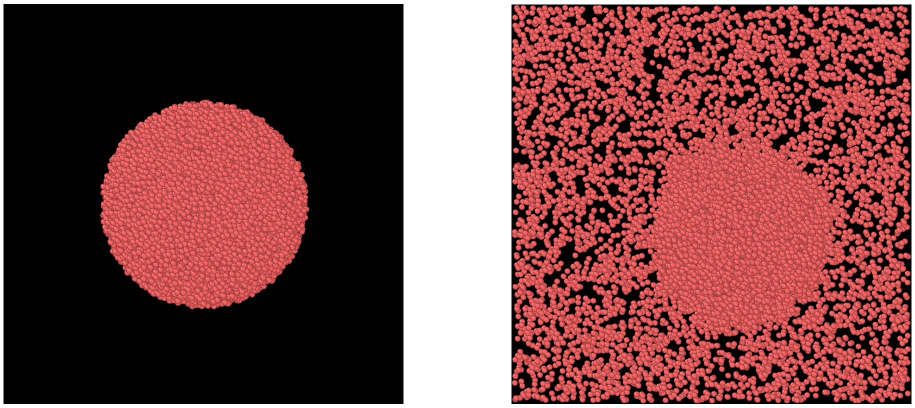
A total of ten droplet-vapor systems were simulated, with varying initial droplet radius. A list of the droplet-vapor systems is shown in Table 1. For efficiency, creation of each of the droplet-vapor systems was performed in two steps. In step one, a simulation box filled with liquid was created. In step two, a liquid droplet with initial radius as listed in Table 1 was created from the liquid-filled box, and the system progressed until it reached an equilibrium state.

Step one was performed in the following fashion. First, a simulation box in the shape of a cube with side length 66 was created. The simulation box was filled with a fcc lattice of particles, with density 0.8. To melt the lattice of particles, a MD simulation was performed for 200 000 time-steps, in the NVT ensemble at a temperature of 10. The final state of the system after simulation was recorded, and was used as a starting point to create each of the ten droplet-vapor systems.

Table 1: An overview of the droplet-vapor systems and their initial radius.

System number	Initial radius
1	14
2	15
3	16
4	17
5	18
6	19
7	20
8	21
9	22
10	23

In step two, all particles in the melted system were deleted, except for those contained in a spherical region centered at the center of the simulation box. A visualization of System 4 right after particle deletion is shown in Figure 11a. At this point, the system is essentially a droplet in vacuum, and not stable. Next, a MD simulation was performed in the NVT ensemble for as many time-steps as needed for the system to reach equilibrium, at a temperature of 0.8. As each of the simulations progressed, part of the liquid droplet evaporated. As the temperature is well below the critical point of the LJ fluid, as shown in Figure 4, this continued until each system reached equilibrium. Figure 11b shows System 4 after it was equilibrated.



(a) System 4 after initial droplet creation.

(b) System 4 at equilibrium.

Figure 11: Visualizations of System 4, right after creation of the initial droplet and after the system reached equilibration. Right after creation (a), the system is effectively a droplet of liquid in a vacuum. This is obviously not a stable system, and in subsequent time-steps particles will evaporate from the droplet until an equilibrium is reached (b).

The amount of time-steps sufficient to reach equilibrium for each system was determined in the following way. First, a MD simulation of the system was performed in the NVT ensemble for X time-steps. During this simulation, the total energy of the system was calculated every 1000th time-step. Afterwards, the time evolution of the total energy of the system was examined, by plotting the potential energy as a function of time.

In addition, the following mathematical test was employed. The average total energies for the last and second-to-last blocks of 300 measurements were calculated. If the difference between the two averages was smaller than twice the standard deviation of the second-to-last block, and the plot of potential energy appeared to have reached a point of fluctuating around a value, X time-steps were deemed sufficient to equilibrate the system. Otherwise, the procedure was performed again with a larger X.

After equilibration, the final step was preparing the system for transitioning to the NVE ensemble. As described in Section 2.3.6, the temperature in a NVT simulation fluctuates around the set temperature. At the moment the thermostat is 'switched off', the temperature will therefore not be exactly equal to the set temperature. As the average kinetic energy of the system scales with temperature, this could cause the temperature in the following NVE simulation to change significantly. To correct for this effect, after equilibration the kinetic energy of the system was set to be equal to the average kinetic energy, by rescaling the particle velocities.

3.3 Creating the reference system

In addition to the ten droplet-vapor systems, a two-phase vapor-liquid system with a planar interface was created. Such a system can be considered as a droplet with infinite radius and an inverse radius of zero, and was used as a reference system to investigate potential curvature effects.

To make the reference system, a rectangular simulation box of dimensions 30x30x60 was created, and one half of the simulation box was filled with a fcc lattice of particles with density 0.8. The system was equilibrated by MD simulation of the system in the NVT ensemble at a temperature of 0.8, for a length of time which was determined as for the droplet-vapor systems, but with the total energy being measured every 100th time-step instead of every 1000th.

3.4 Performing measurements on the systems

After the droplet-vapor and planar interface systems were judged to have reached equilibrium, additional MD simulations were performed on each, for the purpose of measuring data. Five NVE simulations of 20 100 time-steps in length were performed for each system, for a total of 100 500 time-steps. Properties were calculated every 100th time-step.

The position, potential and kinetic energy, Voronoi volume, and stress tensor of each particle was calculated. Additionally, for the particles classified as surface by their Voronoi density, by the method described in Section 2.5.1, the particle's Voronoi neighbors and area of the cell face shared between them was calculated. This calculation was limited to the 'surface' labelled particles in order to limit file size and save processing time.

3.5 Defining the density step limits

The data from each system was analyzed with the Voronoi density-neighbor method, as described in Section 2.5. The parameters X_{low} and X_{high} were determined by comparing the average densities directly. The bulk liquid- and vapor density for a truncated LJ fluid with $r_c = 2.5$ was taken from the work of Nijmeijer et al[23]. They reported that at T = 0.8, the liquid density is 0.7315 with an error of 0.0003, and the vapor density is 0.0195 with an error of 0.0007.[23]

A total of four different ways to define X_{high} were tested. In the first three methods, X_{high} was defined so the liquid phase achieved bulk density after the density step, but before the neighbor step. In the *largest system method*, X_{high} was set so System 10 achieved bulk liquid density. In the *reference system method*, X_{high} was set so the simple planar system achieved bulk liquid density. In the *per-system pre-neighbor method*, a range of X_{high} values was used, with $X_{high}i$ defined so system i achieved bulk density. Finally, in the *per-system post-neighbor method*, a range of X_{high} values was used like in the per-system pre-neighbor method, but the X_{high} values were defined so the liquid phase achieved bulk density after the neighbor step. In all cases, X_{low} was defined by System 10, as it was assumed to not change significantly.

3.6 Droplet counting

A possible utility of the density-neighbor method is the ability to determine the number of separate droplets present in a liquid-vapor system. In a system which is close to the phase boundary, for example, it is conceivable that droplets constantly form and dissolve. A droplet-counting method could determine if this phenomenon is taking place.

As a demonstration of this possibility, a simple iterative algorithm was implemented. The algorithm was performed in the following manner: First, a random unmarked surface particle is marked as belonging to group number 1. Next, any unmarked particles which is a Voronoi neighbor to a particle marked with 1 is also marked as belonging to group 1. This step is repeated until no new particles are marked. At this point, if the number of particles marked as belonging to group 1 is greater than a given limit, the number of droplets present is incremented by one. Finally, if the total number of particles that have been marked is less than the total number of surface particles, another iteration is performed. A random unmarked surface particle is marked as belonging to group 2, and the process is repeated, with the marking number increasing by 1 every iteration.

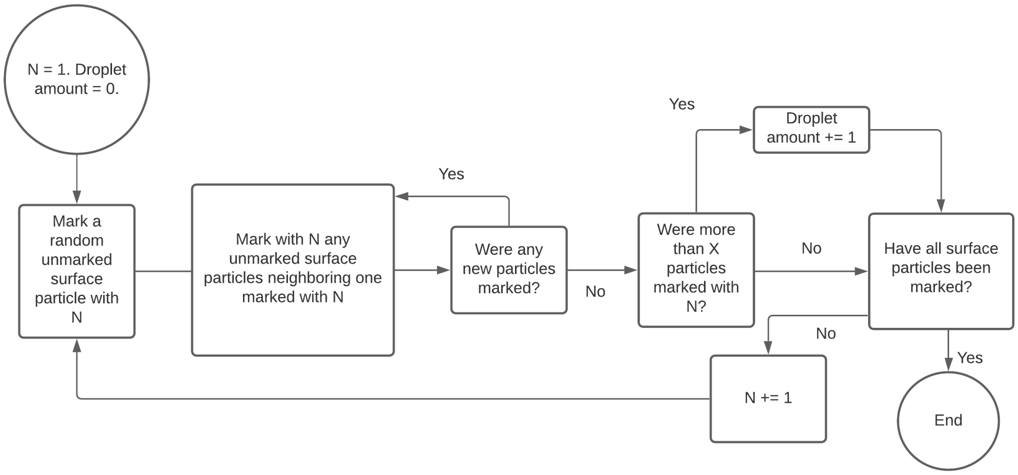

N = 1. Droplet amount = 0.
Mark a random unmarked surface particle with N
Mark with N any unmarked surface particles neighboring one marked with N
Yes
Were any new particles marked?
No
Were more than X particles marked with N?
Yes
Droplet amount += 1
No
Have all surface particles been marked?
No
N += 1
Yes
End

Figure 12: A schematic of the algorithm implemented for counting the number of droplets present.

When the number of marked particles is equal to the number of surface particles, the algorithm is finished. Figure 12 shows a diagram of the algorithm.

To test the droplet counting algorithm, a system consisting of two droplets in equilibrium with a vapor phase was created. A simulation box filled with a bcc lattice of particles was created and melted as in Section 3.2, with side length 99. Two spherical droplets were cut out, each with radius 15 and centered on the coordinates (24.5, 24.5, 24.5) and (74.5, 74.5, 74.5) respectively, after which the system was equilibrated.

4 Results and discussion

All errors shown were calculated using block analysis, as described in Section A.2, with 5 blocks.

4.1 System equilibration

The length of time required to equilibrate each system was found to depend on the initial droplet radius - the smaller the initial droplet, the longer it takes to reach equilibrium. The lengths in time steps deemed sufficient to equilibrate systems 1 through 10, as well as the simple planar system, are shown in Table 2.

The droplet radius at equilibrium was calculated by the standard equation for the volume of a sphere, using the total Voronoi volume of every liquid particle, and assuming the droplets can on average be considered spherical. The equilibrium radius of each droplet is also shown in Table 2, along with the error.

Table 2: The amount of time-steps sufficient to equilibrate each droplet-vapor system and the planar interface system. As the initial radius decreases, more time is required to reach equilibrium.

System number	Time-steps for equilibration	Initial radius	Equilibrium radius	Radius error
1	11 000 000	14	8.3485	0.0047
2	3 000 000	15	10.8887	0.0070
3	3 000 000	16	12.5460	0.0024
4	2 500 000	17	14.2153	0.0106
5	1 000 000	18	15.8130	0.0051
6	1 000 000	19	17.0506	0.0076
7	1 000 000	20	18.3517	0.0071
8	1 000 000	21	19.5988	0.0097
9	1 000 000	22	20.8112	0.0071
10	1 000 000	23	21.9415	0.0102
Simple planar	N/A	500 000	N/A	N/A

4.2 Determining phase definition parameters

As mentioned in Section 3.5, the bulk density values for the simulated LJ potential were taken from Nijmeijer et al.[23] We originally believed the values Nijmeijer et al. provided were for a truncated and not shifted LJ potential. This turned out to be wrong: Trokhymchuk et al.[24] cite the values of Nijmeijer et al. as being for a truncated and shifted potential, and provide different values for the truncated and shifted case. Table 3 shows the bulk densities as well as the surface tension for both cases.

This error was discovered quite late in the process, and so we decided to go ahead with using the truncated and shifted values provided by Nijmeijer et al. This obviously has an effect on the X parameters - namely, X_{high} will be too low and X_{low} will be too high.

Table 3: Liquid density, vapor density and surface tension for a LJ fluid truncated at 2.5 at $T = 0.8$. Values for the truncated and not shifted case are taken from Trokhymchuk et al.[24] Values for the truncated and shifted case are taken from Nijmeijer et al.[23]

System type	Liquid density	Error	Vapor density	Error	Surface tension	Error
Truncated and not shifted	0.7653	N/A	0.0122	N/A	0.623	0.028
Truncated and shifted	0.7315	0.0003	0.0195	0.0007	0.39	0.01

To compare the different ways to define X_{high}, as described in Section 3.5, each of the three first methods were applied on four of the droplet-vapor systems, system 1, 4, 7 and 10, as well as on the simple planar reference system. The resulting liquid phase densities are shown in Figure 13. For each method, the liquid phase density decreases with decreasing droplet size, an effect caused by the neighbor step including more particles of low density. The decrease is greater with the largest system method and the reference system method than with the per-system pre-neighbor method.

The per-system post-neighbor method was found too time-consuming to feasibly implement for all but the smallest droplet-vapor system. As such, data from it is not shown here. Were it successfully implemented, each liquid phase density would per definition be equal to the bulk liquid density.

Figure 14 shows histograms of the liquid phase densities defined by the largest system method, while Figure 15 shows the same defined by the per-system pre-neighbor method. Each histogram is normalized by probability. It can be seen that for each system, there is a sudden drop at $X = X_{high}$. Here we also see that the systems with smaller droplet have proportionally fewer particles of high density, and that this decrease is greater with the largest system method than with the per-system pre-neighbor method.

The drop represents a discontinuity in the density distribution, and the size of the drop can be seen to correspond to the size of the deviation in average Voronoi density from the planar interface case. With both the largest system method and the per-system pre-neighbor method, the closer the average density of the system is to the planar interface case, the more symmetrical the distribution is. Based on this, it was decided to use the per-system pre-neighbor method to define X_{high} in this project.

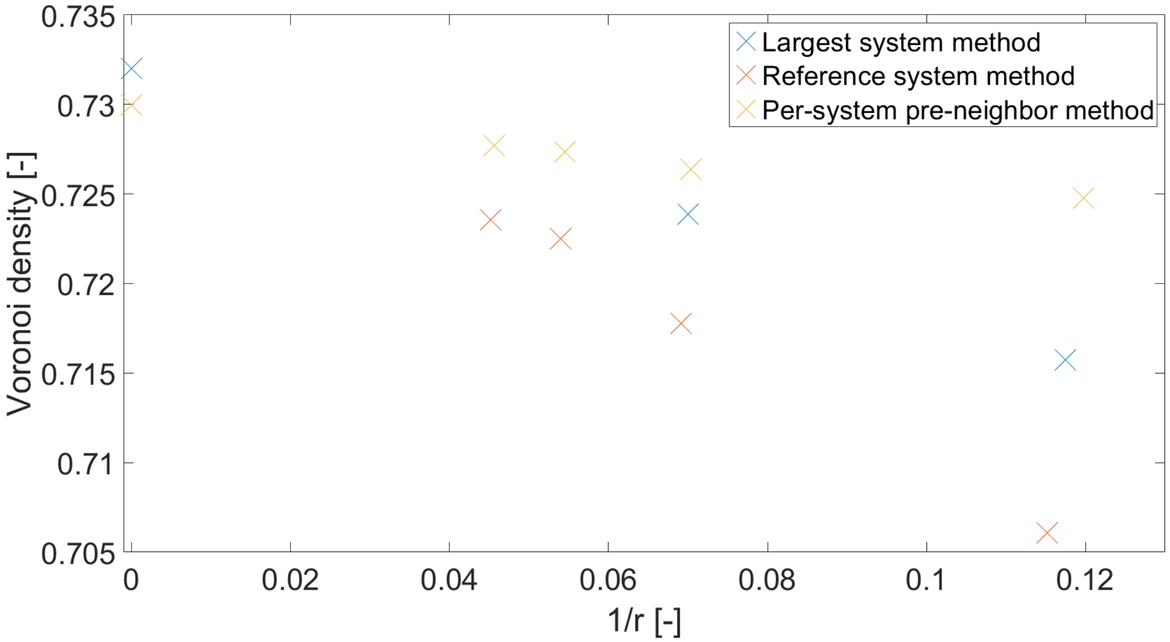


Figure 13: Liquid phase density as a function of r^{-1}, with the cutoff limit X_{high} defined in three different ways: By using the largest system, by using the planar interface reference system, or by using a range of different X_{high}s. For the first point the results from the reference system method and per-system pre-neighbor method coincide, while for the second and third points the results from the largest system method and the per-system pre-neighbor method coincide. The per-system pre-neighbor method clearly gives the lowest decrease in density with r^{-1}.

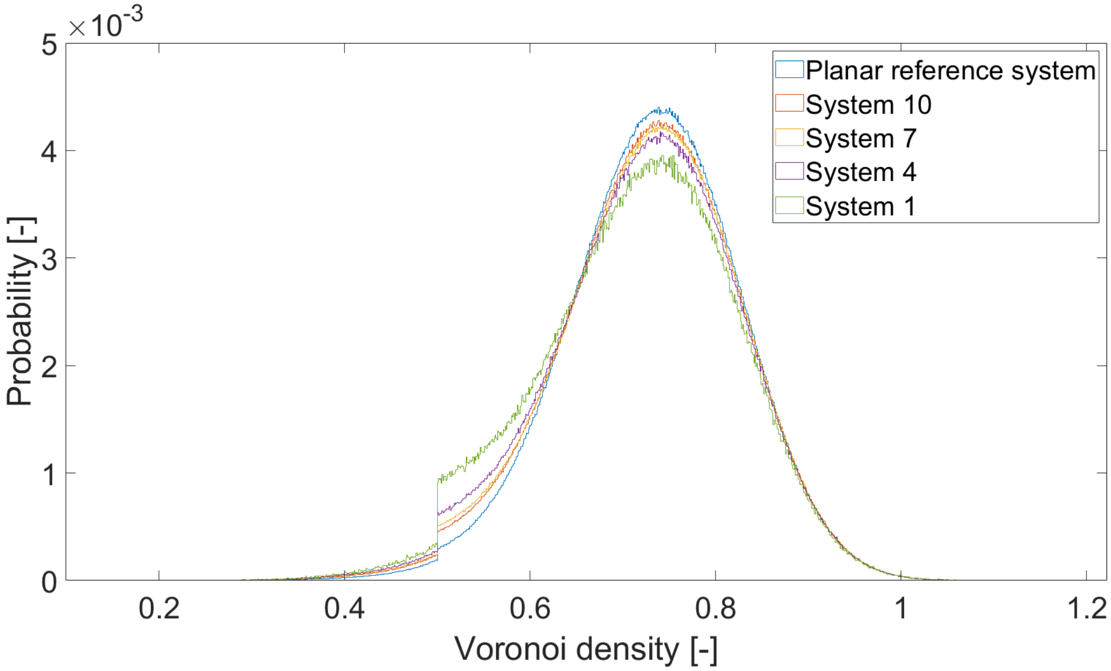


Figure 14: Normalized histograms of liquid phase density for five different systems, with the cutoff limit defined by the largest system method. At the cutoff limit there is a sudden drop, the drop height increases with decreasing droplet size.

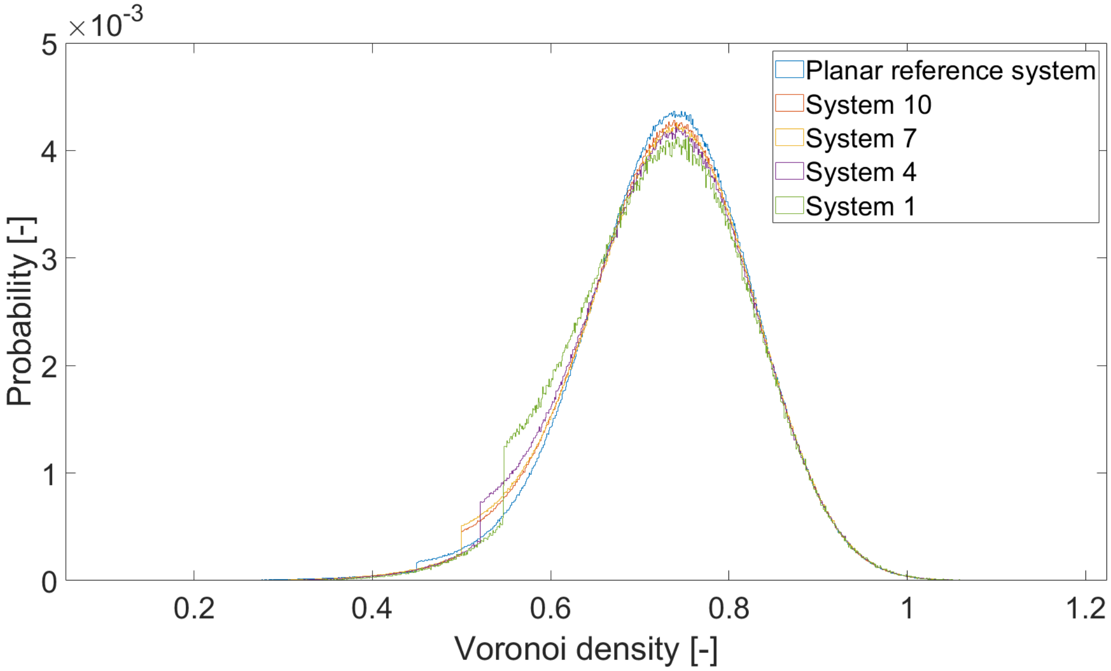


Figure 15: Normalized histograms of liquid phase density for five different systems, with the cutoff limit defined by the per-system pre-neighbor method.

4.2.1 Gaussian fitting

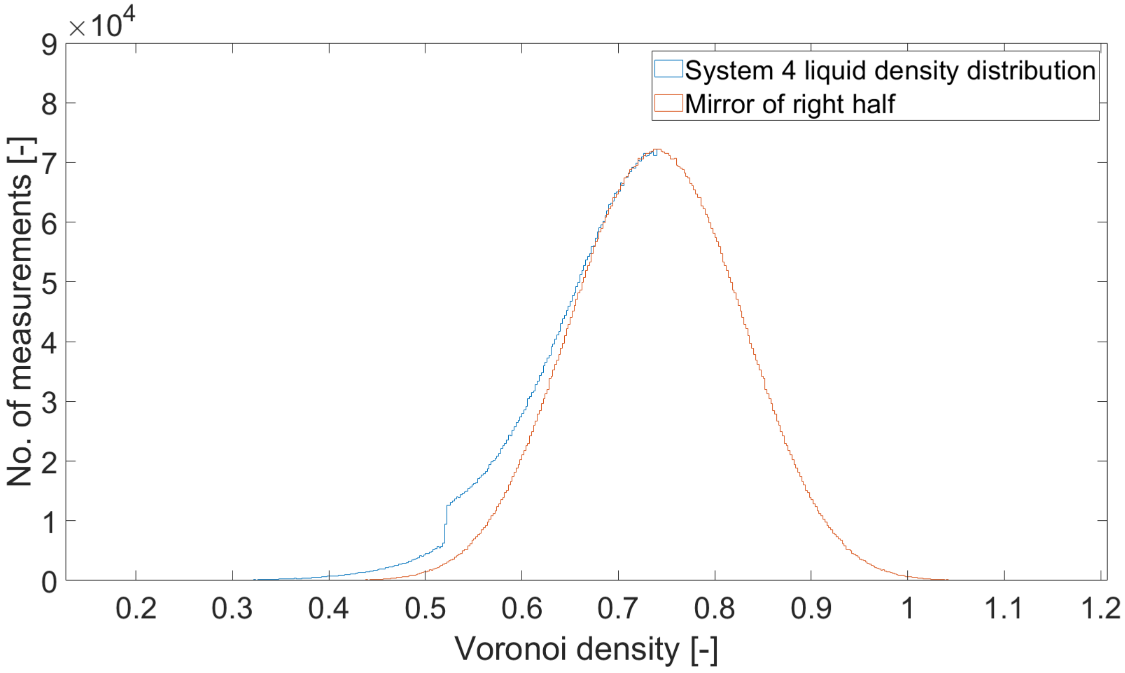


Figure 16: This figure shows the density distribution of the liquid phase for System 4, and the right half of the distribution mirrored around its peak at x = 0.74.

To check how symmetrical or Gaussian the density distributions are, two tests were applied. First, the distribution was cropped at its peak, and the right half of the distribution was mirrored. The mirrored distribution was plotted alongside the original to see to which degree the leftmost 'tail' of the original distribution, after $X = X_{high}$, aligns with the mirrored right half. In Figure 16, this plot is shown for System 4. The peak was determined to be at $X = 0.74$. It can be seen that the leftmost 'tail' of the density distribution does not coincide with the mirrored distribution.

For the second test, a Gaussian with the same peak, mean, and σ as the right half of the distribution was calculated. The Gaussian was then plotted alongside the density distribution to see how they align. Figure 17 shows this process done for System 4. The peak was at $X = 0.74$ as for the mirrored distribution, the amplitude was determined to be 0.0084 and σ to be 0.0845. As with the mirrored distribution, the leftmost 'tails' of the Gaussian and the density distribution do not coincide. The right part of the Gaussian does however show good agreement with the density distribution. This demonstrates what Figure 15 suggests, that the left half of the distribution shows far less Gaussian behavior than the right half. If we assume the individual particle density in the liquid phase is normally distributed, then the degree to which the distribution deviates from Gaussian behavior can be interpreted as the degree to which the method includes non-liquid particles in the liquid phase.

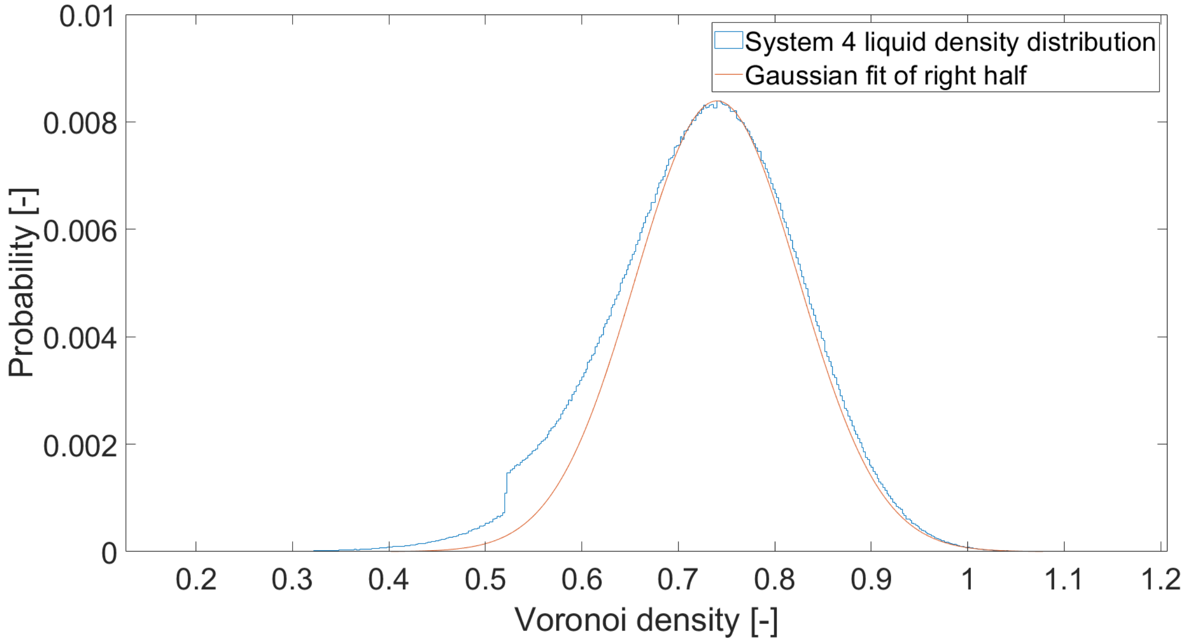


Figure 17: This figure shows the density distribution of the liquid phase for System 4, and a Gaussian with the mean, peak, and (sigma) of the right half of the distribution.

To test how using the per-system post-neighbor method to determine X_{high} would affect the density distribution, a Gaussian comparison was performed on the liquid phase density distribution of the System 1, with X_{high} defined by the per-system pre-neighbor and per-system post-neighbor method respectively. For the pre-neighbor case, the amplitude was determined to be 0.0122 and σ to be 0.0847, while for the post-neighbor case the amplitude was determined to be 0.0126 and σ to be 0.0848. The peak of the Gaussian was determined to be at $X = 0.74$ for both.

Figure 18 shows the Gaussian comparison where the per-system pre-neighbor method is used, while Figure 19 shows the comparison where the per-system post-neighbor is used. The left half of the density distribution after the gap can be seen to be closer to the Gaussian in the Figure 19 than in Figure 18, in other words, using the per-system post-neighbor method results in a more symmetrical distribution. This suggests that the per-system post-neighbor method would, were it successfully applied on every system, give a better definition of the system phases.

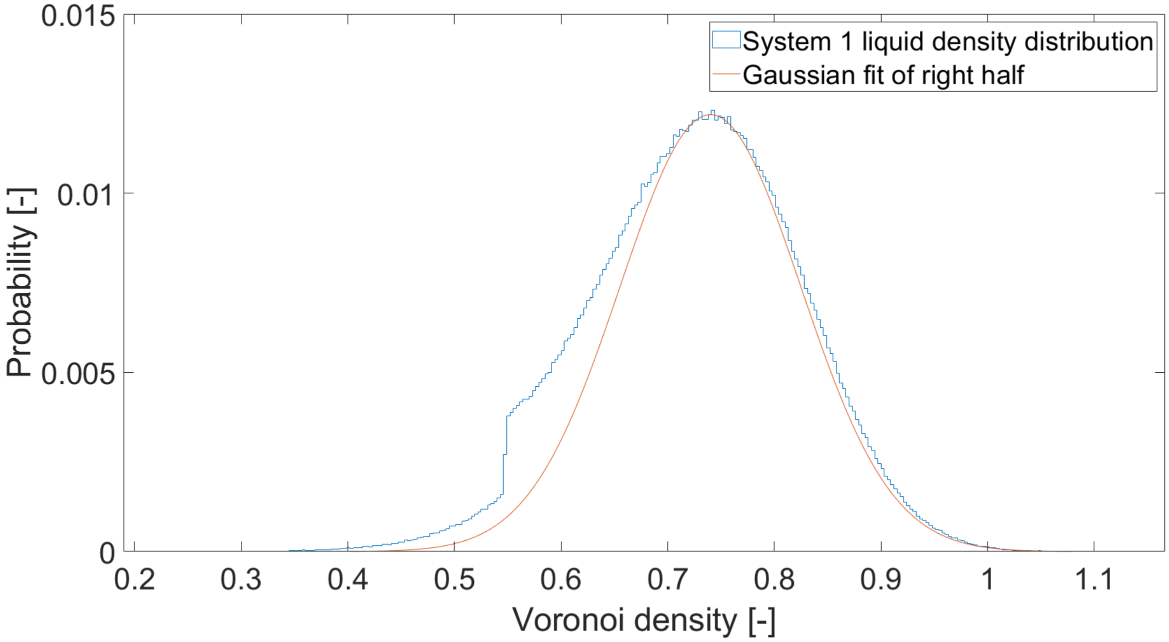


Figure 18: This figure shows the liquid density distribution for System 1, with the cutoff determined for the system before the neighbor step, as well as a Gaussian with the same peak, σ, and mean as the right half of the density distribution.

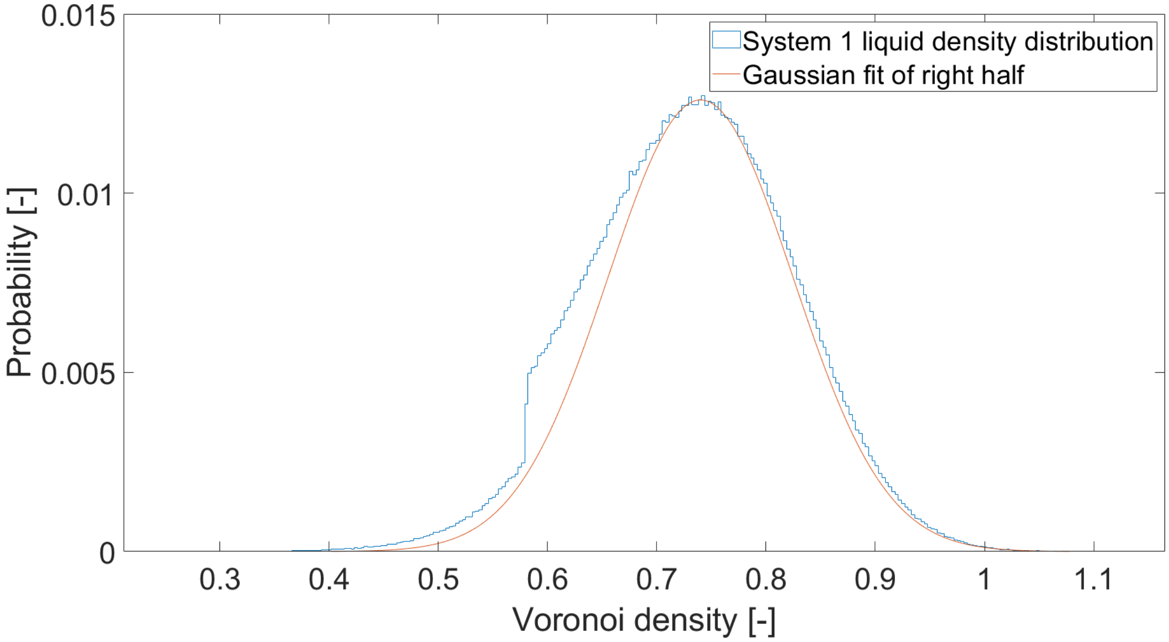


Figure 19: This figure shows the liquid density distribution for System 1, with the cutoff determined after the neighbor step, as well as a Gaussian with the same peak, σ, and mean as the right half of the density distribution

4.2.2 The Zs and N in the neighbor step

The values for Z_{vap} and Z_{liq} were determined qualitatively, by visualization of System 4 after it was classified by the Voronoi density-neighbor method, using different Z values. Figure 20 shows visualisations of System 4, after the Voronoi density-neighbor method was performed with varying Z_{vap} and Z_{liq}: 0.4, 0.5 and 0.6. N is 5 in all cases. Red particles are labeled vapor, yellow particles liquid and blue particles surface. The first choice of parameters is obviously not good, as there are almost no surface particles remaining. The third choice is also wrong, as several particles in the vapor phase are still labelled 'surface'. The second choice is the best of the three, as the 'surface' labelled particles are concentrated at the actual interface. From these results, it appears choosing the Zs around 0.5 is the most appropriate.

Next, more visualizations were made with the Zs around 0.5. Figure 21 shows visualisations of System 4, with varying Z_{liq}: 0.45, 0.5 and 0.55. In all cases, Z_{vap} is 0.45 and N is 5. Here, only the 'surface' labelled particles are shown. The difference between each case is slight, but $Z_{liq} = Z_{vap} = 0.45$ appears to give larger surface 'gaps' than the other parameters, while $Z_{liq} = 0.55$ gives the least 'gaps' while keeping the surface particles concentrated at the interface. Based on this qualitative assessment, Z_{vap} was chosen to be 0.45 and Z_{liq} to be 0.55.

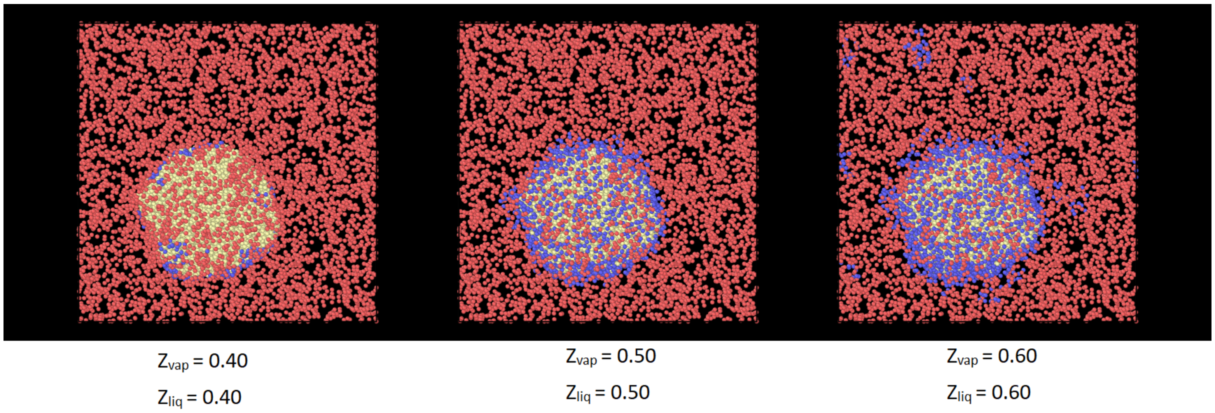


Figure 20: Visualisations of System 4, with varying Z_{vap} and Z_{liq}. $N = 5$. Vapor particles are red, liquid particles yellow and surface particles blue. Choosing $Z_{vap} = Z_{liq} = 0.4$ gives a poor definition of the surface, as does choosing $Z_{vap} = Z_{liq} = 0.6$. Choosing $Z_{vap} = Z_{liq} = 0.5$, in contrast, gives a qualitatively good surface definition.

The number of iterations N was also determined by a qualitative assessment. Here there are two factors to consider: Choosing too few iterations will result in 'surface' particles still being present in the vapor phase, while choosing N too large will make the processing time very long.

Figure 22 shows three different visualisations of System 4, after the Voronoi density-neighbor method was performed with varying N. The red particles were classified as vapor, the yellow particles as liquid, and the blue particles as surface. After only one iteration there are still several vapor particles labeled as 'surface'. After five iterations, there are no more falsely labelled vapor particles. After ten iterations, not much difference is observed from the five iteration case. From this, $N = 5$ was chosen as appropriate for this project.

The Z parameters and N chosen from assessment of System 4 were applied to each of the droplet-vapor systems, with the assumption that they would be equally valid for every droplet-vapor system.

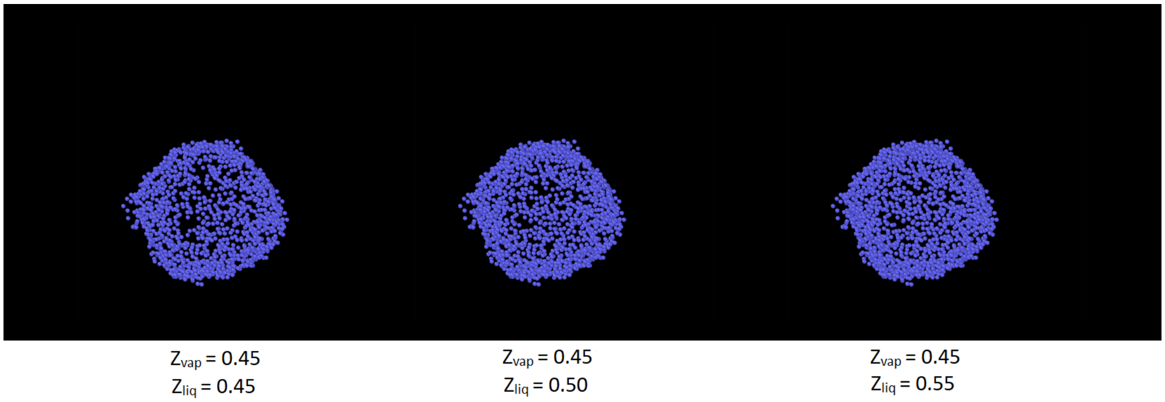


Figure 21: Visualisations of System 4, with varying Z_{liq}. Z_{vap} = 0.45 and $N = 5$. Only the surface particles are shown. Qualitatively, the 'gaps' in the surface are smallest with Z_{liq} = 0.55.

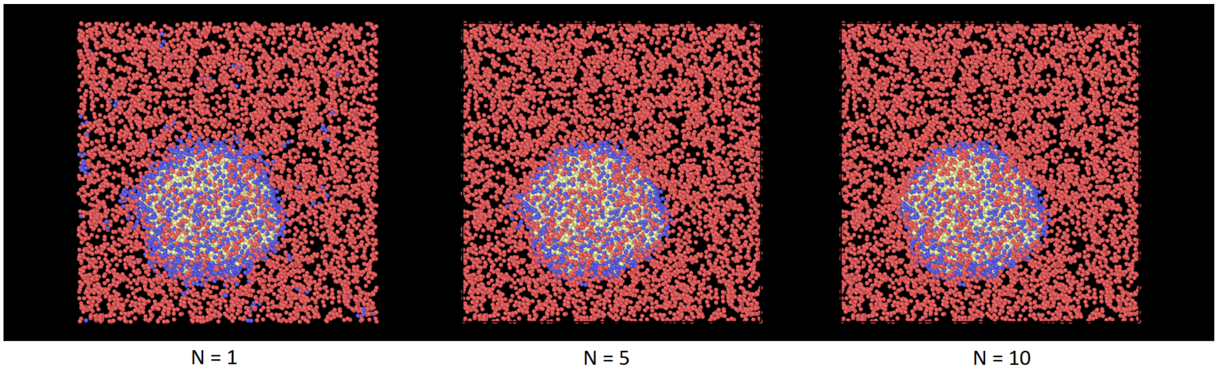


Figure 22: Visualisations of System 4 after the Voronoi density-neighbor method, with N = 1, 5, and 10. In each case, Z_{vap} was 0.45 and Z_{liq} was 0.55. Vapor particles are colored red, liquid particles yellow, and surface particles blue. For N = 1, there are still several 'surface' labelled particles in the vapor phase. For N = 5 and N = 10, the 'surface' labelled particles are concentrated at the droplet/vapor interface. There is little difference between the N = 5 and N = 10 systems.

4.3 Voronoi density of all systems

The average Voronoi density for particles in the liquid, vapor, and surface phases was calculated for each droplet-vapor system and the planar interface reference system. X_{high} was defined by the per-system pre-neighbor method as described in Section 3.5. Figure 23 shows the average Voronoi density of liquid particles as a function of inverse droplet radius. It can be seen that overall, the average Voronoi density decreases as droplet size decreases, in a roughly linear fashion, though there are points where the average density increases slightly for as system size decreases. These increases could be a result of the wrong bulk density being used to determine X_{high}, as described in Section 4.2.

Figure 24 shows the average density of surface particles. While overall the average density increases as droplet size decreases, the change in average surface density from one system to the next corresponds to the change in average liquid density between the same systems. Increasing liquid density is linked to decreasing surface density, and *vice versa*.

Figure 25 shows the average density of vapor particles. It can be seen that the average density remains approximately constant, and does not show any clear trend in how it varies with droplet size. This is as expected for a bulk vapor phase, and shows that the assumption that X_{low} should be approximately equal for all systems is justified.

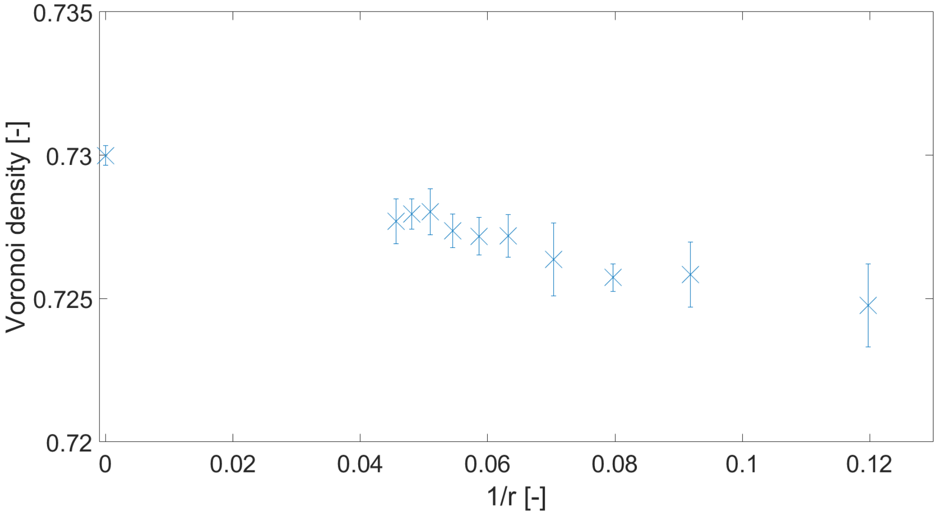


Figure 23: Average Voronoi density of liquid particles as a function of inverse droplet radius. Overall, the average density decreases with decreasing droplet size, in a linear manner.

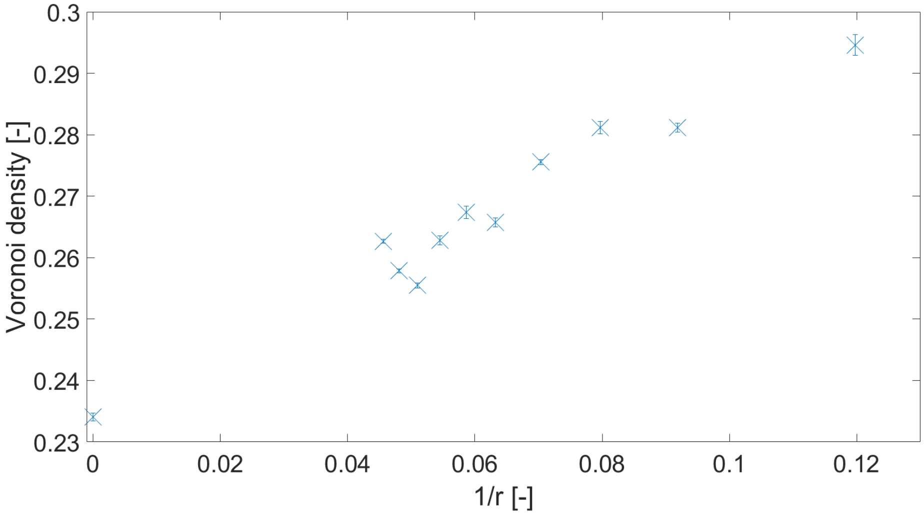


Figure 24: Average Voronoi density of surface particles as a function of inverse droplet radius. Overall, the average density increases with decreasing droplet size. Some of the error-bars are difficult to see as they are smaller than the marker size.

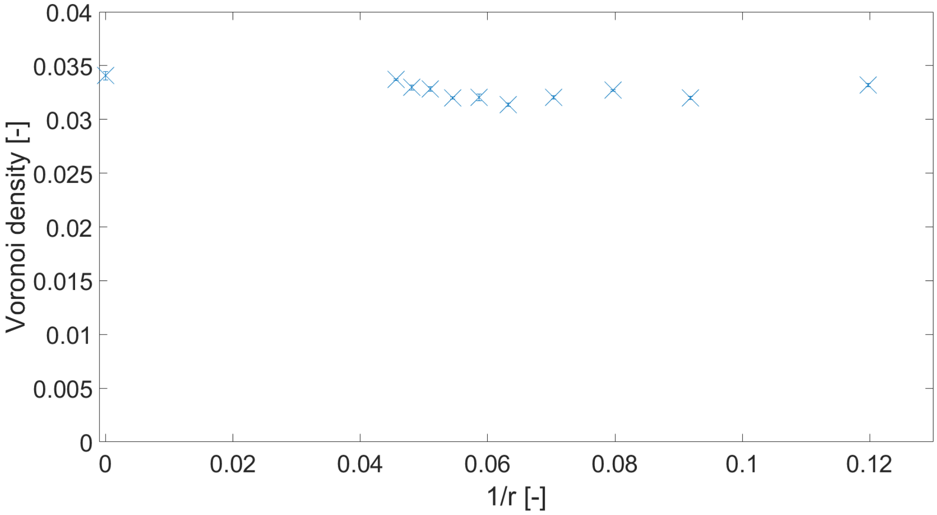


Figure 25: Average Voronoi density of vapor particles as a function of inverse droplet radius. It remains approximately constant with varying droplet size. Some of the error-bars are difficult to see as they are smaller than the marker size.

4.4 Potential energy

The average potential energy per unit volume of the liquid, vapor, and surface phases was calculated. X_{high} was defined by the per-system pre-neighbor method. The potential energy density of each of the liquid phases is shown in Figure 26, for the surface phases in Figure 27, and for the vapor phases in Figure 28.

For the liquid phase, the potential energy density increases as droplet size decreases. This increase is roughly linear with r^{-1}, which agrees with the small-scale hypothesis of Bedeaux et al.[8] For the surface, the potential energy density decreases with droplet size. Comparing with the results for the average Voronoi density, as shown in Figure 23 and Figure 24, the same trend is seen but inverted. This makes sense, as a particle with a high Voronoi density will tend to have closer neighbors and therefore a lower potential energy than a particle with a lower Voronoi density. For the vapor case, the potential energy density is also relatively constant with droplet size.

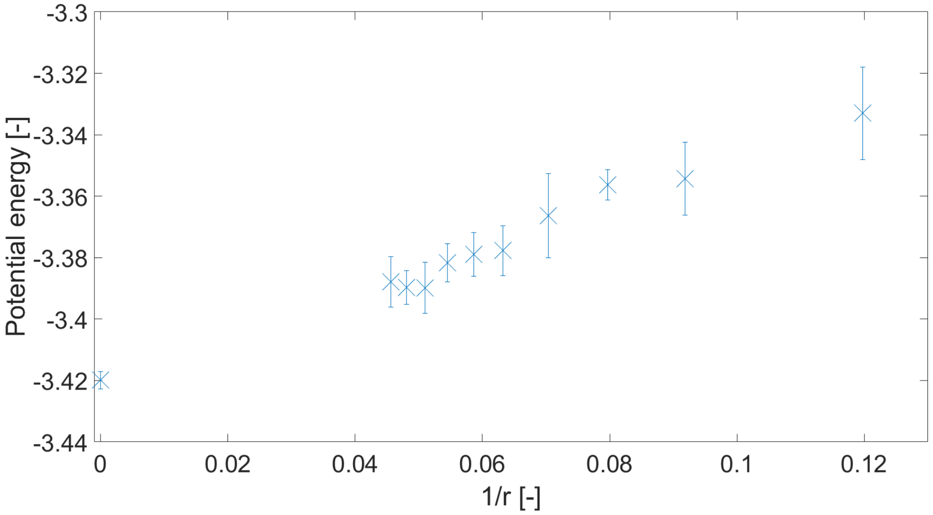


Figure 26: The potential energy density of the liquid phase, as a function of inverse droplet radius. It shows a roughly linear increase with decreasing droplet radius.

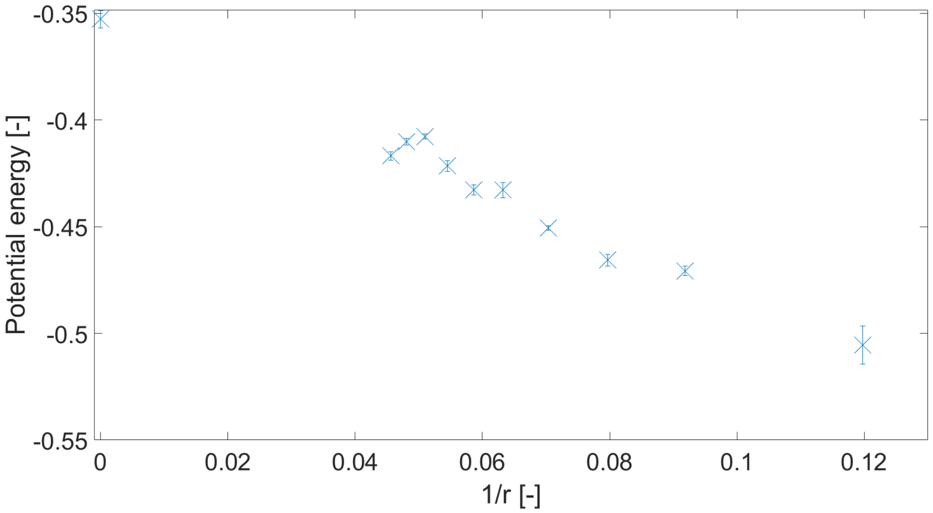


Figure 27: The potential energy density of surface particles, as a function of inverse radius. It decreases with decreasing droplet size. Some of the error-bars are difficult to see as they are smaller than the marker size.

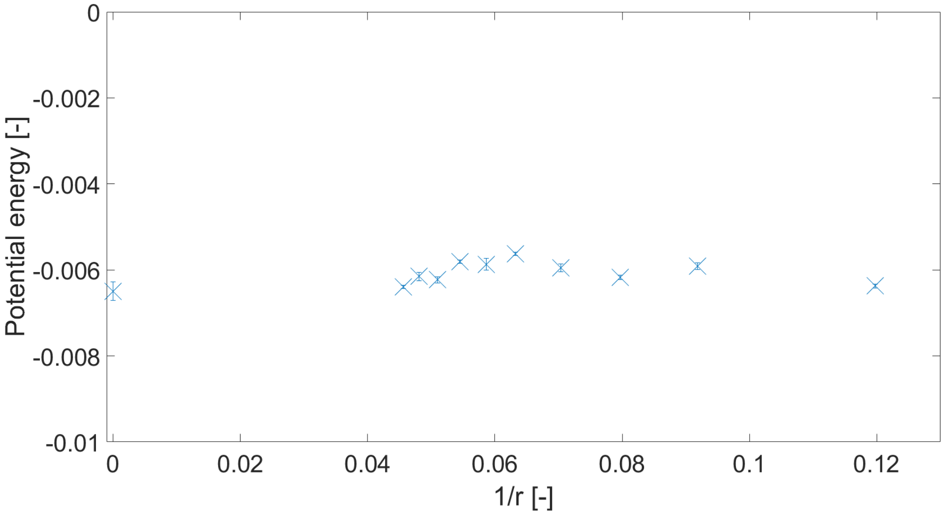


Figure 28: The potential energy density of vapor particles, as a function of inverse radius. It remains approximately constant with size. Some of the error-bars are difficult to see as they are smaller than the marker size.

4.5 Surface to volume ratio

As was explained in the introduction, a system's surface/volume ratio increases as the system gets smaller. Figure 29 shows the ratio of surface phase volume to total droplet volume (the sum of surface and liquid phase volume), while Figure 30 shows the ratio of surface particle amount to total particle amount, both plotted against inverse radius. X_{high} was defined by the per-system pre-neighbor method. Both ratios do indeed behave as expected, increasing as system size decreases.

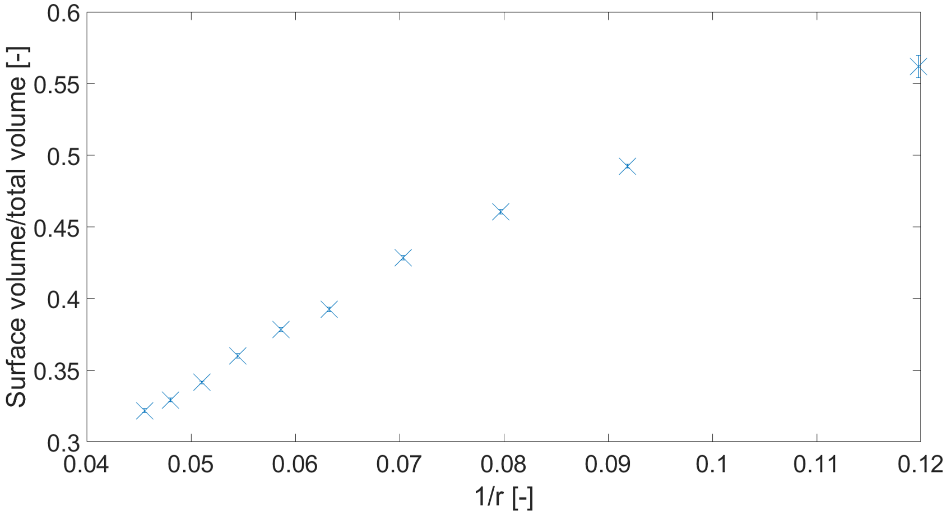


Figure 29: The ratio of surface phase volume to liquid and surface phase volume, as a function of inverse radius. The ratio increases as particle size decreases, which is as expected. Some of the error-bars are difficult to see as they are smaller than the marker size.

4.6 Droplet counting

The droplet counting algorithm described in Section 3.6 was implemented and used on both the single-droplet systems, as well as the two-droplet system. In both cases, the algorithm accurately returned the amount of droplets present. Figure 31 shows a visualization of the two-droplet system, where vapor particles are marked red, liquid particles are marked blue, the surface particles labeled 1 are marked yellow and the surface particles labeled 2 are marked white. The labels correspond to the two particles, demonstrating that the algorithm has accurately distinguished between the two separate droplets. It is also worth noting that the droplet with yellow surface particles crosses a periodic boundary. The Voronoi density-neighbor method handles this situation without any special considerations having to be applied.

The droplet-counting algorithm used is not particularly optimized, but serves as proof of concept that the density-neighbor method can be utilized to determine the amount of distinct units, such as droplets, present in a system.

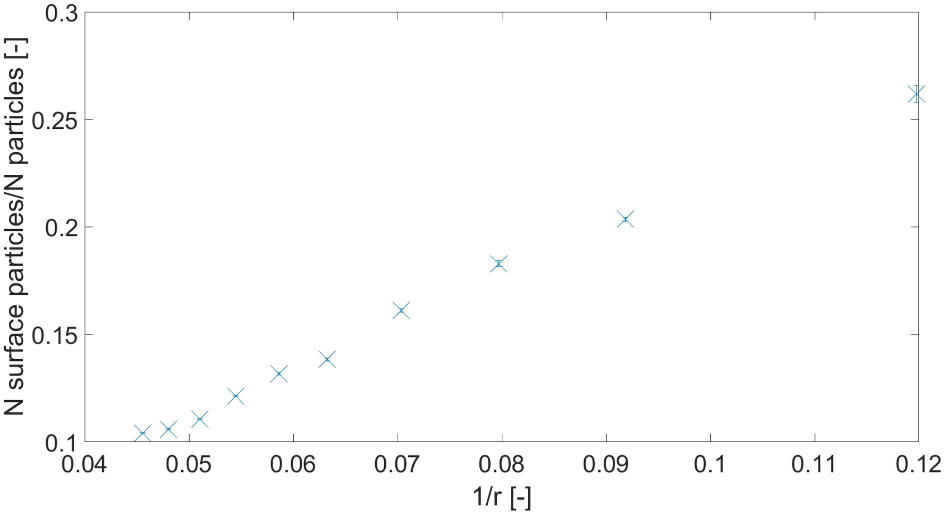


Figure 30: The ratio of the number of surface particles to the number of surface and liquid particles, as a function of inverse radius. It too behaves as expected, the ratio increasing as particle size decreases. Some of the error-bars are difficult to see as they are smaller than the marker size.

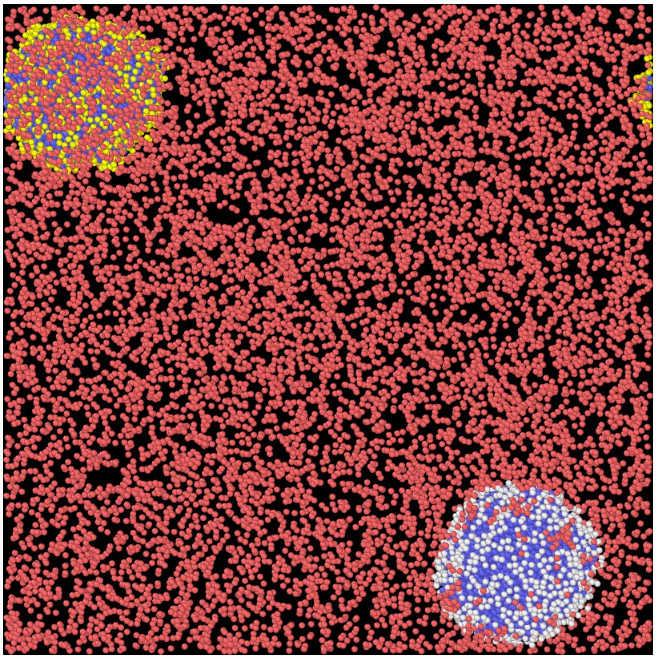


Figure 31: A visualisation of the two-droplet system. Vapor particles are colored red, liquid particles blue, surface particles of the first droplet yellow, and surface particles of the second droplet white. This shows the algorithm has indeed distinguished between the two droplets. The first droplet crosses a periodic boundary, a situation the density-neighbor method handles without issue.

4.7 Enthalpy

The enthalpy per volume for the liquid and vapor phase was calculated for each droplet-vapor system and for the planar interface reference system. The enthalpy of each individual particle was found by the normal equation for enthalpy:

$$H = U + pV \tag{24}$$

Figure 32 shows the enthalpy per volume of liquid, while Figure 33 shows the enthalpy per volume of vapor, both as a function of inverse droplet radius. These figures show that the liquid phase enthalpy per volume decreases as droplet size increases, while the vapor phase enthalpy per volume remains approximately constant. These are the same trends as was seen for how the Voronoi density and potential energy density varies with droplet size.

That the liquid enthalpy per volume follows this same trend is interesting to note, and is an additional sign that extensive properties of the droplet-vapor systems scale linearly with r^{-1}, as the small-scale hypothesis predicts.[8]

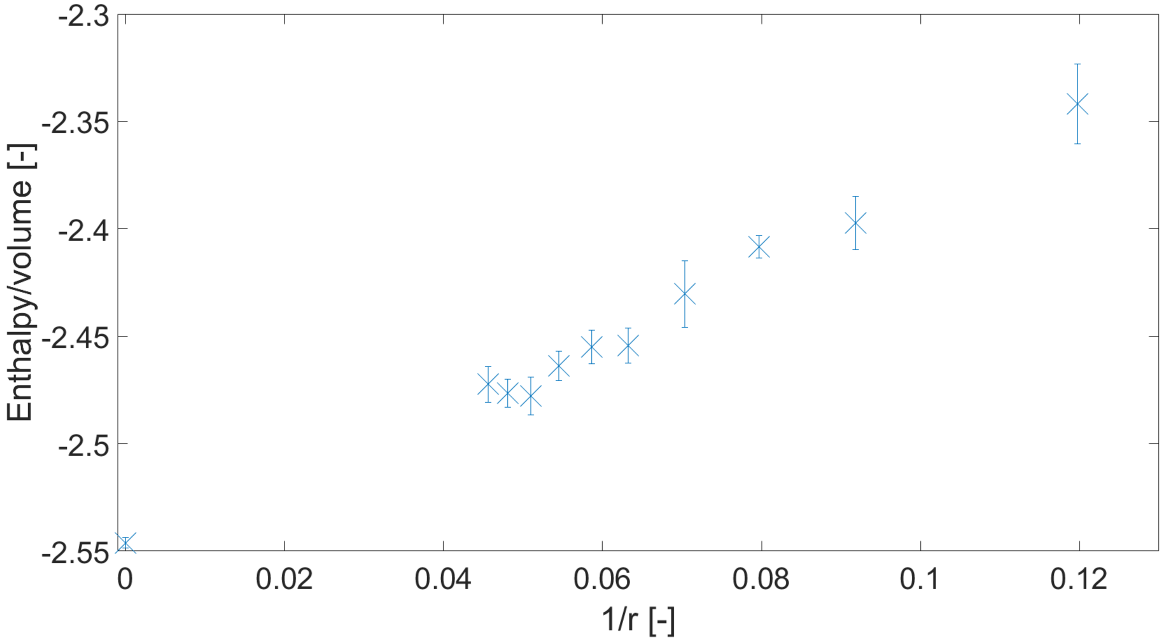


Figure 32: The enthalpy per volume of the liquid phase, for each of the different systems. It appears to decrease linearly with decreasing droplet radius.

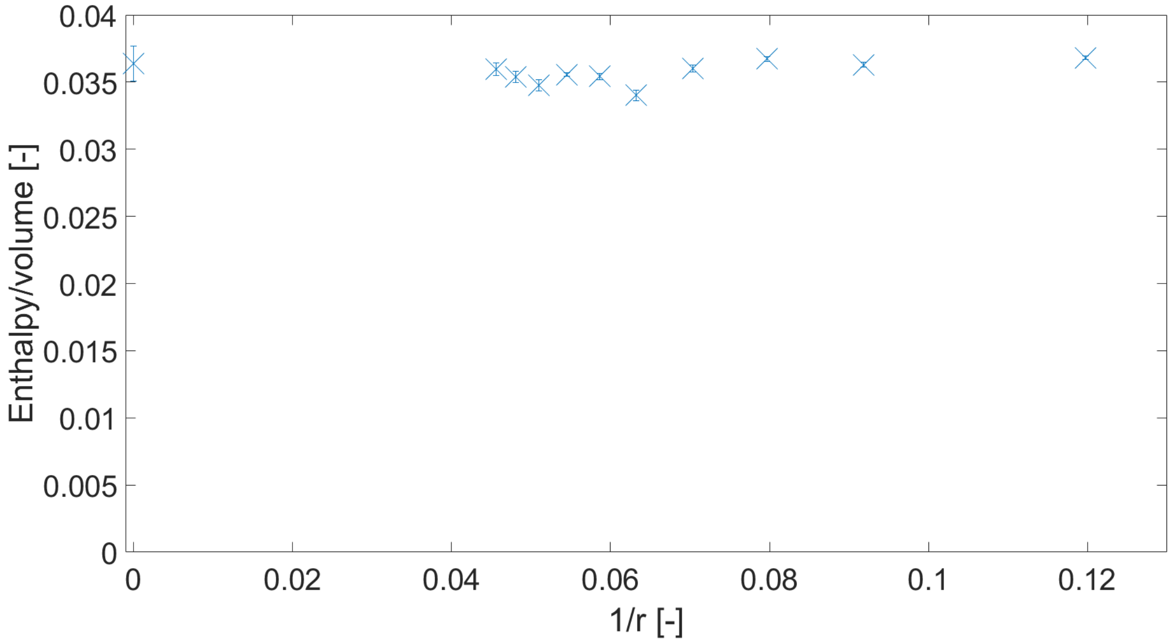


Figure 33: The enthalpy per volume of the gas phase, for each of the different systems. It remains approximately constant and independent of droplet radius. Some of the error-bars are difficult to see as they are smaller than the marker size.

4.8 Surface tension

The difference in pressure between the vapor- and liquid phase, Δp, was calculated for each system and plotted against inverse radius. As explained in Section 2.2, assuming the Young-Laplace equation holds, the slope of this plot will be equal to twice the surface tension.

In Figure 34, Δp for each droplet-vapor system is plotted against r^{-1}. X_{high} was defined by the per-system pre-neighbor method. In addition, a simple linear regression was performed, which is shown in the figure as the yellow line. The line has gradient 1.2, which translates to an approximate surface tension of 0.6, and crosses the y-axis at $y = -0.018$. This results agree with the surface tension for a LJ fluid truncated at 2.5 at $T = 0.8$ given by Trokhymchuk et al,[24] as shown in Table 3.

This result suggests the surface tension of each of the droplet-vapor systems investigated is given by the standard Young-Laplace equation, and as such any surface curvature effects are not strong enough to have significant influence on the surface tension.

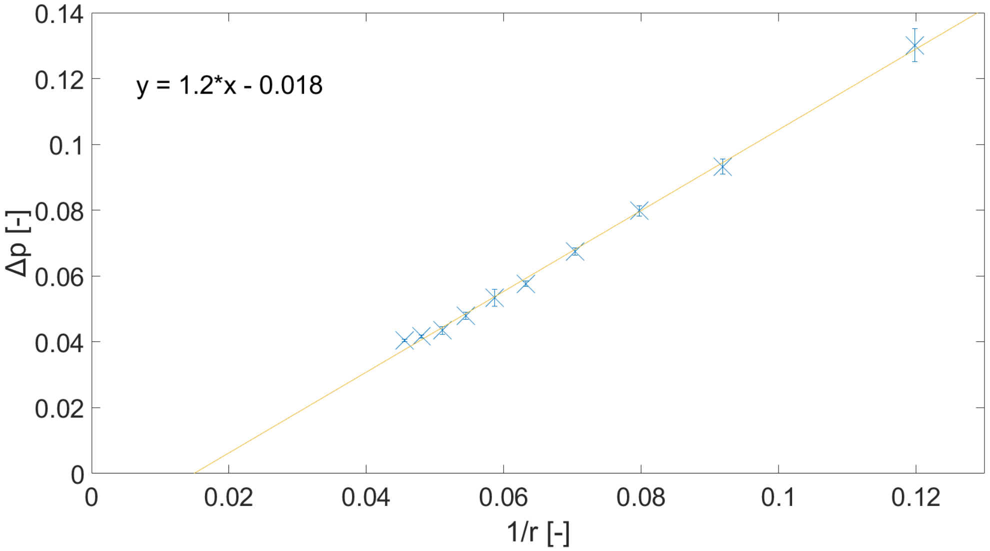


Figure 34: Difference in pressure as a function of r^{-1}*, and a linear fit. The phase limits are defined by the per-system pre-neighbor method. Assuming the Young-Laplace equation applies, the slope of the linear fit corresponds to twice the surface tension.*

5 Conclusion

Molecular dynamics simulation was used to simulate ten systems consisting of a droplet in equilibrium with vapor, with varying droplet size. A novel method for characterizing two-phase systems, the Voronoi density-neighbor method, was applied on the droplet-vapor systems. This method uses the Voronoi volume of each particle, defined through a procedure called Voronoi tessellation, as well as a list of particle neighbors to define which phase a given particle belongs to. It is implemented in two steps, the density step and the neighbor step.

Though we made an error in selecting the appropriate bulk densities, the Voronoi density-neighbor method appears promising for studying such systems. It has been demonstrated that the method can easily handle situations such as droplets crossing periodic boundaries, and the presence of multiple separate droplets. The method also offers the ability to determine the number of droplets present in a simulation, which could be especially useful for studying a two-phase system where a variable number of many small droplets are present.

Several average properties - Voronoi density, potential energy, and enthalpy - were calculated for each phase of the droplet-vapor systems. For the liquid phase, the results appear to show a linear dependence of extensive properties with inverse radius. This agrees well with the small-scale hypothesis put forward by Bedeaux et al.[8] The calculated surface/volume ratio increases with decreasing droplet size, as is expected. The surface tension of the droplet-vapor systems appears to follow the Young-Laplace equation, meaning surface curvature effects on the surface tension are negligible.

There exists an alternative method for defining density limits in the density step, the per-system post-neighbor method. Based on the closer to Gaussian liquid density distribution obtained for System 1 using this alternative method, we believe using it will give better results than the procedure used in this project, the per-system pre-neighbor method. The per-system post-neighbor method was, however, found too time-consuming to implement for every system. Were it successfully implemented, it would likely improve the Voronoi density-neighbor method.

6 Future work

Here we will finally discuss some future work we recommend, to further improve the Voronoi density-neighbor method, and some suggestions for how the method can be applied.

6.1 Investigating the effects of using incorrect bulk density values

In this project, the wrong bulk density values were used to determine X_{low} and X_{high}. A logical next step would be to instead use the bulk density values for a truncated and not shifted potential, provided by Trokhymchuk et al. and shown in Table 3.[24] Then the effects our error had can be determined by comparing the thermodynamic properties obtained in both cases. It is also possible to perform equivalent simulations with a truncated and shifted Lennard-Jones potential, use the same bulk density values as were used in this project to determine X_{low} and X_{high}, and examine the difference in results in this case as well.

6.2 Script optimization

The neighbor step of the Voronoi density-neighbor method is by far the most time-consuming. In this project it was implemented as a simple Python script, which uses only a single processor. If a script for the neighbor step was made which uses multi node processing, high-performance computing could be used to speed up this part of the process. The other post-processing scripts could also be written to support multi node processing. Making the neighbor step faster would also make the per-system post-neighbor method for determining X_{high} more viable.

6.3 Implementing the per-system post-neighbor method

From the Gaussian plots shown in Figure 18 and Figure 19, we see that the per-system post-neighbor method results in a closer to Gaussian distribution of liquid phase density. Still, the per-system post-neighbor method is difficult to implement due to the time required to perform each guess. As discussed above, creating a multi node processing script for the neighbor step would speed it up and make the per-system post-neighbor method more viable. If some sort of relationship between the change in X_{high} and the change of droplet size could be found, it could also make the process of guessing new values faster.

6.4 Quantitative calculation of the Z-parameters in the neighbor step

In this project, the parameters Z_{low}, Z_{high} and N in the neighbor step were determined through a qualitative assessment. A logical next step is developing a more quantitative method for determining appropriate values for them. This could be achieved by calculating properties for different Zs and N, for example the surface tension, and comparing the results.

The parameters Z_{low} and Z_{high} were determined by examining a single droplet-vapor system and assumed to be appropriate for every system. In the future, this assumption should be tested by examining each system separately, and qualitative and quantitative study of how the system properties vary with Z_{low} and Z_{high}.

6.5 Applying the model to a real system

As-is, the droplet-vapor model could be used to study systems where the only inter-particle forces are Pauli repulsion and Van der Waals forces, like systems consisting of noble gases or non-polar molecules like methane. To obtain results for a certain type of real system, it is fairly straightforward convert the unitless results to real units by the process explained in Section 2.3.3.

In order to use the model to study systems where other inter-particle are present, such as those resulting from Coulomb or dipole-dipole interactions, it would be necessary to use a different potential which accounts for these forces. To use the system to study water, for example, the potential would also account for the effects of hydrogen bonds. Switching potentials can introduce additional challenges: For example, as hydrogen bonds are much stronger than Van der Waals forces, a water model system created in the same fashion would result in a much less dense vapor phase.

6.6 Investigation of other properties

In this project, the Voronoi density, potential energy, enthalpy and surface tension were attempted calculated. There are other properties the Voronoi density-neighbor method could be used to calculate in the future. Some examples will be listed here.

The *residence time* is the amount of time a particle spends in a region before leaving. In the case of droplet-vapor systems, the residence time of vapor particles which enter the surface would be interesting to determine. This procedure could involve labeling particles which go from being vapor particles to being surface particles, and computing the number of time-steps such a particle stays a surface particle.

Diffusion describes the movement of particles in a solvent due to Brownian motion. The movement of solvent particles in the solvent is referred to as *self-diffusion.* [9] In a droplet-vapor system, the self-diffusion of liquid particles would be of great interest. This could possibly be studied by labelling specific liquid particles

Another property of great interest is the heat capacity. Determining the heat capacity in a small system can be done by measuring fluctuations in the Hamiltonian, in the grand canonical or μVT ensemble, where the chemical potential, volume, and temperature are kept constant. This calculation is not trivial and requires knowing the chemical potential of the system. [8]

7 Aknowledgements

I would like to thank my supervisors, Sondre Kvalvåg Schnell and Vilde Bråten, for their assistance with my project and with this thesis. I would also like to thank the FERROICS I team for our weekly discussions and presentations, which were very helpful during an especially difficult time.

Also, I would like to thank the student organizations at NTNU, in particular Delta and Timini, for creating a warm and social environment which has been invaluable in my time as a student there. Finally, I want to thank my family for believing in me and supporting me in both good and difficult times.

References

[1] Gabor L. Hornyak et al. *Introduction to Nanoscience & Nanotechnology.* CRC Press, first edition, 2009.

[2] Bjørn A. Strøm et al. Size and shape effects on the thermodynamic properties of nanoscale volumes of water. *Physical Chemistry Chemical Physics*, (13), 2017.

[3] Georgios A. Sotiriou and Sotiris E. Pratsinis. Antibacterial activity of nanosilver ions and particles. *Environmental Science and Technology*, 44, 2010.

[4] Silpa Raj et al. Nanotechnology in cosmetics: Opportunities and challenges. *Journal of Pharmacy and Bioallied Sciences*, 4, 2012.

[5] Seung Woo Lee et al. High-power lithium batteries from functionalized carbon-nanotube electrodes. *Nature Nanotechnology*, 5, 2010.

[6] Benjamin S. Harrison and Anthony Atala. Carbon nanotube applications for tissue engineering. *Biomaterials*, 2007.

[7] Terrell L. Hill. Thermodynamics of small systems. *The Journal of Chemical Physics*, 36(12), 1962.

[8] Dick Bedeaux, Signe Kjelstrup, and Sondre K. Schnell. *Nanothermodynamics. General theory.* PoreLab, 2020.

[9] Daan Frenkel and Berend Smit. *Understanding Molecular Simulation: From Algorithms to Applications.* Academic Press, second edition, 2002.

[10] Michael P. Allen and Dominic J. Tildesley. *Computer Simulation of Liquids.* Oxford University Press, second edition, 2017.

[11] Franz Aurenhammer. Voronoi diagrams—a survey of a fundamental geometric data structure. *ACM Computing Surveys*, 23(3), 1991.

[12] Jared T. Fern, David J. Keffer, and William V. Steele. Measuring coexisting densities from a two-phase molecular dynamics simulation by voronoi tessellations. *Journal of Physical Chemistry B*, 2007.

[13] Ken A. Dill, Sarina Bromberg, and Dirk Stigter. *Molecular Driving Forces.* Garland Science, second edition, 2011.

[14] Paul C. Hiemenz and Raj Rajagopalan. *Principles of Colloid and Surface Chemistry.* Marcel Dekker, third edition, 1997.

[15] Andrew R. Leach. *Molecular Modelling: Principles and Applications.* Pearson Education, second edition, 2001.

[16] B. Smit. Phase diagrams of Lennard-Jones fluids. *The Journal of Chemical Physics*, 96(11), 1992.

[17] J.J. Nicholas et al. Equation of state for the lennard-jones fluid. *Molecular Physics*, 37(5), 1979.

[18] D. J. Evans and B. L. Holian. The nose–hoover thermostat. *The Journal of Chemical Physics*, 83, 1985.

[19] J.M. Kang. *Voronoi Diagram.* In *Encyclopedia of GIS* by S. Shekar, H. Xiong, X. Zhou. Springer, second edition, 2017.

[20] Voronoi diagram generator. By Frederik Brasz. Available at http://cfbrasz.github.io/Voronoi.html. Accessed 22.12.2020.

[21] Alexander Stukowski. Visualization and analysis of atomistic simulation data with ovito–the open visualization tool. *Modelling and Simulation in Materials Science and Engineering*, 18(1), 2010.

[22] Steve Plimpton. Fast parallel algorithms for short-range molecular dynamics. *Journal of Computational Physics*, 117(1), 1995.

[23] M.J.P. Nijmeijer et al. A molecular dynamics simulation of the Lennard-Jones liquid-vapor interface. *The Journal of Chemical Physics*, 89(6), 1988.

[24] Andrij Trokhymchuk and José Alejandre. Computer simulations of liquid/vapor interface in lennard-jones fluids: Some questions and answers. *The Journal of Chemical Physics*, 1999.

[25] LAMMPS Molecular Dynamics Simulator - units command. Available at https://lammps.sandia.gov/doc/units.html, accessed 19.12.2020.

[26] John R. Taylor. *An Introduction to Error Analysis: The Study of Uncertainties in Physical Measurements*. University Science Books, second edition, 1997.

A Appendix

A.1 Reduced units

Table 4 shows a list of the reduced units relevant for this project, and their definitions as used in the molecular dynamics program LAMMPS.[25] The quantities marked with an asterisk are unitless, while the equivalent quantity without an asterisk is in units.

Table 4: This table shows the unitless quantities used by LAMMPS for Lennard-Jones simulations, and their definitions. [25]

Quantity	Lennard-Jones unit	Definition
Mass	m	
Distance	σ	$x^* = \frac{x}{\sigma}$
Energy	ϵ	$E^* = \frac{E}{\epsilon}$
Time	τ	$\tau^* = \tau\sqrt{\frac{\epsilon}{m\sigma^2}}$
Velocity	$\frac{\sigma}{\tau}$	$v^* = v\frac{\tau}{\sigma}$
Force	$\frac{\epsilon}{\sigma}$	$f^* = f\frac{\sigma}{\epsilon}$
Temperature	T^*	$T^* = \frac{Tk_B}{\epsilon}$
Pressure	p^*	$p^* = p\frac{\sigma^3}{\epsilon}$
Density	$\frac{m}{volume}$	$\rho^* = \rho\sigma^{dim}$

A.2 Statistics and error analysis

The expected value for a quantity x, measured at N discrete points, is simply the sum divided by the number of measurements, or the mean:

$$\bar{x} = \frac{1}{N}\Sigma_{i=1}^{N} x_i \tag{25}$$

Where $\bar{x}$ is the mean, and the x_is are each individual measurement.[26] The *standard deviation* is defined as the square root of the expected value of the squared deviation from the mean:

$$\sigma = \sqrt{\frac{\Sigma_{i=1}^{N}(x_i - \bar{x})^2}{N-1}} \tag{26}$$

The reason for dividing by $N-1$ rather than N in Equation 26, is to correct for the fact that the mean calculated from Equation 25 will differ slightly from the true mean. For any one measurement of x, it is 68% likely it will lie within one standard deviation of the true value, and 95% likely it will lie within two standard deviations[26].

The *standard deviation of the mean*, also called the error of the mean, is defined as:

$$\sigma_{\bar{x}} = \frac{\sigma}{\sqrt{M}} \tag{27}$$

Where M is the number of uncorrelated data points used to calculate the mean. This means that 68% of measurements of the mean will lie within $\sigma_{\bar{x}}$ of the true mean, and 95% of them will lie within $2\sigma_{\bar{x}}$.[26]

If every data point is uncorrelated, then $M = N$. If there is correlation between data points, however, the error calculated using N will be too small. To correct for possible correlation in the data, a process called *block analysis* can be used. This involves splitting the data into blocks of equal size, computing the average of each block, and taking the standard deviation of these blocks from the sampled mean. For example, if the data is split into S blocks of equal size, S block averages are obtained: $B_1, B_2, ...B_S$. This results in a corrected standard deviation given by:

$$\sigma(corrected) = \sqrt{\frac{\Sigma_{i=1}^{S}(B_i - \bar{x})^2}{S-1}} \tag{28}$$

And a corrected error: [26]

$$\sigma_{\bar{x}}(corrected) = \frac{\sigma(corrected)}{\sqrt{S}} \tag{29}$$

NTNU

Kunnskap for en bedre verden